Population testing for Lipoprotein(a): Health economic analyses

Protocol
March 3, 2025
https://github.com/jimb0w/LPA

Jedidiah Morton
Research Fellow
Jedidiah.Morton@monash.edu
Monash University, Melbourne, Australia

Contents

1 Introduction

Lipoprotein(a) (Lp(a)) is an important risk factor for cardiovascular disease (CVD) in a subset of the population with high Lp(a) [1, 2, 3]. Depending on the cut-off used to define "high" Lp(a), anywhere from ∼10-20% of individuals may have high Lp(a) [3, 4]. However, it is not currently routinely measured in clinical practice [4], meaning most individuals with high Lp(a) are unaware of this status and, consequently, do not modify their CVD risk or have their CVD risk modified by a clinician.

While there are currently no available options for treating high Lp(a) itself (beyond apheresis), it can still be useful to reclassify risk and prompt more intensive management of other cardiovascular risk factors that might otherwise go untreated. Moreover, highly effective treatments to lower Lp(a) are in a late stage of development [5, 6], and should be available soon.

Thus, widespread Lp(a) testing could be a useful way to improve the prevention of CVD in the population. Indeed, it is fully expressed by the first year of life [7, 8, 9], and it is largely genetically determined [10] (although genetic testing is not required), meaning it will generally only need to be measured once in a lifetime in most people. Moreover, the test cost is relatively cheap (depending on the jurisdiction, ranging from a few dollars to approximately US$50).

However, to date, no studies have evaluated the cost-effectiveness of widespread Lp(a) testing. Therefore, herein, we will construct a state-of-the-art health economic model that simulates the primary and secondary prevention of CVD to perform two health economic analyses. First, we will evaluate the cost-effectiveness of testing for Lp(a) in the primary prevention population, where testing is followed by treatment of other cardiovascular risk factors, compared to current practice. Second, we will evaluate the cost-effectiveness of testing for Lp(a) in both the primary and secondary prevention populations, where testing is followed by treatment for Lp(a) directly, compared to current practice.

2 Population, intervention, and control

2.1 Study 1: Lp(a) testing and treatment of other risk factors

The population for this study will be the primary prevention population initially aged between 40 and 69 years, followed up until age 85 years (i.e., this study will use a lifetime horizon). People aged 70 and above are excluded because they are almost always at high risk and thus warrant treatment [11], meaning testing for Lp(a) with the goal of modifying traditional risk factors would not be appropriate. Conversely, people aged below 40 years are rarely recommended for pharmacological treatment [11] and were thus also excluded. We have deliberately not specified a country as this analysis will be repeated from the perspective of multiple countries.

The control scenario will be based on current standard of care using European guidelines on CVD prevention [11]. These guidelines are, by design, ambiguous and thus do not provide specific treatment recommendations for individuals as they encourage personalising recommendations. For example, the guidelines state: "Across the entire range of CVD risk, the decision to initiate interventions remains a matter of individual consideration and shared decision-making." Therefore, in this analysis, the control scenario will be broadly based on the guidelines, but recommendations of these guidelines will be oversimplified to the following (also outlined in Figure 2.1:

1. All individuals with diabetes receive both a high intensity statin and pharmacological treatment for hypertension if they have a systolic blood pressure (SBP) of ≥140 mmHg.

2. All individuals without diabetes aged 40 and above will be tested for CVD risk every 5 years using the updated Systemic Coronary Risk Estimation (SCORE2) algorithm [12].

3. Individuals deemed "Very high risk" from SCORE2 will be treated with a high intensity statin (regardless of baseline low-density lipoprotein-cholesterol (LDL-C); statins reduce CVD risk regardless of baseline LDL-C levels [13]) and will receive pharmacological anti-hypertensive treatment if they have an SBP of ≥140 mmHg. In the European guidelines, "Very high risk" thresholds are ≥7.5% for individuals aged <50 years and ≥10% for individuals aged 50-69 years.

4. Individuals deemed 'High risk" or lower will not be pharmacologically treated unless they have an LDL-C level of ≥5.0 mmol/L or a SBP of ≥160 mmHg.

5. Individuals become "Very high risk" once they reach age 70 years and receive both a high intensity statin and pharmacological treatment for hypertension if they have a systolic blood pressure (SBP) of ≥140 mmHg.

The high intensity statin in this study will be atorvastatin 80 mg, leading to a mean reduction in LDL-C of 51.7% (51.2, 52.2) [14]. This figure was derived from a systematic review and meta-analysis of randomised clinical trials.

Pharmacological therapy for lowering SBP will be based on the SPRINT trial [15], because it is the main trial cited when recommending pharmacological SBP lowering in the European guidelines [11]. In the intensive SBP control arm of this trial, it took a mean of 2.8 medications to achieve an approximately 20 mmHg reduction in SBP. We will base the intervention on the most common medications used in the intensive arm – losartan 100 mg, chlortalidone 25 mg, and amlodipine 10 mg and assume they lead to a 20 mmHg (±25%) reduction in SBP.

This is obviously a *gross* oversimplification of clinical practice. But it is simply impossible to capture the nuances of clinical practice in a population of this size and in such an ambiguous area as primary prevention of CVD. We think this will be a reasonable approximation of current practice for CVD and, thus, a fair comparator for the implementation of an Lp(a) testing study.

The intervention scenario will be as above, except with further testing for Lp(a) among individuals in the "High risk" group immediately following risk estimation with SCORE2. The result of this testing will be to reclassify risk based on Lp(a) levels. People with an Lp(a) of at least 192 nmol/L (90 mg/dL) [1] will be reclassified into the "Very high risk" category and treated accordingly (even if more sophisticated interpretations have been recently recommended [17], it will be impossible to incorporate them into this analysis). Lp(a) levels of 105 nmol/L (50 mg/dL), 149 nmol/L (70 mg/dL) and 236 nmol/L (110 mg/dL) will be used in scenario analyses. Lp(a) testing will only occur once at the age the person first reaches threshold for testing. In the control scenario, no Lp(a) testing will occur.

Therefore, the key difference between the intervention and control scenarios will be the number of individuals deemed "Very high risk" – testing for Lp(a) in the intervention scenario will lead to an increased number of individuals receiving LDL-C-lowering and SBP-lowering treatment.

Both the intervention and control are presented graphically in Figures 2.1-2.2.

[1] Just a note on units for Lp(a) in this protocol. the original units provided in the UK Biobank and the prferred units for reporting in Australia and the UK are nmol/L. However, for the model to function, we had to use studies that reported the association of Lp(a) with outcomes in mg/dL. Therefore, throughout, mg/dL are used in the model machinery, but results are reported in nmol/L. The conversion from nmol/L to mg/dL is not ideal, but the formula Lp(a), nmol/L $= 2.18 \times$ Lp(a), mg/dL $- 3.83$ has been shown to be highly accurate [16]. All uses of Lp(a) in the protocol show both values, except when referring to the results of Mendelian randomisation studies.

Figure 2.1: Schematic of the control scenario in study 1.

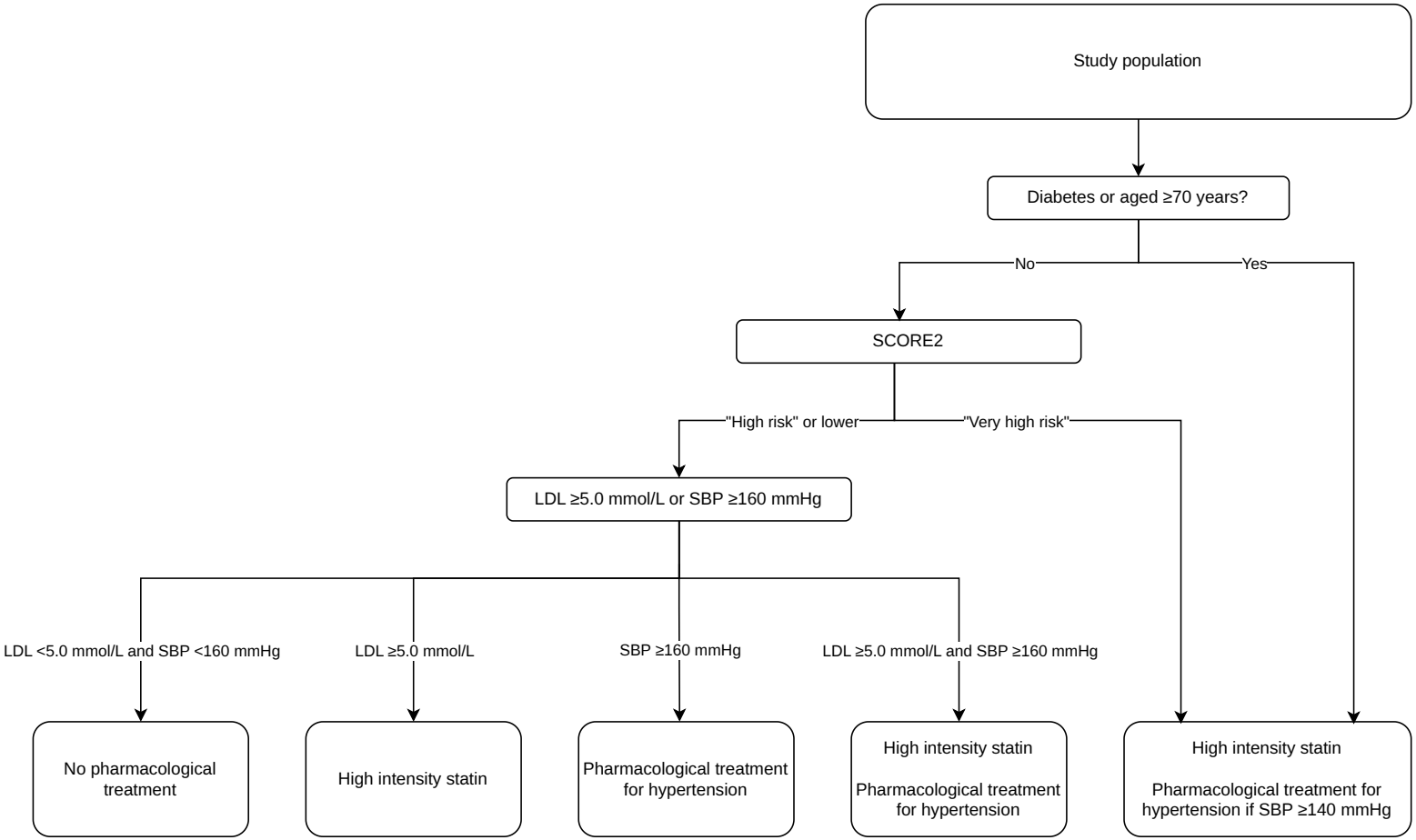
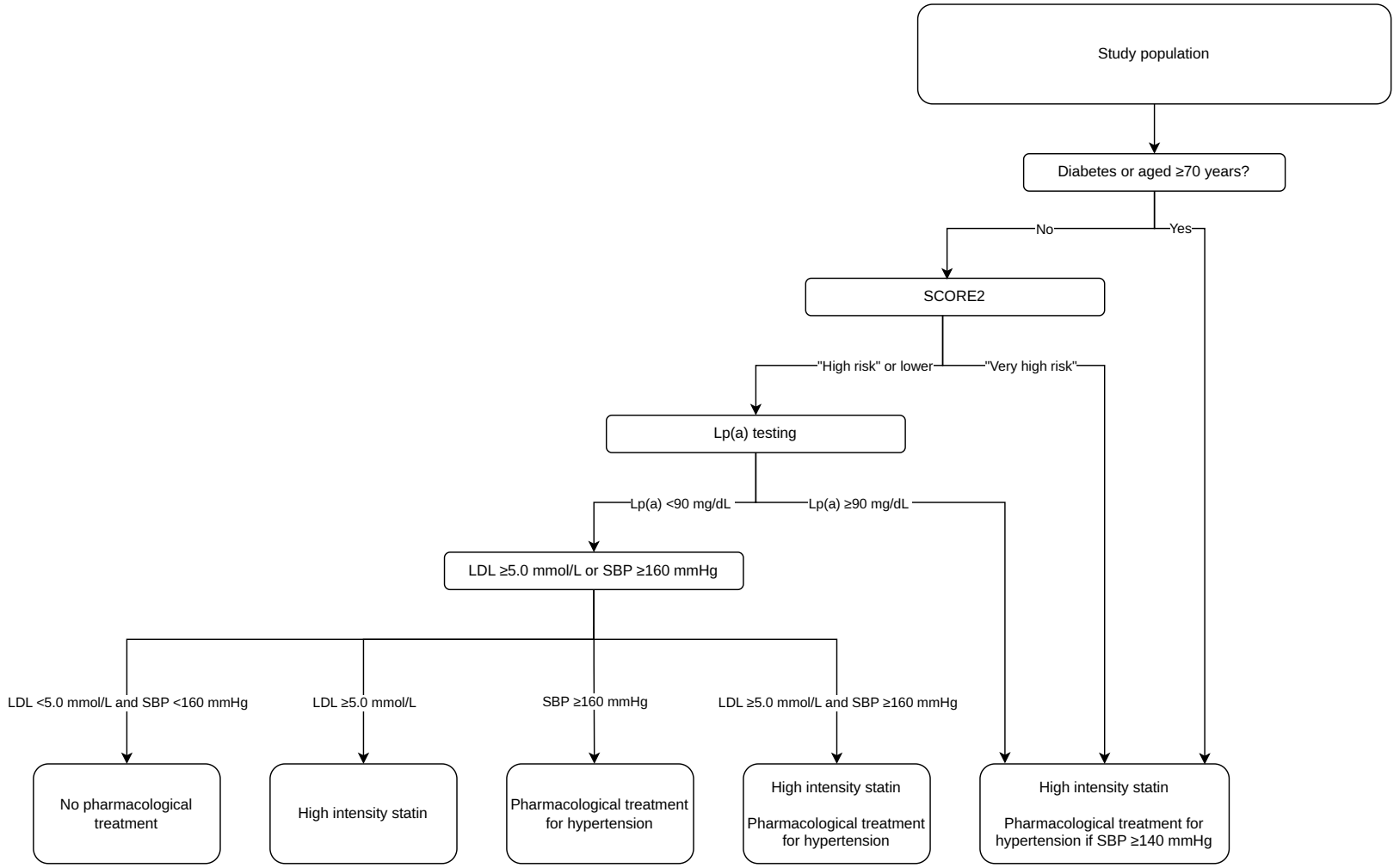


Figure 2.2: Schematic of the intervention scenario in study 1.



2.2 Study 2: Lp(a) testing and treatment of Lp(a)

The populations for this study will be the primary and secondary prevention populations initially aged between 40 and 60 years, followed up until age 85 years (i.e., this study will use a lifetime horizon). We have deliberately not specified a country as this analysis will be repeated from the perspective of multiple countries.

The control scenario will be based on current standard of care using European guidelines on CVD prevention [11]. Therefore, the control scenario in the primary prevention population will be identical to that described in the previous section. For the secondary prevention population, the control situation will also be based on current guidelines for CVD prevention – these people are universally considered very high risk, and so will be treated as per the "very high risk" category described above. The control is outlined in Figure 2.3.

As above, the high intensity statin in this study will be atorvastatin 80 mg and pharmacological therapy for lowering SBP will be based on the SPRINT trial (losartan 100 mg, chlortalidone 25 mg, and amlodipine 10 mg) [15].

The intervention scenario will be as above, except with further testing for Lp(a) among all individuals in the "Very high risk" category (via SCORE2 or from having diabetes or prior CVD). From this testing, individuals with an Lp(a) of at least 192 nmol/L (90 mg/dL) will be treated with Olpasiran [5]. Based on the results of the phase 2 OCEAN(a)-DOSE trial (Olpasiran Trials of Cardiovascular Events and Lipoprotein[a] Reduction–Dose Finding Study), the most effective dosing regimen of Olpasiran was 225 mg every 12 weeks, with similar adverse events to other dosing regimens. This dosing reduced Lp(a) by 97.5% (95%CI: 94.0, 100.0). Lp(a) testing will only occur once at the age the person first reaches threshold for testing. (In the control, no Lp(a) testing or treatment will occur). The intervention is outlined in Figure 2.3.

In scenario analyses, we will change the Lp(a) treatment threshold to 105 nmol/L (50 mg/dL), 149 nmol/L (70 mg/dL) and 236 nmol/L (110 mg/dL).

Figure 2.3: Schematic of the control scenario in study 2.

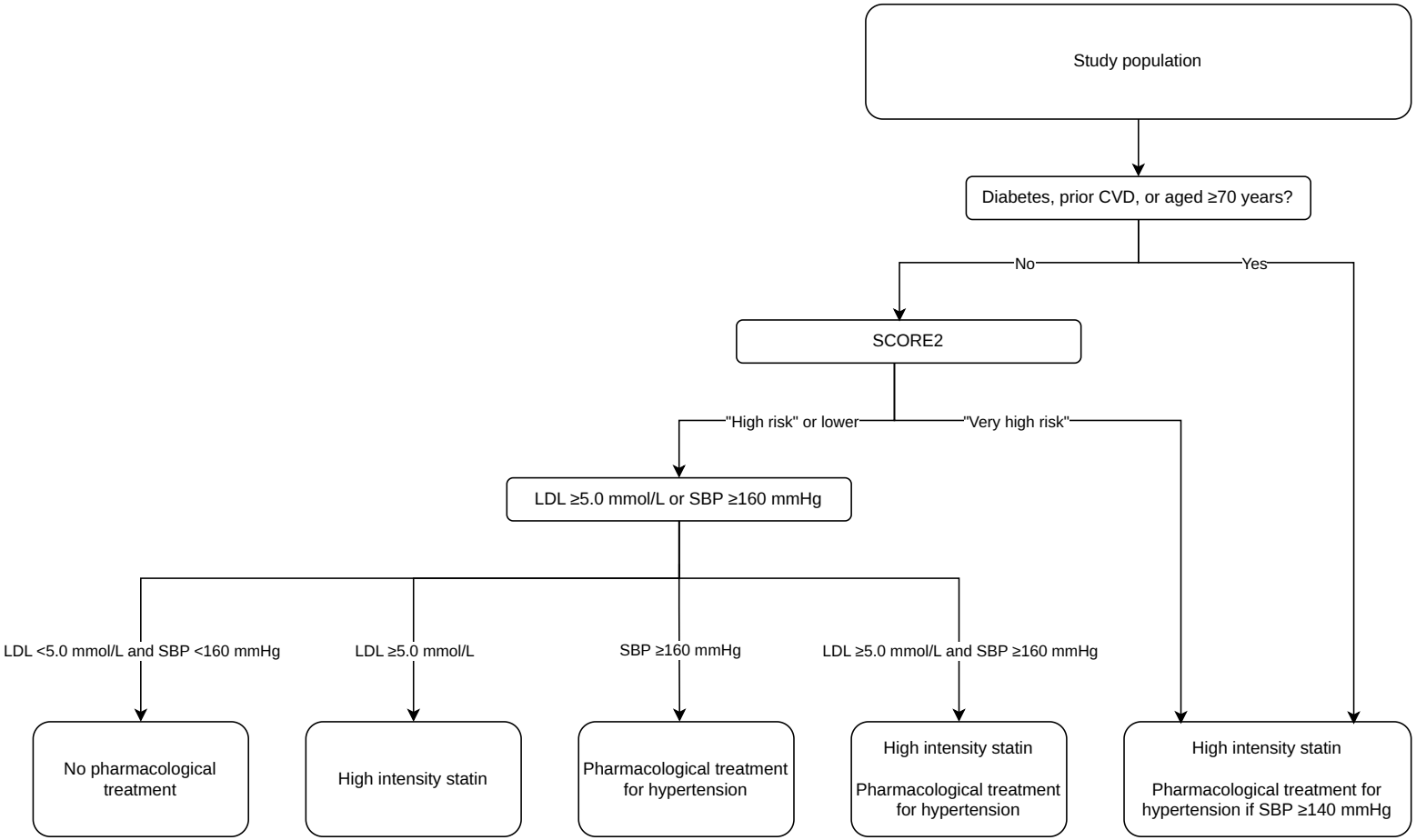
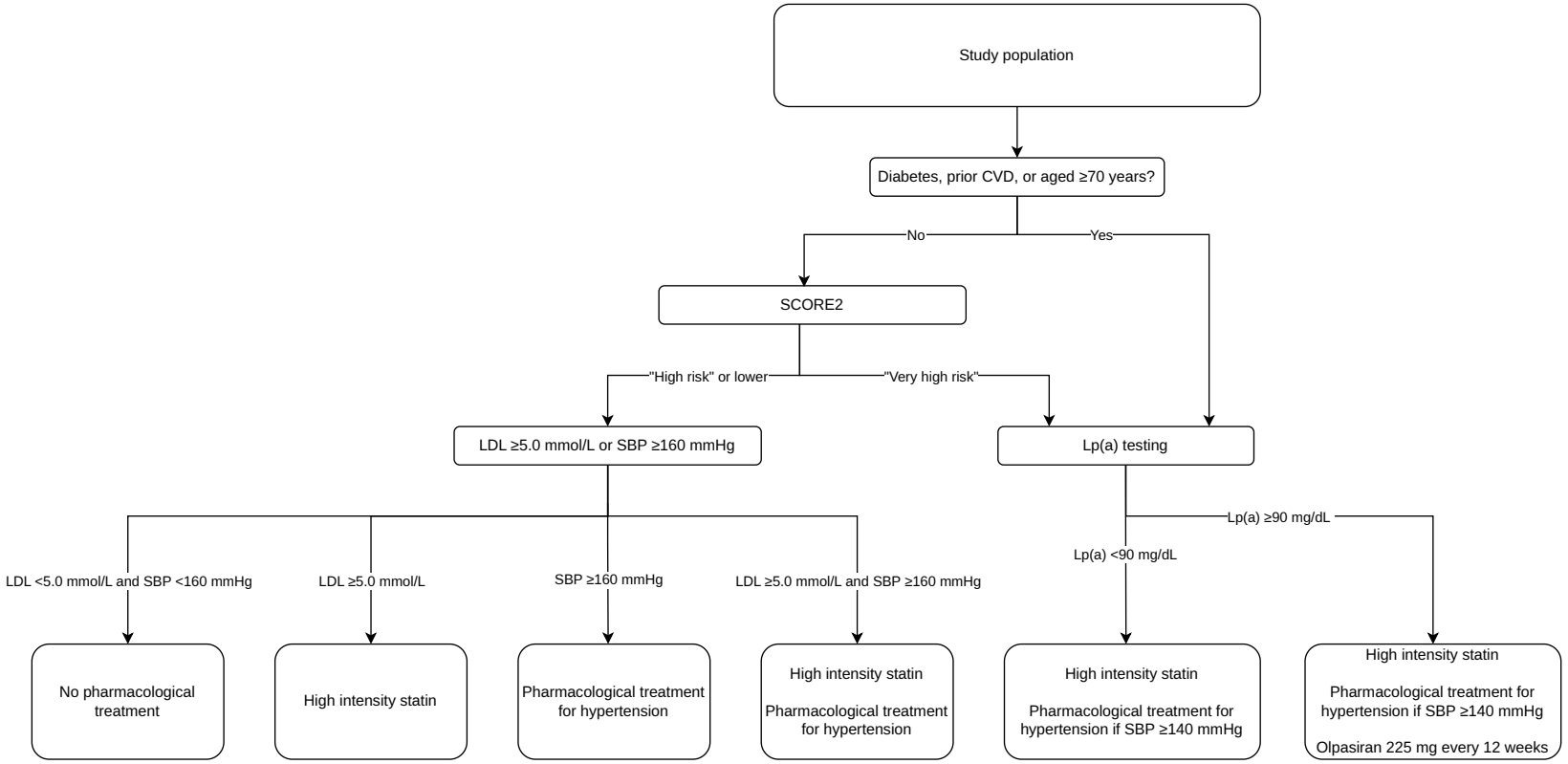


Figure 2.4: Schematic of the intervention scenario in study 2.



3 Model inputs

3.1 Model structure

The structure of the model we will build to complete these studies is shown in Figure 3.1. First note that of all the CVDs, only MI and stroke are included in the model. This is for two reasons – first, MI/CHD and stroke/CBD are responsible for the majority of the burden of CVD [18]; second, the causes of these two conditions have been studied at length, meaning that a model with reliable relationships between risk factors and the risk for these two outcomes can be constructed.

The structure of the model is as follows. The model can run from ages 30 to 84 years (the risk of diabetes and CVD before age 30 is negligible). Individuals can start in any health state. The model runs in 1-year cycles (but cycle length is optional). In each cycle, the following events can occur:

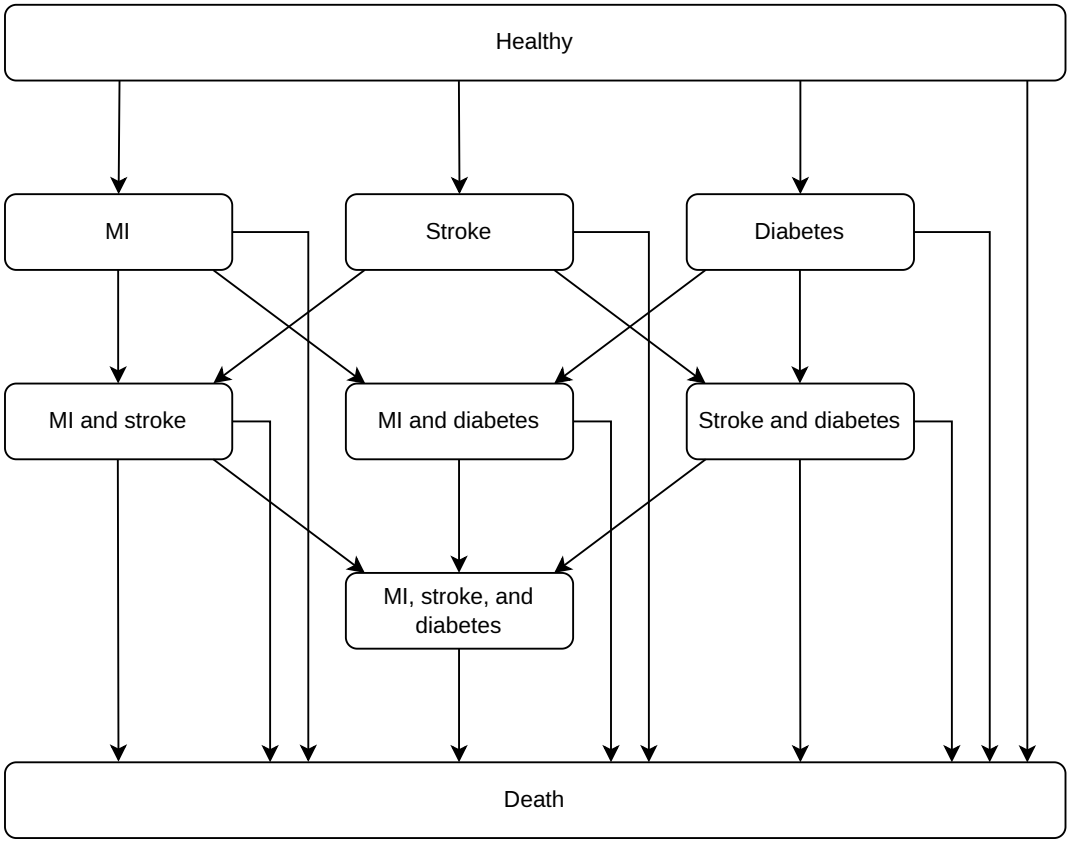
- Individuals in the *healthy* health state (i.e., individuals without diabetes or CVD) can experience the following events: development of diabetes; non-fatal MI; non-fatal stroke; fatal MI or coronary heart disease (CHD) death; fatal stroke or cerebrovascular disease (CBD) death; or non-CVD mortality.
- Individuals who are in the *diabetes* health state can experience the following events: non-fatal MI; non-fatal stroke; fatal MI or CHD death; fatal stroke or CBD death; or non-CVD mortality.
- Individuals in the *MI* or *stroke* health states can experience the following events: development of diabetes; non-fatal MI; non-fatal stroke (i.e., repeat events); fatal MI or CHD death; fatal stroke or CBD death; or non-CVD mortality.
- Individuals in the combination *diabetes* and *MI* and/or *stroke* health states can experience the following events: non-fatal MI; non-fatal stroke (i.e., repeat events); fatal MI or CHD death; fatal stroke or CBD death; or non-CVD mortality.

Note that non-CVD mortality includes a range of causes of death, allowing for competing risks. After each event, individuals move into the appropriate health state (or remain in their present health state), with death being the only absorbing state in this model (Figure 3.1). People can only experience one event per cycle (hence the preference for a short cycle length).

The model includes 7 risk factors for the primary prevention population:

1. Age
2. Sex
3. Low density lipoprotein-cholesterol (LDL-C)
4. Lipoprotein(a) (Lp(a))
5. Systolic blood pressure (SBP)
6. Diabetes
7. Smoking

Figure 3.1: Model structure. Repeat events can occur, but do not involve state transitions. Abbreviation – MI: Myocardial infarction.



LDL-C, SBP, diabetes, and smoking were selected because they are the modifiable causal risk factors in both the Framingham risk score and pooled cohort equation [19, 20]; Lp(a) was included because it is causal for both MI and stroke and is an important risk factor for CVD in a subset of the population with high Lp(a) [1, 2, 3].

The way each of these will be integrated into the model is different and depends on the causal structure of the relationship between each risk factor and each outcome. All the modifiable risk factors (i.e., all except age and sex) will be integrated using evidence from Mendelian randomisation (MR) studies (see below). Age and sex are the risk factors used as a baseline for others to interact with – i.e., they will be used to establish the age and sex-specific incidence of MI, stroke, diabetes, and death, and other risk factors can modify this incidence via their causal effects derived from MR.

For secondary prevention the risk is just related to age and sex, because the risk for people post-

event is very high and less related to the risk factors preceding the event. Nevertheless, Olpasiran will still be assumed to modify the risk of a secondary event, which will also also be incorporated using evidence from MR. However, the mechanism by which this will be done is still be developed and will be included in an updated version of the protocol. Therefore, all discussion of risk factors hereafter refers to the primary prevention population.

The rationale for using MR to derive the causal effects has been explained in detail previously [21]. Briefly, while randomised controlled trials remain the gold standard for causal evidence, there are situations in which trials cannot be performed (such as when the time horizon is too long or cost is too high, as in primary prevention of CVD). Observational studies cannot fill this evidence gap because they provide less reliable estimates of the effects of risk factor modification. Therefore, we have proposed that when a randomised clinical trial that would produce the results required to inform a health economic model is not possible or practical, health economic models should be based on causal effects derived from Mendelian randomisation analyses [21]. This is because Mendelian randomisation, when combined with knowledge of disease pathophysiology and other forms of epidemiological evidence, can allow for causal inference from observational data, which can be used for the design of health economic models that incorporate the long-term, cumulative effects of risk factors (e.g., cholesterol-years or blood pressure-years) on the risk for CVD, an approach we have pioneered [22].

3.2 Risk factors and outcomes

The effects of risk factors on outcomes (in the primary prevention population) in this study will be integrated via Mendelian randomisation, based on a state-of-the-art approach we have previously designed and described [22]. This approach is underpinned by the simulation of risk factor trajectories over the lifetime, which are used to estimate the cumulative exposure to cardiovascular risk factors (e.g., cholesterol-years or blood pressure-years), which in turn, are used to modify cardiovascular risk. This approach is preferable to the use of hazard ratios from clinical trials, which likely don't incorporate the long-term benefits of exposure to lower levels of cumulative, causal risk factors [21].

Indeed, for four of the key cardiovascular risk factors – LDL-C, SBP, Lp(a), and smoking – the effect of these risk factors on cardiovascular risk is proportional to both magnitude of exposure (i.e., absolute levels of the risk factor) and duration of exposure (i.e., time spent exposed to the risk factor) [23, 24, 25]. This is essentially the concept of cholesterol-years, blood pressure-years, or smoking pack-years [26]. Results from Mendelian randomisation can quantify the effect of exposure to the risk factor over the lifetime but cannot provide an age- or exposure-specific estimate of effect. Thus, the key assumption underlying the relationship of LDL-C, Lp(a), and SBP with CVD risk will be that CVD risk is proportional to the time-weighted mean cumulative level of the risk factor at a given age. For smoking, the cumulative impact will be captured using the lifetime smoking index (LSI) [27], which incorporates the non-linear effects of duration of smoking, time since cessation of smoking, and number of cigarettes smoked per day.

Thus, simulating cumulative risk factors values integrates both duration of exposure to, and magnitude of, the risk factors in estimating their effects on CVD risk. In this way, the causal biology of CVD is integrated into the health economic model.

Thus, the first task is to summarise the existing evidence for each risk factor/outcome pair and go through their causal structures. While conducting a complete systematic review for each exposure-outcome pair would be the ideal to go about this, it would be prohibitively time-consuming. Therefore, studies were selected by 1) performing a targeted literature search; 2) identifying key studies that contain information on the exposure-outcome pair; 3) summarising the results; 4) selecting an estimate based on consistency, study robustness (e.g., sufficient sample size, investigation of instrumental variable assumptions is clear, there is no evidence of directional pleiotropy, etc.), and precision of the estimate. In general, we have used inverse-variance weighted estimates/the main estimate presented by the authors.

Finally, before describing the specific associations between risk factors and outcomes, the following are some general notes on the methods used.

First, the effects of the risk factors are incorporated into the model using the following formula:

$$R_a = R \times M^{x-\mu}$$

where R_a is the adjusted age and sex-specific rate of CVD, R is the age- and sex-specific rate of CVD for the whole population, M is the measure of association from a Mendelian Randomisation study, x is the time-weighted mean cumulative level of the risk factor (or LSI) for the individual in the simulation, and μ is the time-weighted mean cumulative level of the risk factor (or LSI) across the population used to derive the age- and sex- specific rate for the whole population. Thus, this formula provides an estimate of the age-specific rate of CVD for an individual based on their exposure to risk factors over the lifetime. To estimate the time-weighted mean cumulative level of

the risk factor (or LSI) at any age, the risk factor trajectory over the lifetime needs to be simulated. The approach to this for each risk factor is outlined in section 6.

For diabetes, incorporating duration of exposure and magnitude of exposure (i.e., to elevated glucose) was deemed impractical for this model; thus, the effect of diabetes on the risk of CVD was incorporated using the following formula:

$$R_{DM} = R_{NoDM} \times M$$

where R_{DM} is the age and sex-specific rate of CVD in people with diabetes, R is the age and sex-specific rate of CVD in people without diabetes, and M is the measure of association from a Mendelian Randomisation study (a rate ratio in this instance).

Second, MR studies usually report odds ratios (ORs), which need to be converted into relative risks (RRs) before use in the model we are building here. Thus, we will convert ORs to RRs using the following formula from Zhang et al [28]:

$$RR = \frac{OR}{(1-P_0)+(P_0 \times OR)}$$

where OR is the OR, RR the relative risk, and P_0 the risk of the outcome in the unexposed group (we use the overall risk when risk in the unexposed group is unavailable). (The confidence intervals (CIs) will be determined by passing the lower and upper bounds of the original CIs through this formula.)

Third, it will not be sufficient to simply use mean cumulative values of risk factors; instead we will need to use time-weighted mean cumulative values. This is because it is highly likely that the effect of risk factors is not constant throughout life and more recent values are more influential. For example, with LDL-C, the most compelling evidence for this is simply the hazard ratio (HR) from a meta-analysis of statin trials – the HR for any major coronary event in trials with durations ranging from 2-5+ years and starting from around age 60 years is 0.77 (95% CI: 0.74, 0.80) per 1 mmol/L reduction in LDL-C [13]. Compare this to the lifetime estimate of 0.46, above, which was derived from a population with a mean age of 65 years. If the effect of LDL-C on coronary heart disease (CHD) risk was constant over the lifetime, there is no way that 65 years of "damage" could be un-done in such a small amount of time and produce such a favourable HR. The exact relationship between time, risk factors, and CVD risk has been established [29].

Fourth, the LSI is defined by the following formula:

$$LSI = (1 - 0.5^{\frac{dur}{18}}) \times (0.5^{\frac{tsc}{18}}) \times ln(int + 1)$$

where LSI is the lifetime smoking index, dur is duration of smoking, tsc is time since cessation of smoking, 18 is the half-life (derived in [27]) that captures the non-linear risk of smoking on health, and int is number of cigarettes per day.

3.2.1 Risk factors for MI

- LDL-C. For LDL-C and MI, the effect is summarised with the OR of 0.46 (95%CI: 0.43, 0.48) per mmol/L reduction in LDL-C over the lifetime [23] (an effect estimate consistent with other major MR studies of this association [30, 31, 32]). Converting into a relative risk yields 0.48 (95%CI: 0.45, 0.50) per mmol/L reduction in LDL-C over the lifetime, or 2.083 (2.000, 2.222) per 1 mmol/L increase in lifetime LDL-C (i.e., flipping the fraction).

- SBP. For SBP and MI, the effect is summarised with the OR of 0.55 (0.52, 0.59) per 10mmHg reduction in SBP [23] (again, consistent with other MR studies of this association [33, 34]). This yields a relative risk of 0.57 (95%CI: 0.54, 0.61) per 10 mmHg reduction in SBP over the lifetime or 1.754 (1.639, 1.862) per 10 mmHg increase in SBP over the lifetime. Mathematically, it's easier to deal in per-unit increases, so this relative risk is betwee summarised as 1.058 (1.051, 1.064) per 1 mmHg increase in SBP.

- Lp(a). For Lp(a) and MI, the effect is summarised with an OR for CHD of 0.912 (0.899, 0.925) per 10-mg/dL lower Lp(a) over the lifetime [1]. This study was selected over the other major Lp(a) study in this space by Burgess et al. [25] as the major population in Burgess et al. does not appear to have representative Lp(a) values [1]. Lamina et al. estimated that each 10-mg/dL lower Lp(a) level is associated with an OR for CHD of 0.912 (0.899, 0.925). Converting to a relative risk per unit increase and flipping the association yields 1.0054 (1.0045, 1.0062) per 1 mg/dL increase in lifetime Lp(a).

- Smoking. For smoking and MI, the effect is summarised by the OR of 1.48 (1.25, 1.75) per 1 σ increase in the LSI [35]. This study was selected over the other major study of this by Larsson et al (1.71 (1.49, 1.98) per 1 σ increase in the LSI [36]) because it is more conservative. The standard deviation (σ) of the LSI in the population under study was 0.694. Converting to a relative risk per unit increase gives us an input to the model of 1.43 (1.22, 1.62) per 1 unit increase in the LSI.

- Diabetes. For diabetes and MI, the effect is summarised with the OR of 1.57 (95%CI: 1.16, 2.05) for people with diabetes vs. without diabetes [37]. This was selected over the other study in this space (OR of 2.43 (1.21, 4.91) [38]) because it was more conservative and precise. Converting to a relative risk yields 1.26 (1.08, 1.40) for diabetes vs. no diabetes.

To integrate these effects into the model, it is necessary to determine how they interact. For LDL-C and SBP, Ference et al. have shown that the effects are additive [23] – i.e., they're independent, so they can be integrated by applying the effects sequentially (applied sequentially, which people refer to as additive, but multiplicative in mathematical terms). Similarly, Lp(a) is independent of LDL-C (and thus very likely also independent from SBP) [25].

For smoking, Levin et al. [35] showed, using multivariable MR, that the genetic liability to smoke was not mediated through other cardiovascular risk factors (including lipids and SBP). Thus, the effect of smoking on CHD can be considered additive to LDL-C, SBP, and LP(a). Similarly, for diabetes, Ross et al. [37] and Yeung et al. [39] have both shown evidence that the effect of diabetes on CHD is similar adjusted the association of diabetes with CHD is independent of other risk factors.

Thus, all risk factors for MI will be incorporated additively (multiplicatively).

3.2.2 Risk factors for stroke

In this model, stroke is split into ischaemic and haemorrhagic (or intracerebral haemorrhage (ICH)) as the two major kinds of stroke, with different aetiologies [40] and thus different risk factors.

First, for ischaemic stroke (IS), the risk factors and their strength of association are as follows.

- LDL-C. For LDL-C and IS, the effect is summarised with the OR of 1.08 (1.03, 1.14) per σ increase in LDL-C over the lifetime [41]. This study was selected as the most precise of the studies in this area ([42, 43]). One standard deviation in this study was 0.87 mmol/L. Thus, converting this estimate to a relative risk for use in this study yields 1.08 (1.03, 1.14) per 1 mmol/L increase in lifetime LDL-C.

- SBP. For SBP and IS, the effect is summarised with the OR of 1.32 (1.20, 1.46) per 10mmHg increase in SBP [44], selected as the most conservative study in the area [44, 45, 46]. Converting to a one-unit relative risk yields 1.027 (1.018, 1.037) per 1 mmHg increase in SBP.

- Lp(a). For Lp(a) and IS, the effect is summarised with the OR of 1.19 (1.12, 1.25) for IS per 50 mg/dL increase in Lp(a) [2], or a relative risk of 1.0035 (1.0023, 1.0045) per mg/dL increase in Lp(a). This was selected as the only study we could find at the time of the search.

- Smoking. For smoking and IS, the effect is summarised with the OR of 1.24 (1.17, 1.33) per 1 σ increase in the LSI [47]. This was selected as the most precise estimate out of the avialable studies in the field [36, 47, 48]. Converted to a relative risk per unit increase yielded 1.33 (1.22, 1.46) per 1 unit increase in the LSI.

- Diabetes. For diabetes and IS, the effect is summarised with the OR of 1.91 (1.22, 2.98) for people with vs. without diabetes [38]. This was selected as the only study we could find that presented the effect of diabetes on IS for people with vs. without diabetes (despite others presenting the association between diabetes risk and IS risk [49, 50, 51]). Converting to a relative risk yields 1.74 (1.19, 2.47).

Second, for ICH, the risk factors and their strength of association are as follows.

- LDL-C. There is no consistent evidence of an association between LDL-C and ICH [52, 53].

- SBP. For SBP and ICH, the effect is summarised with the OR of 1.47 (1.10, 1.95) per 10mmHg increase in SBP [44], which was more conservative when compared with the other study in this area [46]. Converting to a relative risk per unit increase yields 1.039 (1.010, 1.069) per 1 mmHg increase in SBP over the lifetime.

- Lp(a). Larsson et al. found no evidence that Lp(a) influences risk of ICH [2].

- Smoking. Smoking does not appear to cause ICH [36, 47, 48].

- Diabetes. Diabetes does not appear to be causual for ICH [50, 51].

For IS, we will assume a similarity in biology to MI. Therefore, the interactions will be assumed to be the same as for MI (i.e., all additive), and for ICH, there is only one risk factor included in the model, so there is no need to address any interactions.

3.2.3 Risk factors for diabetes

- LDL-C. The relative risk for diabetes will be 0.763 (0.645, 0.909) per 1 mmol/L increase in lifetime LDL-C in a scenario analysis, but excluded from the primary analysis. This is discussed in detail below.
- SBP. There is no clear effect of SBP on type 2 diabetes risk [54, 55, 56].
- Lp(a). There is no clear effect of Lp(a) on type 2 diabetes risk [57, 58, 59, 60, 61].
- Smoking. The effect of smoking on diabetes risk can be summarised using the OR of 1.18 (1.02, 1.37) per 1 σ increase in the LSI [62]. This study was selected as the most conservative out of the available studies that used the LSI to estimate the effect [63, 64, 62]. Converted to a relative risk per unit increase yields 1.21 (1.03, 1.41) per unit increase in the LSI.

While not explicitly addressed by either study, it's probably reasonable to assume the the effects of LDL-C and smoking on diabetes risk are independent.

Because it may be considered controversial, it is worth outlining in detail the relationship between LDL-C and type 2 diabetes risk. Lowering LDL-C via statins increases the risk of diabetes in trials in a dose-dependent manner [65, 66, 67], suggesting a causal relationship between response to statin response/dose and risk of diabetes. Indeed, multiple MR studies have shown an an association between genetically predicted LDL-C and type 2 diabetes risk [68, 32, 69].

While a recent review found the increased risk of type 2 diabetes was mostly among people who were already close to the diagnostic threshold for diabetes at the beginning of the trials [67], it is not surprising that in short-term clinical trials the only people who develop diabetes or experience an increase in blood glucose are people with already elevated risk for type 2 diabetes, as there is a long period prior to onset of diabetes where insulin resistance can worsen without any impact on glycaemia (the pathophysiology of type 2 diabetes is excellently reviewed here: [70]), meaning people at earlier stages of insulin dysfunction could experience a decrement in insulin resistance without any change in blood glucose.

Ultimately, it is plausible that lower or lowering LDL-C causes type 2 diabetes, especially with statins. However, it is unclear by what mechanism this occurs and whether this occurs across all methods of LDL-C lowering. The reasons we argue this should be included as a scenario analysis and not in the primary analysis are: 1) because of the high degree of uncertainty about how this association plays out, and its magnitude; and 2) when included, it is extremely influential on the findings, which shouldn't be the case for an association that is this uncertain. Thus, we have elected to exclude this controversial and uncertain effect from the primary analysis.

The key question for incorporating this effect into the model is whether the effect is related to cumulative LDL or instantaneous LDL. One piece of evidence for the relationship between LDL and CHD being cumulative is the fact that MR effect estimates are always much greater than clinical trials [71]. For diabetes, the effect estimate from a meta-analysis of clinical trials was an OR of 1.09 (95%CI: 1.02, 1.17) for statin use vs. placebo (usually at least 1mmol/L reduction in LDL-C), whereas the MR associations are greater:

- 1.11 (1.04, 1.19) per 10mg/dl (0.25mmol/L) reduction in LDL-C (PCSK9 genetic score) [32]
- 1.13 (1.06, 1.20) per 10mg/dl (0.25mmol/L) reduction in LDL-C (HMGCR genetic score) [32]

- 1.13 (1.00, 1.29) per mmol/L reduction in LDL-C (LDLR) [69]
- 1.15 (0.89, 1.48) per mmol/L reduction in LDL-C (ABCG5/G8) [69]
- 1.19 (1.02, 1.38) per mmol/L reduction in LDL-C (PCSK9) [69]
- 1.39 (1.12, 1.73) per mmol/L reduction in LDL-C (HMGCR) [69]
- 2.42 (1.70, 3.43) per mmol/L reduction in LDL-C (NPC1L1) [69]

(Abbreviations: PCSK9 – proprotein convertase subtilisin/kexin type 9; HMGCR – 3-hydroxy-3-methyl-glutaryl-coenzyme A reductase; LDLR – LDL receptor; NPC1L1 – Niemann-Pick C1-Like 1) I.e., like for CHD, the MR associations for diabetes are greater than those in clinical trials, suggesting the effect of LDL-C on diabetes risk is indeed cumulative.

The effect we will use in the model will be the 1.39 (1.12, 1.73) per 1 mmol/L reduction in LDL-C for the HMGCR from Lotta et al [69]. This was selected because it is the only estimate that is consistent between the two studies above and because there is additional causal evidence from statin clinical trials to support this association (HMGCR is the target of statins). This is the same as a relative risk of 0.763 (0.645, 0.909) per 1 mmol/L increase in lifetime LDL-C.

3.2.4 Risk factors for death

The risk of fatal MI/CHD death and stroke/CBD death will be incorporated in the same way as for the non-fatal events. I.e., the effect of risk factors on these outcomes will be the same for non-fatal and fatal events.

However, even though they are a substantial proportion, CVD deaths are a minority of deaths. This means other deaths are an important competing risk. As previously determined (ref), the causes of death (and their ICD-10 codes) in the UK Biobank study that constitute $\geq$1% of all deaths are as follows.

- Cancers
  - Oesophageal cancer (C15)
  - Stomach cancer (C16)
  - Colorectal cancer (C18-20)
  - Liver cancer (C22)
  - Pancreatic cancer (C25)
  - Lung cancer (C34)
  - Mesothelioma (C45)
  - Breast cancer (C50)
  - Ovarian cancer (C56)
  - Prostate cancer (C61)
  - Kidney cancer (C64)
  - Bladder cancer (C67)
  - Brain cancer (C71)
  - Unspecified site (C80)
  - Multiple myeloma (C90)
  - Monocytic leukaemia (C92)
- CVD
  - Ischaemic heart diseases (I20-25)
  - Cerebrovascular diseases (I60-69)
- Respiratory diseases
  - Pneumonia (J18)
  - Other chronic obstructive pulmonary disease (J44)
  - Other interstitial pulmonary diseases (J84)
- Nervous system diseases
  - Spinal muscular atrophy and related syndromes (G12)

- Parkinson's disease (G20)
- Alzheimer's disease/dementia (F01, F03, G30)

We should briefly note that these causes of death are not necessarily representative of the overall causes of death across life (UK Biobank participants were overwhelmingly aged 40-69 at recruitment), but will be useful for our purposes because they will be the main competing causes of death for CVD in the model timeframe (ages 30-84).

For cancers, there is little evidence that LDL-C, Lp(a), SBP, or diabetes are causal for any major cancers [72, 73, 61, 74, 75]. Of course, smoking is causual for a range of cancers, and the causual effect of smoking on most cancers estimated in MR studies has recently been reviewed by Larsson and BUrgess [62]. In this study, they show associations between the LSI and the following cancers that were on the list of major causes of death above (all ORs per 1 σ increase in the LSI):

- Bladder cancer – OR: 1.91 (1.42, 2.56)
- Colorectal cancer – OR: 1.18 (1.04, 1.33)
- Oesophageal cancer – OR: 2.47 (1.43, 4.27)
- Kidney cancer – OR: 1.50 (1.03, 2.17)
- Lung cancer – OR: 6.39 (4.64, 8.80)
- Ovarian cancer – OR: 1.20 (1.03, 1.41)
- Pancreatic cancer – OR: 1.69 (1.10, 2.58)

which can be converted into RRs per unit increase as follows:

- Bladder cancer – RR: 2.52 (1.66, 3.81)
- Colorectal cancer – RR: 1.24 (1.06, 1.44)
- Oesophageal cancer – RR: 3.67 (1.67, 8.02)
- Kidney cancer – RR: 1.69 (1.04, 3.05)
- Lung cancer – RR: 13.64 (8.85, 21.03)
- Ovarian cancer – RR: 1.27 (1.04, 1.57)
- Pancreatic cancer – RR: 2.13 (1.15, 3.90)

For respiratory diseases, LDL-C, Lp(a), and type 2 diabetes don't appear to be causal for lower respiratory tract infections (Pneumonia) [76, 61, 76, 77]. Whereas SBP and smoking do appear to be [76, 78, 76, 79, 80, 81].

The effect sizes for lower respiratory tract infections are as follows:

- SBP: HR of 1.08 (1.04, 1.13) per 5 mmHg increase in SBP [78], converted into an RR of 1.016 (1.008, 1.025) per 1 mmHg increase in SBP.

- Smoking: OR of 2.83 (2.34, 3.42) per σ increase in the LSI [76], conveted into an RR of 4.03 (3.16, 5.11) per unit increase in the LSI.

For chronic obstructive pulmonary disease (COPD), LDL-C, Lp(a), and SBP do not appear to be associated with COPD [82, 61, 46], and there is scant evidence for diabetes.

Smoking is causal for COPD, although this is so well-known that we can only find two studies that bothered to do the MR. The first, Rosoff et al., found an OR of 2.08 (1.49, 2.88) per unit increase in the LSI. This doesn't seem plausible given how strong the association between the LSI and lung cancer is (OR: 6.39 (4.64, 8.80) [62]) and the fact that the association of smoking with emphysema was stronger than smoking and respiratory cancers in the only other study we found on this [79]. Unfortunately, this other study reports the effect of smoking per extra cigarette smoked per day. Therefore, we will use the OR from lung cancer for emphysema/COPD.

The final category contains mostly "idiopathic pulmonary fibrosis", which, being idiopathic, shouldn't be included in a model that involves integrating disease biology (given that by definition we don't know the biology of this disease).

Finally, the nervous system diseases. "Spinal muscular atrophy and related syndromes" is almost entirely motor neuron disease, for which SBP, Lp(a), smoking, and diabetes all do not appear to be causal [83, 61, 84, 85]. LDL-C, on the other hand, does appear to be causal [86, 87, 85], with the only useable estimate coming from Zeng et al. [86], with an OR of ALS of 1.14 (1.05, 1.24) per 39.0 mg/dL increase in LDL-C (1 mmol/L). This can be converted into an RR of 1.09 (1.03, 1.14) per 1 mmol/L increase in LDL-C.

For Parkinson's disease, LDL-C, Lp(a), SBP, and diabetes don't appear to be causal [88, 89, 90, 91, 92, 93, 61, 94, 92]. However, smoking is protective [95, 96, 97, 98, 62]. The one study that has evaluated this using the LSI is Larsson et al [62], which presents an OR of 0.6 (0.4, 1.01) per σ increase in the LSI, converted to an RR of 0.48 (0.27, 1.01) per unit increase in the LSI.

For Alzheimer's disease/dementia, there is no conclusive evidence that any of LDL-C, Lp(a), SBP, smoking, or diabetes are causal risk factors [99, 89, 100, 101, 2, 102, 103, 104, 105, 106, 107, 62].

3.3 Utilities

The primary analysis will be conducted in Australia and the UK. The utility inputs will be as follows:

- The utility value for people without diabetes, MI, or stroke in Australia, which has age and sex-specific values. This was derived from a cross-sectional study of the Australian general population by McCaffrey et al [108] and modelled to get values for each age in 1-year increments below.
- The chronic utility in people without diabetes, MI, or stroke in the UK, which has age and sex-specific values defined by the equation: $0.9454933 + 0.0256466 \times male - 0.0002213 \times age - 0.0000294 \times age^2$ This was derived from a repeated cross-sectional study of the UK general population by Ara and Brazier [109].
- The chronic utility for people with diabetes, set at 0.785 (0.681, 0.889), as recommended in a review of utility values for type 2 diabetes and its complications [110].
- The chronic utility for people with MI, set at 0.79 (95%CI 0.73, 0.85), derived from a systematic review of utility values in MI [111].
- The chronic utility for people with stroke, set at 0.65 (95%CI 0.63, 0.67), derived from a systematic review of utility values in stroke [112].
- The chronic disutility for people with diabetes and MI, set at -0.055 (-0.067, -0.042), again, as recommended in a review of utility values for type 2 diabetes and its complications [110].
- The chronic disutility for people with diabetes and stroke, set at -0.164 (-0.222, -0.105), again, as recommended in a review of utility values for type 2 diabetes and its complications [110].
- The acute disutility associated with an MI, set at -0.01125 (± 20%). This was derived from a clinical trial that assessed the EQ-5D before hospital discharge and at 6 weeks and 3 months [113]. The difference between discharge EQ-5D score and the EQ-5D score at 3 months was 0.09, so assuming a linear increase means the mean disutility throughout this period would be 0.045 (meaning a loss of 0.01125 for a cycle).
- The acute disutility associated with a stroke, set at -0.03 (± 20%). This was derived from a systematic review of utility values in stroke that examined the utility value prior to discharge from acute hospitalisation or rehabilitation [112]. At this time, the mean utility value was 0.41, which is 0.24 less than the value above (0.65), which is the value from 3 months on. As with MI, assuming a linear increase, the mean value in this period would be 0.12 (a loss of 0.03 over a cycle).

To estimate 1-year sex and age-specific utility values for people without diabetes, MI, or stroke in Australia, we will use beta regression. However, the first step must be to simulate a dataset, as all that McCaffrey and colleagues have published are means and standard deviations for age ranges, as shown in Table 3.1.

```
clear
set obs 2
gen sex = _n-1
```

Table 3.1: Utility values for people without diabetes, MI, or stroke in Australia from McCaffrey et al [108].

Age category	Females		Males	
(years)	N	Mean (SD)	N	Mean (SD)
15-24	226	0.95 (0.08)	238	0.96 (0.07)
25-34	224	0.95 (0.11)	225	0.95 (0.10)
35-44	241	0.91 (0.13)	238	0.93 (0.12)
45-54	253	0.87 (0.16)	247	0.90 (0.16)
55-64	226	0.88 (0.15)	217	0.90 (0.14)
65-74	193	0.87 (0.16)	153	0.87 (0.16)
$\geq$75	122	0.82 (0.15)	104	0.85 (0.16)

```
expand 7
bysort sex : gen age = _n*10+10
gen UT = 0.95 in 1
replace UT = 0.95 in 2
replace UT = 0.91 in 3
replace UT = 0.87 in 4
replace UT = 0.88 in 5
replace UT = 0.87 in 6
replace UT = 0.82 in 7
replace UT = 0.96 in 8
replace UT = 0.95 in 9
replace UT = 0.93 in 10
replace UT = 0.90 in 11
replace UT = 0.90 in 12
replace UT = 0.87 in 13
replace UT = 0.85 in 14
gen SD = 0.08 in 1
replace SD = 0.11 in 2
replace SD = 0.13 in 3
replace SD = 0.16 in 4
replace SD = 0.15 in 5
replace SD = 0.16 in 6
replace SD = 0.15 in 7
replace SD = 0.07 in 8
replace SD = 0.10 in 9
replace SD = 0.12 in 10
replace SD = 0.16 in 11
replace SD = 0.14 in 12
replace SD = 0.16 in 13
replace SD = 0.16 in 14

. list, separator(0)
```

	sex	age	UT	SD
1.	0	20	.95	.08
2.	0	30	.95	.11
3.	0	40	.91	.13
4.	0	50	.87	.16
5.	0	60	.88	.15
6.	0	70	.87	.16
7.	0	80	.82	.15
8.	1	20	.96	.07
9.	1	30	.95	.1
10.	1	40	.93	.12
11.	1	50	.9	.16
12.	1	60	.9	.14
13.	1	70	.87	.16

```
14. |  1   80   .85   .16 |
```

In order to generate random beta distribution values from this data, I first have to calculate the α and β values. The mean (μ) and variance (σ^2) of a beta distribution are given by:

$$\mu = \frac{\alpha}{\alpha+\beta}$$

and

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

So, if solving for α and β, it can be derived that:

$$\alpha = \mu^2(\frac{1-\mu}{\sigma^2} - \frac{1}{\mu})$$

and

$$\beta = \alpha(\frac{1}{\mu} - 1)$$

I can apply this to the values in the dataset:

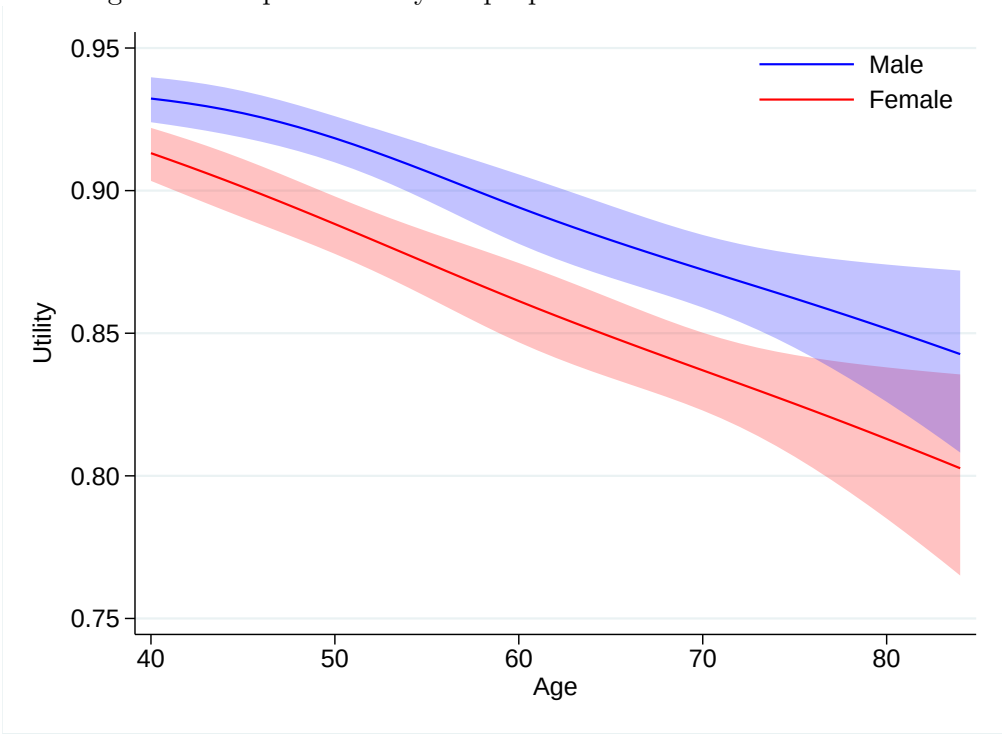
```
gen alph = UT^2*(((1-UT)/(SD^2))-(1/UT))
gen beta = alph*((1/UT)-1)
```

Unfortunately, this gives an unusable value for females aged 25-34 years, so I will make a slight adjustment:

```
replace beta = 0.15 in 2
gen N = 226 in 1
replace N = 224 in 2
replace N = 241 in 3
replace N = 253 in 4
replace N = 226 in 5
replace N = 193 in 6
replace N = 122 in 7
replace N = 238 in 8
replace N = 225 in 9
replace N = 238 in 10
replace N = 247 in 11
replace N = 217 in 12
replace N = 153 in 13
replace N = 104 in 14
expand N
set seed 1836772
gen double A = rbeta(alph,beta)
replace A = 0.9999999999999 if A == 1
mkspline agesp=age, cubic knots(30(15)75)
betareg A agesp* if sex == 0
preserve
clear
set obs 45
gen age = _n + 39.5
mkspline agesp=age, cubic knots(30(15)75)
predict UT1
predict xb, xb
predict errr, stdp
gen UT = invlogit(xb)
gen UTlb = invlogit(xb - 1.96*errr)
gen UTub = invlogit(xb + 1.96*errr)
keep age xb errr UT-UTub
```

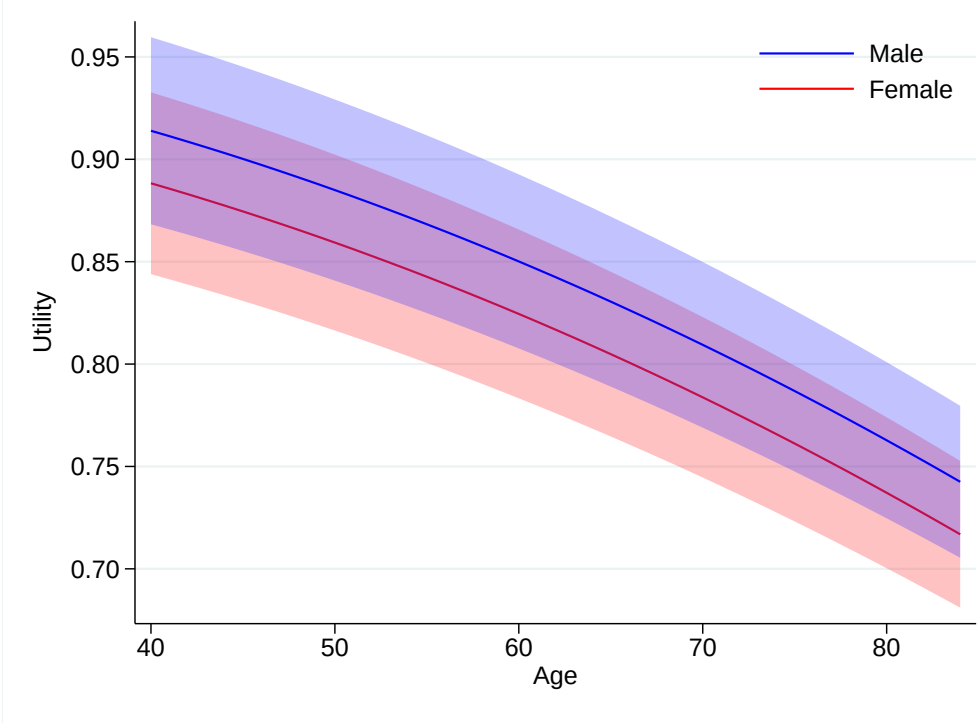
```
gen sex = 0
replace age = age-0.5
save UTF, replace
restore
betareg A agesp* if sex == 1
clear
set obs 45
gen age = _n + 39.5
mkspline agesp=age, cubic knots(30(15)75)
predict xb, xb
predict errr, stdp
gen UT = invlogit(xb)
gen UTlb = invlogit(xb - 1.96*errr)
gen UTub = invlogit(xb + 1.96*errr)
keep age xb errr UT-UTub
gen sex = 1
replace age = age-0.5
save UTM, replace
clear
append using UTF UTM
save UTvals_AU, replace
clear
set obs 2
gen sex = _n-1
expand 45
bysort sex : gen age = _n+39.5
gen UT = 0.9454933+0.0256466*sex-0.0002213*age - 0.0000294*(age^2)
replace age = age-0.5
save UTvals_UK, replace
```

Figure 3.2: Age and sex-specific utility for people without CVD or diabetes in Australia.




```
use UTvals_AU, clear
twoway ///
(rarea UTub UTlb age if sex == 0, col(red%30) fintensity(inten80) lwidth(none)) ///
```

Figure 3.3: Age and sex-specific utility for people without CVD or diabetes in the UK.




```
(line UT age if sex == 0, color(red)) ///
(rarea UTub UTlb age if sex == 1, col(blue%30) fintensity(inten80) lwidth(none)) ///
(line UT age if sex == 1, color(blue)) ///
, legend(order(4 "Male" ///
2 "Female") ///
cols(1) ring(0) position(1) region(lcolor(white) color(none))) ///
graphregion(color(white)) ///
ytitle(Utility) xtitle(Age) ///
ylabel(,angle(0) format(%9.2f)) xlabel(40(10)80)
> c utility for people without CVD or diabetes in Australia.)
use UTvals_UK, clear
gen UTlb = UT-(UT*0.0255*1.96)
gen UTub = UT+(UT*0.0255*1.96)
twoway ///
(rarea UTub UTlb age if sex == 0, col(red%30) fintensity(inten80) lwidth(none)) ///
(line UT age if sex == 0, color(red)) ///
(rarea UTub UTlb age if sex == 1, col(blue%30) fintensity(inten80) lwidth(none)) ///
(line UT age if sex == 1, color(blue)) ///
, legend(order(4 "Male" ///
2 "Female") ///
cols(1) ring(0) position(1) region(lcolor(white) color(none))) ///
graphregion(color(white)) ///
ytitle(Utility) xtitle(Age) ///
ylabel(,angle(0) format(%9.2f)) xlabel(40(10)80)
> fic utility for people without CVD or diabetes in the UK.)
```

3.4 Costs

3.4.1 Healthcare perspective

The cost inputs for Australia (in 2023 AUD) will be as follows:

- The annual chronic cost of diabetes, set at \$3,588 (95% CI: 2,816, 4,539) [114, 115].
- The annual chronic cost of MI, set at \$6,222 (±10%) [116, 117].
- The annual chronic cost of stroke, set at \$4,388 (±10%) [116, 117].
- The annual chronic cost of MI among people with diabetes, set at \$8,870 (6,804, 10,937) [114, 115].
- The annual chronic cost of stroke among people with diabetes, set at \$8,870 (6,804, 10,937) [114, 115].
- The acute cost of non-fatal MI, set at \$14,434 (±10%) [114].
- The acute cost of non-fatal stroke, set a \$15,659 (±10%) [114].
- The acute cost of fatal MI, set at \$3,363 (±10%) (the non-fatal cost multiplied by the proportion of MIs that result in a hospitalisation (23.3%; [29])).
- The acute cost of fatal stroke, set a \$13,154 (±10%) (the non-fatal cost multiplied by the proportion of strokes that result in a hospitalisation (84.0%; [29])).
- Lp(a) test cost, set at \$25, based on current out-of-pocket costs in Australia (Adam Livori, personal communication).
- the annual cost of atorvastatin 80 mg, set at \$200 [118].
- The annual cost of losartan 100 mg, set at \$200 [118].
- The annual cost of chlortalidone 25 mg, set at \$143 [118].
- The annual cost of amlodipine 10 mg, set at \$212 [118].
- The annual cost of Olpasiran, set at set at \$4,360. Because there is no available price for Olpasiran yet, this price was based on the cost of another comparable biologic therapy used in the treatment of CVD – evolocumab – which has an annual price of \$4,360 in Australia [118].

All costs introduced came either directly from the PBS or from studies using costs in 2020 AUD, inflated to 2023 AUD using the Health Price Index of $1.020 \times 1.029 \times 1.042$ [119]. How drug costs were arrived at is shown below – losartan is not widely available in Australia, so costs were sourced from candesartan 32mg.

```
quietly {
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"ATORVASTATIN")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
keep if drug == "ATORVASTATIN Tablet 80 mg (as calcium)"
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Atorvastatin 80 mg " 365.25*r(mean)/30
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"CANDESARTAN")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
keep if drug == "CANDESARTAN Tablet containing candesartan cilexetil 32 mg"
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Candesartan " 365.25*r(mean)/30
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"CHLORTALIDONE")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Chlortalidone " 365.25*r(mean)/50
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"AMLODIPINE")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Amlodipine " 365.25*r(mean)/30
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"EVOLOCUMAB")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
keep if drug == "EVOLOCUMAB Injection 140 mg in 1 mL single use pre-filled pen"
gen unitcost = total_cost/prescriptions
gen injectioncost = unitcost/3 if item_code == "10958R" | item_code == "11977J"
```

```
replace injectioncost = unitcost/2 if item_code == "11484K" | item_code == "11985T"
su injectioncost [aweight=prescriptions]
noisily di "Evolocumab 140mg " 365.25*r(mean)/14
}

. quietly {
Atorvastatin 80 mg 199.58548
Candesartan 200.34522
Chlortalidone 143.14599
Amlodipine 211.92004
Evolocumab 140mg 4359.7639
```

The cost inputs for the UK (in 2023 GBP) will be as follows:

- The annual chronic cost of diabetes, set at £2,546 (95% CI: 2,462, 2,633) for females and £2,170 (2,090, 2,254) for males [120].
- The annual chronic cost of MI, set at £3,304 (3,026, 3,607) for females and £2,917 (2,701, 3,149) for males [120].
- The annual chronic cost of stroke, set at £7,021 (5,852, 8,421) for females and £7,351 (5,923, 8,062) for males [120].
- The annual chronic cost of MI and stroke, set at £14,442 (9,929, 21,004) for females and £12,616 (8,484, 18,756) for males [120].
- The annual chronic cost of MI among people with diabetes, set at £4,511 (3,947, 5,155) for females and £3,917 (3,480, 4,409) for males [120].
- The annual chronic cost of stroke among people with diabetes, set at £10,014 (7,615, 13,167) for females and £10,494 (7,701, 14,300) for males [120].
- The acute cost of non-fatal MI, set at £2,212 (±10%) (see Table 3.2) [121].
- The acute cost of non-fatal stroke, set a £4,626 (±10%) (see Table 3.2) [121].
- The acute cost of fatal MI, set at £515 (±10%) (derived from the NHS cost schedule [121], multiplied by the proportion of MIs that result in a hospitalisation (23.3%; [29])).
- The acute cost of fatal stroke, set a £3,886 (±10%) (derived from the NHS cost schedule [121], multiplied by the proportion of strokes that result in a hospitalisation (84.0%; [29])).
- Lp(a) test cost, set at £40 (assumption).
- The annual cost of atorvastatin 80 mg, set at £18.00 [122].
- The annual cost of losartan 100 mg, set at £15.91 [122].
- The annual cost of chlortalidone 25 mg, set at £12.42 [122] (this cost was based on indapimide 2.5 mg as chlortalidone is very expensive in the UK).
- The annual cost of amlodipine 10 mg, set at £9.91 [122].
- The annual cost of Olpasiran, set at set at £3975. Because there is no available price for Olpasiran yet, this price was based on the cost of the other siRNA therapy used in the treatment of CVD – inclisiran – which has an annual price of £3975 in the UK [123].

All costs (except Lp(a) testing and Olpasiran) introduced came either directly from the NHS drug tariff in June 2023 [122] , the NHS 2022/23 cost schedule [124], or from a Public Health England healthcare costs study [120], which used costs in 2016 GBP, which were inflated to 2023 GBP using the NHS cost inflaction index [125] of: $1.0206 \times 1.0130 \times 1.0159 \times 1.0130 \times 1.0084 \times 1.0170 \times 1.0634$.

Table 3.2: MI and stroke costs from the UK NHS 2022/23 cost schedule

Outcome	Code	Description	Number	Unit cost (£)	Weighted mean (£)
	EB10A	Actual or Suspected Myocardial Infarction, with CC Score 13+	25,923	3,480	
	EB10B	Actual or Suspected Myocardial Infarction, with CC Score 10-12	23,854	2,444	
MI	EB10C	Actual or Suspected Myocardial Infarction, with CC Score 7-9	25,115	1,931	2,212
	EB10D	Actual or Suspected Myocardial Infarction, with CC Score 4-6	26,344	1,654	
	EB10E	Actual or Suspected Myocardial Infarction, with CC Score 0-3	19,668	1,367	
	AA22C	Cerebrovascular Accident, Nervous System Infections or Encephalopathy, with CC Score 14+	6,986	6,114	
	AA22D	Cerebrovascular Accident, Nervous System Infections or Encephalopathy, with CC Score 11-13	3,892	3,603	
	AA22E	Cerebrovascular Accident, Nervous System Infections or Encephalopathy, with CC Score 8-10	4,155	3,067	
	AA22F	Cerebrovascular Accident, Nervous System Infections or Encephalopathy, with CC Score 5-7	4,763	2,467	
	AA22G	Cerebrovascular Accident, Nervous System Infections or Encephalopathy, with CC Score 0-4	7,700	1,709	
	AA23C	Haemorrhagic Cerebrovascular Disorders with CC Score 14+	3,280	6,467	
	AA23D	Haemorrhagic Cerebrovascular Disorders with CC Score 10-13	2,729	3,977	
Stroke	AA23E	Haemorrhagic Cerebrovascular Disorders with CC Score 6-9	3,057	3,127	4,626
	AA23F	Haemorrhagic Cerebrovascular Disorders with CC Score 3-5	2,325	2,438	
	AA23G	Haemorrhagic Cerebrovascular Disorders with CC Score 0-2	1,606	2,106	
	AA35A	Stroke with CC Score 16+	46,703	7,966	
	AA35B	Stroke with CC Score 13-15	32,766	5,543	
	AA35C	Stroke with CC Score 10-12	32,060	4,156	
	AA35D	Stroke with CC Score 7-9	28,766	3,067	
	AA35E	Stroke with CC Score 4-6	22,176	2,395	
	AA35F	Stroke with CC Score 0-3	12,894	1,836	

3.4.2 Societal perspective

We will also estimate the cost-effectiveness in these models using a societal perspective. For this, we will use the human capital approach [126]. This includes costs attributable to lost earnings due to absenteeism (acute and chronic), workforce dropout due to CVD, and loss of future earnings due to premature mortality. All indirect costs can be calculated by multiplying lost work time by the mean earnings, adjusted for workforce participation and unemployment. The inputs for this are as follows:

- Retirement age was set at 67 years in Australia and 66 years in the UK, as the current state pension ages.
- Average annual earnings were set at $73,003 in Australia, derived from the Australia Bureau of Statistics ($1399.70 per week in May 2023; $(1399.10 \times \frac{365.25}{7} = 73,003)$ [127]; and £34,855 in the UK derived from the average weekly total earnings from the Office for National Statistics in the UK (£668 per week in June 2023; $(668 \times \frac{365.25}{7} = 34,855)$ [128].
- Sex and age-specific employment rates (defined as the proportion of people actually in employment, thereby accounting for both workforce participation and unemployment), derived from the Australian Bureau of Statistics for Australia (for June 2023 [129]) with the values shown in Table 3.3; and the Office for National Statistics in the UK (From May-July 2022 [130]), set at:
  - Females, 40-49: 80.6%
  - Females, 50-65: 65.9%
  - Males, 40-49: 90.0%
  - Males, 50-65: 74.3%
- The acute absenteeism period for MI, set at 55 days, which was derived from a cohort study of individuals admitted for acute coronary syndrome or for coronary artery bypass graft surgery (from mean time to return to work of 7.89 weeks for people with acute coronary syndrome) [131].
- The acute absenteeism period for stroke, set at 90 days, which was derived from a retrospective analysis of clinical trial data on stroke survivors [132].
- The chronic absenteeism for CVD (MI or stroke), set at 21.5 days annually, based on a large study in the US [133] (calculated from year 1 in table 5, where people with CVD had 35.0 absent hours and (matched-cohort) people without had 21.4: $(35.0 - 21.4) \times \frac{12}{7.6} = 21.5$).
- The chronic absenteeism for diabetes, set at 6 days annually, based on a systematic review of absenteeism studies; 6 was the mid-point value for the range provided in the study (2-10) [134].
- Workforce participation among people with MI, calculated using the prevalence ratio for not being in the workforce of 1.46 (95%CI: 1.36, 1.55), which was derived from a large Australian cross-sectional study [135].

- Workforce participation among people with stroke, calculated using the prevalence ratio for not being in the workforce of 1.92 (95%CI: 1.80, 2.06), which was derived from the same study as MI [135].
- Workforce participation among people with diabetes, calculated as an absolute reduction in the probability of being in the labour force (but it will be applied to the employment rate), derived from an Australian study [136], set at:
  - Females, 40-49: -3.7%
  - Females, 50-65: -0.2%
  - Males, 40-49: -3.9%
  - Males, 50-65: -11.5%

Table 3.3: Employment, workforce participation, and unemployment by age and sex in Australia. Workforce participation and unemployment were derived from the Australian Bureau of Statistics [129] and employment was calculated based on these numbers.

Sex	Age	Employment	Workforce participation	Unemployment
	15-24	67.3%	72.3%	6.9%
	25-34	79.2%	81.4%	2.7%
	35-44	80.1%	82.5%	2.9%
Females	45-54	80.1%	82.3%	2.7%
	55-66*	62.2%	64.1%	2.9%
	15-24	64.6%	70.4%	8.2%
	25-34	87.0%	89.8%	3.1%
	35-44	89.0%	91.1%	2.3%
Males	45-54	86.5%	88.9%	2.7%
	55-66*	72.5%	74.7%	3.0%

*Uses estimates from 55-64 but extrapolates to 65-66 (as opposed to using 65 years and above)

3.5 Summary

A summary of the inputs to this model are shown in Tables 3.4- 3.6. The discounting rate in the primary analysis will be 5.0% in Australia, as recommended by Australian guidelines [137]; and 3.5% in the UK, as recommended by UK guidelines [138]. The primary outcomes will be the incremental cost-effectiveness ratio (ICER) for each intervention compared to control, which will be compared to the willingness-to-pay thresholds of each country. These are $28,000 per QALY gained in Australia [139] and £20,000 to £30,000 per QALY in the UK [138].

Table 3.4: Model inputs – epidemiological

Input	Value	Distribution	Source
Transition probabilities	See Section 5	Log-normal	UK Biobank and [
Risk factor levels	See Section 6	Normal and log-normal	UK Biobank
LDL-C on incident MI	2.083 (2.000, 2.222) per 1 mmol/L increase in lifetime LDL-C	Log-normal	[23]
Lp(a) on incident MI	1.0054 (1.0045, 1.0062) per 1 mg/dL increase in lifetime Lp(a)	Log-normal	[1]
SBP on incident MI	1.058 (1.051, 1.064) per 1 mmHg increase in lifetime SBP	Log-normal	[23]
Smoking on incident MI	1.43 (1.22, 1.62) per 1 unit increase in the LSI	Log-normal	[35]
Diabetes on incident MI	1.26 (1.08, 1.40) for diabetes vs. no diabetes	Log-normal	[37]
LDL-C on incident IS	1.08 (1.03, 1.14) per 1 mmol/L increase in lifetime LDL-C	Log-normal	[41]
Lp(a) on incident IS	1.0035 (1.0023, 1.0045) per 1 mg/dL increase in lifetime Lp(a)	Log-normal	[2]
SBP on incident IS	1.027 (1.018, 1.037) per 1 mmHg increase in lifetime SBP	Log-normal	[44]
Smoking on incident IS	1.33 (1.22, 1.46) per 1 unit increase in the LSI	Log-normal	[47]
Type 2 diabetes on incident IS	1.74 (1.19, 2.47) for diabetes vs. no diabetes	Log-normal	[38]
SBP on incidence ICH	1.039 (1.010, 1.069) per 1 mmHg increase in lifetime SBP	Log-normal	[44]
LDL-C on incident type 2 diabetes*	0.763 (0.645, 0.909) per 1 mmol/L increase in lifetime LDL-C	Log-normal	[69]
Smoking on incident type 2 diabetes	1.21 (1.03, 1.41) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from bladder cancer	2.52 (1.66, 3.81) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from colorectal cancer	1.24 (1.06, 1.44) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from oesophageal cancer	3.67 (1.67, 8.02) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from kidney cancer	1.69 (1.04, 3.05) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from lung cancer	13.64 (8.85, 21.03) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from ovarian cancer	1.27 (1.04, 1.57) per 1 unit increase in the LSI	Log-normal	[62]
Smoking on death from pancreatic cancer	2.13 (1.15, 3.90) per 1 unit increase in the LSI	Log-normal	[62]
SBP on death from Pneumonia	1.016 (1.008, 1.025) per 1 mmHg increase in lifetime SBP	Log-normal	[78]
Smoking on death from Pneumonia	4.03 (3.16, 5.11) per 1 unit increase in the LSI	Log-normal	[76]
Smoking on death from COPD	13.64 (8.85, 21.03) per 1 unit increase in the LSI	Log-normal	Inferred based on [
LDL-C on death from ALS	1.09 (1.03, 1.14) per 1 mmol/L increase in lifetime LDL-C	Log-normal	[86]
Smoking on death from Parkinson's disease	0.48 (0.27, 1.01) per 1 unit increase in the LSI	Log-normal	[62]
Effect of atorvastatin 80 mg	51.7% (51.2, 52.2) LDL-C reduction	Normal	[14]
Effect of losartan 100 mg, chlortalidone 25 mg, and amlodipine 10 mg	20 mmHg (± 25%) SBP reduction	Normal	[15]
Effect of Olpasiran 225 mg	97.5% (94.0, 100.0) Lp(a) reduction	Beta	[5]

*Scenario analysis only.

Abbreviations: LDL-C – Low-density lipoprotein-cholesterol; MI – Myocardial infarction; SBP – Systolic blood pressure; Lp(a) – Lipoprotein(a); IS – Ischaemic stroke; ICH – Intracerebral haemorrhage; COPD – chronic obstructive pulmonary disease; ALS – amyotrophic lateral sclerosis.

Table 3.5: Model inputs – utilities

Category	Input	Value	Distribution	Source
Utilities	Utility for no CVD or diabetes in Australia	Age and sex-specific (see Figure 3.2)	Beta	[108]
	Utility for no CVD or diabetes in the UK	Age and sex-specific (± 5%; see Figure 3.3)	Beta	[109]
	Chronic utility for diabetes	0.785 (0.681, 0.889)	Beta	[110]
	Chronic utility for MI	0.79 (0.73, 0.85)	Beta	[111]
	Chronic utility for stroke	0.65 (0.63, 0.67)	Beta	[112]
	Chronic utility for MI and stroke	0.65 (0.63, 0.67)	Beta	[112]
	Chronic disutility for MI in diabetes	-0.055 (-0.067, -0.042)	Normal	[110]
	Chronic disutility for stroke in diabetes	-0.164 (-0.222, -0.105)	Normal	[110]
	Chronic disutility for MI and stroke in diabetes	-0.164 (-0.222, -0.105)	Normal	[110]
	Acute disutility for MI	-0.01125 (± 20%)	Normal	[113]
	Acute disutility for stroke	-0.03 (± 20%)	Normal	[112]

Abbreviations: MI – Myocardial infarction.

Table 3.6: Model inputs – costs

Category	Input	Value	Distribution	Source
Direct healthcare costs – Australia	Chronic cost of diabetes	$3,588 (2,816, 4539)	Gamma	[114, 115]
	Chronic cost of MI	$6,222 (±10%)	Gamma	[116, 117]
	Chronic cost of stroke	$4,388 (±10%)	Gamma	[116, 117]
	Chronic cost of MI and stroke	$6,222 (±10%)	Gamma	[116, 117]
	Chronic cost of MI among people with diabetes	$8,870 (6,804, 10,937)	Gamma	[114, 115]
	Chronic cost of stroke among people with diabetes	$8,870 (6,804; 10,937)	Gamma	[114, 115]
	Chronic cost of MI and stroke among people with diabetes	$8,870 (6,804; 10,937)	Gamma	[114, 115]
	Acute cost of non-fatal MI	$14,434 (±10%)	Gamma	[114]
	Acute cost of non-fatal stroke	$15,659 (±10%)	Gamma	[114]
	Acute cost of fatal MI	$3,363 (±10%)	Gamma	[114]
	Acute cost of fatal stroke	$13,154 (±10%)	Gamma	[114]
	Lp(a) test cost	$25	Fixed	Current co
	Annual cost of atorvastatin 80 mg	$200	Fixed	[118]
	Annual cost of losartan 100 mg	$200	Fixed	[118]
	Annual cost of chlortalidone 25 mg	$143	Fixed	[118]
	Annual cost of amlodipine 10 mg	$212	Fixed	[118]
	Annual cost of olpasiran 225 mg	$4,360	Fixed	Assumptio
Direct healthcare costs – UK	Chronic cost of diabetes	£2,546 (2,462, 2,633) in females; £2,170 (2,090, 2,254) in males	Gamma	[120]
	Chronic cost of MI	£3,304 (3,026, 3,607) in females; £2,917 (2,701, 3,149) in males	Gamma	[120]
	Chronic cost of stroke	£7,021 (5,852, 8,421) in females; £7,351 (5,923, 8,062) in males	Gamma	[120]
	Chronic cost of MI and stroke	£14,442 (9,929, 21,004) in females; £12,616 (8,484, 18,756) in males	Gamma	[120]
	Chronic cost of MI in diabetes	£4,511 (3,947, 5,155) in females; £3,917 (3,480, 4,409) in males	Gamma	[120]
	Chronic cost of stroke in diabetes	£10,014 (7,615, 13,167) in females; £10,494 (7,701, 14,300) in males	Gamma	[120]
	Chronic cost of MI and stroke in diabetes	£14,442 (9,929, 21,004) in females; £12,616 (8,484, 18,756) in males	Gamma	[120]
	Acute cost of non-fatal MI	£2,212 (±10%)	Gamma	[121]
	Acute cost of non-fatal stroke	£4,626 (±10%)	Gamma	[121]
	Acute cost of fatal MI	£515 (±10%)	Gamma	[121]
	Acute cost of fatal stroke	£3,886 (±10%)	Gamma	[121]
	Lp(a) test cost	£40	Fixed	Assumptio
	Annual cost of atorvastatin 80 mg	£18.00	Fixed	[122]
	Annual cost of losartan 100 mg	£15.91	Fixed	[122]
	Annual cost of chlortalidone 25 mg	£12.42	Fixed	[122]
	Annual cost of amlodipine 10 mg	£9.91	Fixed	[122]
	Annual cost of olpasiran 225 mg	£3,975	Fixed	Assumptio
Indirect cost inputs	Retirement age – Australia	67 years	Fixed	Policy
	Retirement age – UK	66 years	Fixed	Policy
	Average annual earnings – Australia	$73,003	Fixed	[127]
	Average annual earnings – UK	£34,855	Fixed	[128]
	Employment rates	Age and sex-specific	Fixed	[129, 130]
	Acute absenteeism for MI	55 days (±20%)	Normal	[131]
	Acute absenteeism for stroke	90 days (±20%)	Normal	[132]
	Chronic absenteeism for CVD	21.5 days annually (±20%)	Normal	[141]
	Chronic absenteeism for diabetes	6 days annually (2-10)	Normal	[134]
	Workforce non-participation ratio for MI	1.46 (1.36, 1.55)	Log-normal	[135]
	Workforce non-participation ratio for stroke	1.92 (1.80, 2.06)	Log-normal	[135]
	Workforce non-participation reduction for diabetes	Age and sex-specific	Fixed	[136]

Abbreviations: MI – Myocardial infarction.

4 UK Biobank data cleaning

Before estimation of the transition probabilities or risk factor trajectories, data cleaning is required.

4.1 Variables

The UK Biobank main dataset variables of interest for this study are:

- Participant ID (UDI: eid)
- Sex (UDI: 31)
- Date of assessment (53)
- UK Biobank assessment centre (54)
- Year of birth (34)
- Month of birth (52)
- Date and causes of death (40000–40002)
- Date and source of first myocardial infarction (MI) (42000–42001)
- Date and source of first stroke (42006–42007)
- Date and source of first IS (42008–42009)
- Date and source of first ICH (42010–42011)
- Diabetes diagnosed by doctor (2443)
- Age diabetes diagnosed (2976)
- Gestational diabetes only (4041)
- Medication for cholesterol, blood pressure, diabetes (6153 (females) & 6177 (males))
- Treatment/medications (20003)
- LDL-C (30780)
- Lp(a) and reportability (30790 & 30796)
- SBP, manual reading (93)
- SBP, automatic reading (4079)
- Current tobacco smoking (1239)
- Past tobacco smoking (1249)
- Age started smoking in current smokers (3436)
- Age started smoking in former smokers (2867)

- Age stopped smoking (2897)
- Number of cigarettes currently smoked daily (3456)
- Number of cigarettes previously smoked daily (2887)
- Type of tobacco currently smoked (3446)
- Type of tobacco previously smoked (2877)
- Smoking compared to 10 years previous (3506)
- Pack years of smoking (20161)

(Note that we include results from the first assessment only – follow-up is incomplete for all other assessments.)

From HESIN (hospital episode statistics for inpatients), we need the following fields:

- eid
- Instance index (unique identifier for a record)
- Inpatient record origin (England, Wales, or Scotland)
- Inpatient record format
- Episode start date
- Episode end date
- Episode status (has the episode finished)
- Episode order
- Spell index
- Spell sequence
- Date of admission to hospital
- Sources of admission to hospital (recoded)
- Date of discharge from hospital
- Destination on discharge from hospital (recoded)

And from HESIN_DIAG, we need:

- eid
- Instance index (correpsonds to the unique identifier for a record in HESIN)
- Array index (unique identifier for a diagnosis)

- Classification of diagnosis (i.e., primary, secondary, or external)
- Diagnoses (ICD-10)

```
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1:1)
save DS_1, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(7:7)
rename v1 sex
save DS_2, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(74:81)
rename (v1 v5) (da1 ac1)
keep da1 ac1
save DS_3, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(8:8)
rename v1 yob
save DS_4, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(73:73)
rename v1 mob
save DS_5, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11680:11684)
rename (v1 v2 v3 v4 v5) (dod1 dod2 ucod ucod2 ccod1)
save DS_6, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(13731:13732)
rename (v1 v2) (midate misource)
save DS_7, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(13737:13742)
rename (v1 v2 v3 v4 v5 v6) (stdate stsource isdate issource ichdate ichsource)
save DS_8, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1030:1030)
rename (v1) (dmdoc1)
save DS_9, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1178:1178)
rename (v1) (agedmdx1)
save DS_10, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1520:1520)
rename (v1) (gdmonly1)
save DS_11, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(6070:6073)
forval i = 1/4 {
rename v`i´ fmeduse`i´
}
save DS_12, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(6270:6272)
forval i = 1/3 {
rename v`i´ mmeduse`i´
}
save DS_13, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(6641:6688)
forval i = 1/48 {
rename v`i´ med`i´
}
save DS_14, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11510:11511)
rename (v1 v2) (ldl1 ldl2)
save DS_15, replace

*Needs to be updated with new field

import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11524:11525)
rename (v1 v2) (lpa1 lpa2)
save DS_16, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11536:11537)
rename (v1 v2) (lpar1 lpar2)
save DS_17, replace

*Needs to be updated with new field

import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(376:377)
rename (v1 v2) (msbp00 msbp01)
```

```
save DS_18, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1540:1541)
rename (v1 v2) (asbp00 asbp01)
save DS_19, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(631:631)
rename (v1) (cs1)
save DS_20, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(635:635)
rename (v1) (ps1)
save DS_21, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1349:1349)
rename v1 agestartsmokecur1
save DS_22, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1142:1142)
rename v1 agestartsmokeprev1
save DS_23, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1154:1154)
rename v1 agestopsmoke1
save DS_24, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1357:1357)
rename v1 ncigcur1
save DS_25, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1150:1150)
rename v1 ncigprev1
save DS_26, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1353:1353)
rename v1 typetobcur1
save DS_27, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1146:1146)
rename v1 typetobprev1
save DS_28, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(1365:1365)
rename v1 delta10smoke1
save DS_29, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(8677:8677)
rename v1 packyear1
save DS_30, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11470:11470)
destring v1, replace
gen a1c = 0.0915*v1+2.15
drop v1
save DS_31, replace
import delimited "/home/jed/Documents/ukb669154.csv", clear varnames(1) colrange(11482:11483)
rename (v1 v2) (hdl1 hdl2)
save DS_32, replace
use DS_1, clear
forval i = 2/32 {
merge 1:1 _n using DS_`i´, nogen
}
save DS_comb, replace
import delimited "/home/jed/Documents/w88775_2023-04-25.csv", clear
rename v1 eid
save eiddrop, replace
use DS_comb, clear
merge 1:1 eid using eiddrop, keep(1) nogen
save DS, replace
import delimited "/home/jed/Documents/hesin.txt", varnames(1) clear
keep eid ins_index dsource source epistart epiend epistat ///
epiorder spell_index spell_seq admidate admisorc_uni disdate disdest_uni
save HESIN, replace
import delimited "/home/jed/Documents/hesin_diag.txt", varnames(1) clear
keep eid ins_index arr_index level diag_icd10
save HESIN_DIAG, replace
```

4.2 Main dataset

I have previously checked and cleaned UK Biobank data in detail (e.g., [29], so will not repeat the checks here.

```
use DS, clear
destring sex, replace
label variable sex
gen da = date(da1,"YMD")
format da %td
gen ac = "England"
replace ac = "Wales" if ac1 == "11003" | ac1 == "11022" | ac1 == "11023"
replace ac = "Scotland" if ac1 == "11004" | ac1 == "11005"
set seed 19287
gen db = runiformint(1,28) if mob == "2"
replace db = runiformint(1,31) if mob == "1" | mob == "3" | mob == "5" | mob == "7" | mob == "8" | m
> ob == "10" | mob == "12"
replace db = runiformint(1,30) if mob == "4" | mob == "6" | mob == "9" | mob == "11"
tostring db, replace
replace db = "0" + db if length(db)==1
replace mob = "0" + mob if length(mob)==1
gen dbb = db+mob+yob
gen dob = date(dbb,"DMY")
format dob %td
drop if dob==.
gen dod = date(dod1,"YMD")
format dod %td
label variable ucod
gen chddeath = 1 if inrange(ucod,"I21","I2599")
gen isdeath = 1 if substr(ucod,1,3)=="I63" | substr(ucod,1,3)=="I64"
gen ichdeath = 1 if substr(ucod,1,3)=="I61"
gen oescdeath = 1 if substr(ucod,1,3)=="C15"
gen colcdeath = 1 if inrange(ucod,"C18","C209")
gen pancdeath = 1 if substr(ucod,1,3)=="C25"
gen luncdeath = 1 if substr(ucod,1,3)=="C34"
gen ovacdeath = 1 if substr(ucod,1,3)=="C56"
gen kidcdeath = 1 if substr(ucod,1,3)=="C64"
gen blacdeath = 1 if substr(ucod,1,3)=="C67"
gen pneudeath = 1 if substr(ucod,1,3)=="J18"
gen copddeath = 1 if substr(ucod,1,3)=="J44"
gen alsdeath = 1 if ucod=="G122"
gen pddeath = 1 if substr(ucod,1,3)=="G20"
gen otherdeath = 1 if dod!=.
foreach var of varlist chddeath-pddeath {
replace otherdeath =. if `var´ == 1
}
gen mid = date(midate, "YMD")
gen std = date(stdate, "YMD")
gen isd = date(isdate, "YMD")
gen ichd = date(ichdate, "YMD")
format mid-ichd %td
save midclean_1, replace
use HESIN_DIAG, clear
gen MI = 1 if inrange(diag,"I21","I238") | diag == "I241" | diag == "I252"
gen ST = 1 if inrange(diag,"I60","I619") | inrange(diag,"I63","I649")
gen IS = 1 if inrange(diag,"I63","I6499")
gen ICH = 1 if inrange(diag,"I61","I6199")
keep if MI == 1 | ST == 1 | IS == 1 | ICH == 1
save HESINevents, replace
use HESIN, clear
merge 1:m eid ins_index using HESINevents
keep if _merge == 3
drop _merge
gen admdate = date(epistart,"DMY")
replace admdate = date(admidate,"DMY") if admdate==.
format admdate %td
save HESINeventss, replace
```

```
foreach i in MI ST IS ICH {
use HESINeventss, clear
keep if `i´ == 1
bysort eid (admdate level) : keep if _n == 1
keep eid diag level `i´ admdate
save First`i´HESIN, replace
}
use midclean_1, clear
merge 1:1 eid using FirstMIHESIN, keep(1 3) nogen
replace mid = admdate if admdate < mid
drop level-admdate
merge 1:1 eid using FirstSTHESIN, keep(1 3) nogen
replace std = admdate if admdate < std
drop level-admdate
merge 1:1 eid using FirstISHESIN, keep(1 3) nogen
replace isd = admdate if admdate < isd
drop level-admdate
merge 1:1 eid using FirstICHHESIN, keep(1 3) nogen
replace ichd = admdate if admdate < ichd
drop level-admdate
save midclean_2, replace
*copy "/home/jed/Documents/WuESM1.dta" WuESM1.dta
*ESM1 from Wu et al. https://pubmed.ncbi.nlm.nih.gov/31015401/
use WuESM1, clear
forval i = 1/10 {
gen njm = _n
gen delim = strpos(medic,"|")
expand 2 if delim!=0
bysort njm : replace medic = substr(medic,1,delim-2) if _n==1 & _n!=_N
bysort njm : replace medic = substr(medic,delim+1,length(medic)) if _n > 1
drop njm delim
}
save WuESM1long, replace
use WuESM1long, clear
gen DM = 1 if substr(medic,1,3)=="A10"
keep if DM == 1
keep codinga DM
save DMdrugcodes, replace
use WuESM1long, clear
gen LL = 1 if substr(medic,1,5)=="C10AA" | medica == "C10AX09"
keep if LL == 1
keep codinga LL
save LLdrugcodes, replace
use WuESM1long, clear
gen HT = 1 if substr(medic,1,3)=="C03" | substr(medic,1,3)=="C07" | substr(medic,1,3)=="C08" | subst
> r(medic,1,3)=="C09"
keep if HT == 1
keep codinga HT
save HTdrugcodes, replace
use DS, clear
keep eid med1-med48
gen med = med1
forval i = 2/48 {
bysort eid : gen njm = _n
bysort eid : gen Njm = _N
expand 2 if med`i´!="" & njm == Njm
bysort eid (njm) : replace med = med`i´ if med`i´ !="" & _n == _N
drop njm Njm
}
keep eid med
destring med, replace
rename med codinga
foreach i in DM LL HT {
merge m:1 codinga using `i´drugcodes
drop if _merge == 2
drop _merge
}
collapse (sum) DM-HT, by(eid)
```

```
replace DM = 1 if DM > 1
replace LL = 1 if LL > 1
save codedmeds, replace
use midclean_2, clear
drop med1-med48
merge 1:1 eid using codedmeds, keep(1 3) nogen
gen DMA = 1 if dmdoc1 == "1"
replace DMA = . if gdmonly1 == "1"
gen DMC = 1 if (DMA == 1 & DM == 1) | (a1c >=6.5 & a1c!=.) | (DMA == 1 & a1c==.)
*Diabetes if doctor told you confirmed by drugs/a1c, or a1c alone (missing a1c, defer to self report
> )
destring agedmdx1, replace
gen dmdxdate = round(dob+(agedmdx1*365.25),1) if DMA == 1 & agedmdx1 > 0
replace dmdxdate = da if DMA == 1 & dmdxdate==.
format dmdxdate %td
gen agedmdx = (dmdxdate-dob)/365.25
gen T1Dlikely = 1 if agedmdx < 20
destring ldl1 ldl2, replace
destring hdl1 hdl2, replace
destring lpa1 lpa2, replace
gen ldl = ldl1
replace ldl = ldl2 if ldl1==.
gen hdl = hdl1
replace hdl = hdl2 if hdl1==.
gen lpa = lpa1
replace lpa = lpa2 if lpa1==.
*Needs to be updated with new field
*Convert from nmol/L to mg/dL using https://pubmed.ncbi.nlm.nih.gov/30608559/
replace lpa = (lpa+3.83)/2.18
rename lpar1 lpar
label variable lpar
rename LL LLT
label variable LLT
gen sbp1 = asbp00
replace sbp1 = msbp00 if missing(asbp00)
gen sbp2 = asbp01
replace sbp2 = msbp01 if missing(asbp01)
destring sbp1 sbp2, replace
gen sbp = (sbp1+sbp2)/2
replace sbp = sbp1 if sbp2==.
replace sbp = sbp2 if sbp1==.
rename HT AHT
label variable AHT
gen eversmoke = 1 if cs1 == "1" | ps1 == "1"
recode eversmoke .=0
destring agestartsmokeprev1 agestartsmokecur1, replace
gen agess = min(agestartsmokeprev1,agestartsmokecur1)
su(agess) if agess > 0
replace agess = round(r(mean),1) if agess < 0
gen smkdate = round(dob+(agess*365.25),1)
format smkdate %td
destring agestopsmoke1, replace
rename agestopsmoke1 agests
replace agests= -2 if agests < agess & agests > 0
gen dursmk = agests-agess if agests > 0
replace agests = round(agess+r(mean),1) if agests < 0
drop dursmk
gen quitdate = round(dob+(agests*365.25),1)
replace agests = round((da-dob)/365.25,1) if quitdate>da & quitdate!=.
replace quitdate = da if quitdate>da & quitdate!=.
format quitdate %td
gen dursmkq = agests-agess
gen dursmkc = round((da-smkdate)/365.25,1) if quitdate==.
gen dursmk = min(dursmkq,dursmkc)
destring ncigprev1 ncigcur1, replace
gen ncig = min(ncigprev1,ncigcur1)
replace ncig=0.5 if ncig == -10
su(ncig)
```

```
replace ncig = round(r(mean),1) if eversmoke==1 & (ncig==. | ncig < 0)
gen tsc = 0 if quitdate==. & eversmoke==1
replace tsc = (da-quitdate)/365.25 if quitdate!=.
gen LSI = 0
replace LSI = (1-(0.5^(dursmk/18)))*(0.5^(tsc/18))*ln(ncig+1) if eversmoke==1
su LSI, detail
drop if dob==.
drop if T1Dlikely==1
***
**To be removed when get the new variable
set seed 29487
replace lpa = runiform(0.01,3.49) if lpar == "4"
replace lpa = rexponential(20)+88.5 if lpar == "5"
*****
save fullset, replace
drop if mid < da
drop if std < da
keep eid sex da ac dob dod ucod chddeath-otherdeath DMC ldl hdl LLT lpa lpar sbp AHT LSI dursmk tsc
> ncig
order eid sex da ac dob dod ucod chddeath-otherdeath DMC ldl LLT hdl lpa lpar sbp AHT LSI dursmk tsc
>  ncig
save mainset, replace
erase midclean_1.dta
erase midclean_2.dta
```

4.3 HESIN

```
use HESIN_DIAG, clear
keep if level == 1
gen MI = 1 if inrange(diag,"I21","I229")
gen IS = 1 if inrange(diag,"I63","I6499")
gen ICH = 1 if inrange(diag,"I61","I6199")
keep if MI == 1 | IS == 1 | ICH == 1
save hesinmiisich, replace
use HESIN, clear
merge 1:m eid ins_index using hesinmiisich, nogen
gen event=min(MI,IS,ICH)
bysort eid : egen A = min(event)
keep if A == 1
drop A
gen epist = date(epistart,"DMY")
gen epien = date(epiend,"DMY")
format epist epien %td
sort eid epist epien
bysort eid (epist epien) : gen dist = epist-epien[_n-1]
replace epist = date(admidate,"DMY") if epist==.
replace epist = epien if missing(epist)
drop if epist==.
replace epien = date(disdate,"DMY") if epien==.
replace epien = epist if epien==.
gen admmode = 1
replace admmode = 0 if inrange(admisorc,1000,2002) | inrange(admisorc,4000,4001) ///
 | inrange(admisorc,7000,7003) | (admisorc >= 10000 & admisorc!=11000)
gen sepmode = 1
replace sepmode = 0 if inrange(disdest,1000,2002) | inrange(disdest,4000,4001) ///
 | inrange(disdest,7000,7003) | (disdest >= 10000 & disdest!=11000)
replace sepmode = 2 if disdest==11001
quietly {
forval i = 1/100 {
bysort eid (epist epien sepmode) : gen A = 1 if epist < epien[_n-1] & epien[_n-1]!=.
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epist = epist[_n+1] if epist[_n+1] < epist & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==1 & epien[_n+1]
>  > epien & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==2 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if epien[_n+1] > epien & A[_n+1]==1
drop if A == 1
drop A
}
forval i = 1/5 {
bysort eid (epist epien sepmode) : gen A = 1 if epist == epist[_n-1] & epien == epien[_n-1]
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]!=0 & A[_n+1]==1
drop if A == 1
drop A
}
}
quietly {
forval i = 1/100 {
bysort eid (epist epien sepmode) : gen A = 1 if epist < epien[_n-1] & epien[_n-1]!=.
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epist = epist[_n+1] if epist[_n+1] < epist & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==1 & epien[_n+1]
```

```
>  > epien & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==2 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if epien[_n+1] > epien & A[_n+1]==1
drop if A == 1
drop A
}
forval i = 1/5 {
bysort eid (epist epien sepmode) : gen A = 1 if epist == epist[_n-1] & epien == epien[_n-1]
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]!=0 & A[_n+1]==1
drop if A == 1
drop A
}
}
bysort eid (epist epien sepmode) : gen ptr = 1 if admmode==1 | sepmode[_n-1]==1
bysort eid (epist epien sepmode) : gen transferdist = epist-epien[_n-1]
gen tr = 1 if ptr == 1 & inrange(transferdist,0,1)
bysort eid (epist epien sepmode) : replace tr = 1 if transferdist==0 & (MI==1 | MI[_n-1]==1 | IS ==
> 1 | IS[_n-1]==1 | ICH == 1 | ICH[_n-1]==1)
bysort eid (epist epien sepmode) : replace tr =. if tr[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if tr[_n+1]==1
bysort eid (epist epien sepmode) : drop if tr == 1 & tr[_n-1]==.
drop ptr tr transferdist
quietly {
forval i = 1/100 {
bysort eid (epist epien sepmode) : gen ptr = 1 if admmode==1 | sepmode[_n-1]==1
bysort eid (epist epien sepmode) : gen transferdist = epist-epien[_n-1]
gen tr = 1 if ptr == 1 & inrange(transferdist,0,1)
bysort eid (epist epien sepmode) : replace tr = 1 if transferdist==0 & (MI==1 | MI[_n-1]==1 | IS ==
> 1 | IS[_n-1]==1 | ICH == 1 | ICH[_n-1]==1)
bysort eid (epist epien sepmode) : replace tr =. if tr[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if tr[_n+1]==1
bysort eid (epist epien sepmode) : drop if tr == 1 & tr[_n-1]==.
drop ptr tr transferdist
bysort eid (epist epien sepmode) : gen A = 1 if epist < epien[_n-1] & epien[_n-1]!=.
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epist = epist[_n+1] if epist[_n+1] < epist & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==1 & epien[_n+1]
>  > epien & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==2 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if epien[_n+1] > epien & A[_n+1]==1
drop if A == 1
drop A
}
}
preserve
keep if MI == 1
keep eid MI epist epien
save AllMI, replace
restore
preserve
keep if IS == 1
keep eid IS epist epien
save AllIS, replace
restore
keep if ICH == 1
```

```
keep eid ICH epist epien
save AllICH, replace
use HESIN_DIAG, clear
gen MI = 1 if inrange(diag,"I21","I238") | diag == "I241" | diag == "I252"
keep if MI == 1
bysort eid ins_index : keep if _n == 1
keep eid ins_index MI
save hesinmi, replace
use HESIN_DIAG, clear
gen IS = 1 if inrange(diag,"I63","I6499")
keep if IS == 1
bysort eid ins_index : keep if _n == 1
keep eid ins_index IS
save hesinis, replace
use HESIN_DIAG, clear
gen ICH = 1 if inrange(diag,"I61","I6199")
keep if ICH == 1
bysort eid ins_index : keep if _n == 1
keep eid ins_index ICH
save hesinich, replace
use hesinmi, clear
merge 1:1 eid ins_index using hesinis, nogen
merge 1:1 eid ins_index using hesinich, nogen
sort eid ins
save hesinmiisichfirst, replace
use HESIN, clear
merge 1:m eid ins_index using hesinmiisichfirst, nogen
gen event=min(MI,IS,ICH)
bysort eid : egen A = min(event)
keep if A == 1
drop A
gen epist = date(epistart,"DMY")
gen epien = date(epiend,"DMY")
format epist epien %td
sort eid epist epien
replace epist = date(admidate,"DMY") if epist==.
replace epist = epien if missing(epist)
drop if epist==.
replace epien = date(disdate,"DMY") if epien==.
replace epien = epist if epien==.
gen admmode = 1
replace admmode = 0 if inrange(admisorc,1000,2002) | inrange(admisorc,4000,4001) ///
 | inrange(admisorc,7000,7003) | (admisorc >= 10000 & admisorc!=11000)
gen sepmode = 1
replace sepmode = 0 if inrange(disdest,1000,2002) | inrange(disdest,4000,4001) ///
 | inrange(disdest,7000,7003) | (disdest >= 10000 & disdest!=11000)
replace sepmode = 2 if disdest==11001
quietly {
forval i = 1/100 {
bysort eid (epist epien sepmode) : gen A = 1 if epist < epien[_n-1] & epien[_n-1]!=.
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epist = epist[_n+1] if epist[_n+1] < epist & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==1 & epien[_n+1]
>  > epien & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==2 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if epien[_n+1] > epien & A[_n+1]==1
drop if A == 1
drop A
}
forval i = 1/5 {
bysort eid (epist epien sepmode) : gen A = 1 if epist == epist[_n-1] & epien == epien[_n-1]
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
```

```
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]!=0 & A[_n+1]==1
drop if A == 1
drop A
}
}
bysort eid (epist epien sepmode) : gen ptr = 1 if admmode==1 | sepmode[_n-1]==1
bysort eid (epist epien sepmode) : gen transferdist = epist-epien[_n-1]
gen tr = 1 if ptr == 1 & inrange(transferdist,0,1)
bysort eid (epist epien sepmode) : replace tr = 1 if transferdist==0 & (MI==1 | MI[_n-1]==1 | IS ==
> 1 | IS[_n-1]==1 | ICH == 1 | ICH[_n-1]==1)
bysort eid (epist epien sepmode) : replace tr =. if tr[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if tr[_n+1]==1
bysort eid (epist epien sepmode) : drop if tr == 1 & tr[_n-1]==.
drop ptr tr transferdist
quietly {
forval i = 1/100 {
bysort eid (epist epien sepmode) : gen ptr = 1 if admmode==1 | sepmode[_n-1]==1
bysort eid (epist epien sepmode) : gen transferdist = epist-epien[_n-1]
gen tr = 1 if ptr == 1 & inrange(transferdist,0,1)
bysort eid (epist epien sepmode) : replace tr = 1 if transferdist==0 & (MI==1 | MI[_n-1]==1 | IS ==
> 1 | IS[_n-1]==1 | ICH == 1 | ICH[_n-1]==1)
bysort eid (epist epien sepmode) : replace tr =. if tr[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & tr[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if tr[_n+1]==1
bysort eid (epist epien sepmode) : drop if tr == 1 & tr[_n-1]==.
drop ptr tr transferdist
bysort eid (epist epien sepmode) : gen A = 1 if epist < epien[_n-1] & epien[_n-1]!=.
bysort eid (epist epien sepmode) : replace A =. if A[_n-1]==1
bysort eid (epist epien sepmode) : replace MI = 1 if MI[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace IS = 1 if IS[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace ICH = 1 if ICH[_n+1]==1 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epist = epist[_n+1] if epist[_n+1] < epist & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==1 & epien[_n+1]
>  > epien & A[_n+1]==1
bysort eid (epist epien sepmode) : replace sepmode = sepmode[_n+1] if sepmode[_n+1]==2 & A[_n+1]==1
bysort eid (epist epien sepmode) : replace epien = epien[_n+1] if epien[_n+1] > epien & A[_n+1]==1
drop if A == 1
drop A
}
}
preserve
keep if MI == 1
keep eid MI epist epien
bysort eid (epist) : keep if _n == 1
save FirstMI, replace
restore
preserve
keep if IS == 1
keep eid IS epist epien
bysort eid (epist) : keep if _n == 1
save FirstIS, replace
restore
keep if ICH == 1
keep eid ICH epist epien
bysort eid (epist) : keep if _n == 1
save FirstICH, replace
use mainset, clear
keep eid
save includedeid, replace
foreach i in MI IS ICH {
use All`i´, clear
merge 1:1 eid epist epien using First`i´, nogen
merge m:1 eid using includedeid, keep(3) nogen
```

```
save All_`i´, replace
}
```

4.4 Summary and population characteristics

In this section, we cleaned the UK Biobank main and HESIN datasets, creating the datasets: *mainset, All_MI, All_IS,* and *All_ICH*. The main dataset (*mainset*) includes all individuals with available information on date of birth, who were not diagnosed with diabetes before age 20 years (i.e., likely type 1 diabetes), and who were free of MI and stroke at baseline (N=‘eid1’).

The event datasets (*All_MI, All_IS,* and *All_ICH*) included all events available in the HESIN datasets, which were defined as follows:

- First MI: presence of the ICD-10 codes I21-I238, I241, or I252 as any diagnosis for an admission.
- Repeat MI: presence of the ICD-10 codes I21-I22 as the primary diagnosis for an admission.
- First IS: presence of the ICD-10 codes I63-I64 as any diagnosis for an admission
- Repeat IS: presence of the ICD-10 codes I63-64 as the primary diagnosis for an admission.
- First ICH: presence of the ICD-10 code I61 as any diagnosis for an admission
- Repeat ICH: presence of the ICD-10 code I61 as the primary diagnosis for an admission.

Population characteristics are shown in Table 4.1.

```
*mkdir CSV
forval s = 0/2 {
use mainset, clear
drop if sex == `s´
count
matrix A = (r(N),.,.,.,.)
gen ageda = (da-dob)/365.25
su age, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
count if DMC!=.
matrix A = (A\r(N),.,.,.,.)
su ldl, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su lpa, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su sbp, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su LSI, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
mat A`s´ = (J(7,1,`s´),A)
}
mat A = (A0\A1\A2)
clear
svmat double A
gen njm = _n
bysort A1 (njm) : gen perc = 100*A2/A2[_n-2] if _n == 3
forval i = 2/6 {
bysort A1 (njm) : replace A`i´ = (A`i´*2.18)-3.83 if _n == 5
replace A`i´ = 0 if A`i´<0
}
tostring A2-A6, gen(B2 B3 B4 B5 B6) format(%9.0fc) force
tostring A2 A3 A4 A5 A6 perc, replace format(%9.1f) force
bysort A1 (njm) : gen A = B2 if _n == 1
bysort A1 (njm) : replace A = B2 + " (" + perc + "\%)" if _n == 3
bysort A1 (njm) : replace A = B4 + " (" + B3 + ", " + B5 + "; " + B2 + ", " + B6 + ")" if _n == 2 |
> _n == 6
bysort A1 (njm) : replace A = A4 + " (" + A3 + ", " + A5 + "; " + A2 + ", " + A6 + ")" if A==""
```

```
bysort A1 (njm) : gen row = _n
keep A1 A row
reshape wide A, j(A1) i(row)
drop row
gen R1 = "N" if _n == 1
replace R1 = "Age" if _n == 2
replace R1 = "N (\%) with diabetes" if _n == 3
replace R1 = "LDL-C (mmol/L)" if _n == 4
replace R1 = "Lp(a) (nmol/L)" if _n == 5
replace R1 = "SBP (mmHg)" if _n == 6
replace R1 = "LSI" if _n == 7
order R1 A1 A0 A2
export delimited using CSV/popchar.csv, delimiter(":") novarnames replace
```

Table 4.1: Baseline characteristics for the UK Biobank sample

	Females	Males	Overall
N	267,476	215,102	482,578
Age	58 (50, 63; 40, 71)	58 (50, 64; 37, 74)	58 (50, 63; 37, 74)
N (%) with diabetes	8,579 (3.2%)	12,770 (5.9%)	21,349 (4.4%)
LDL-C (mmol/L)	3.6 (3.0, 4.2; 0.8, 9.8)	3.5 (2.9, 4.1; 0.3, 9.0)	3.5 (3.0, 4.1; 0.3, 9.8)
Lp(a) (nmol/L)	21.7 (8.2, 79.3; 0.0, 700.4)	17.2 (6.9, 67.3; 0.0, 634.3)	19.5 (7.6, 73.6; 0.0, 700.4)
SBP (mmHg)	133 (121, 147; 72, 255)	140 (129, 152; 65, 252)	136 (124, 150; 65, 255)
LSI	0.0 (0.0, 0.2; 0.0, 3.7)	0.0 (0.0, 0.6; 0.0, 4.0)	0.0 (0.0, 0.4; 0.0, 4.0)

Numeric variables are presented as median (25th centile, 75th centile; minimum, maximum).
Abbreviations: LDL-C – Low density lipoprotein-cholesterol; Lp(a) – Lipoprotein(a); SBP – Systolic blood pressure; LSI – Lifetime smoking index.

5 Transition probabilities

Next, the unadjusted transition probabilities for the model need to be estimated. By unadjusted, we mean the transition probabilities that will be modified by the causal effect of risk factors, as discussed in Section 3.2. MI, IS, ICH, and mortality will be drawn directly from the UK Biobank; the age and sex-specific incidence of type 2 diabetes in the UK will be sourced from the published literature.

According to UK Biobank (https://biobank.ctsu.ox.ac.uk/showcase/exinfo.cgi?src=Data_providers_and_dates ; accessed 14 November 2022), at the time this extract was downloaded follow-up was complete up to:

- 30 September 2021 for mortality data from England and Wales
- 31 October 2021 for mortality data from Scotland
- 30 September 2021 for hospital data from England
- 31 July 2021 for hospital data from Scotland
- 28 February 2018 for hospital data from Wales

So it would make sense to censor at 28 February 2018 for Wales, 31 July 2021 for Scotland, and 30 September 2021 for England. This means that I'm assuming that people remain in the areas they were living at first assessment, and the assessment centre was where they live. This is unlikely a major issue, as the overwhelming majority attended their first assessment in England.

5.1 MI, IS, ICH, and cause-specific mortality prior to CVD

To estimate the age specific-incidence of MI, IS, and ICH (both fatal and non-fatal) and mortality prior to development of CVD, we will use age-period-cohort models [142]. In these, follow-up data will be tabulated into 0.5-year intervals by age and date of follow-up, with each unit containing the number of events and person-years of follow-up, and the value of age and date of follow-up corresponding to the midpoint of the interval. A Poisson model is then fit on this tabulated data. The outcome is the number of events in the interval, the offset log-person-years, and predictors spline effects of age, year, and cohort (year minus age), with a log link function. Knot locations are as suggested by Frank Harrel [143]. Models will be fit separately for males and females. These models are then used to predict the incidence of each outcome by age in 1-year increments, with the prediction year set at 2016, which is roughly the mid-point of follow-up.

The definition of a fatal MI, IS, or ICH will be death during admission for the event, determined by a date of death less than or equal to the end of the stay in hospital. These also include deaths where the underlying cause of death is MI (this includes all CHD deaths), IS, and ICH. Note also here this is the incidence of the first of any of these events, because in the model, once one is developed, the individual enters a new, post-CVD health state.

First, we will tabulate the events, person-years, and calculate the crude rates in 10-year age groups. Then, fit the models and predict the incidence of each outcome.

```
clear
append using All_MI All_IS All_ICH
bysort eid (epist) : keep if _n == 1
save EV1, replace
use mainset, clear
merge 1:1 eid using EV1
drop _merge
gen faildate = td(30,9,2021) if ac == "England"
replace faildate = td(31,7,2021) if ac == "Scotland"
replace faildate = td(28,2,2018) if ac == "Wales"
replace faildate = min(dod,epist,faildate)
gen fail = 1 if (epist==faildate | dod==faildate)
recode fail .=0
gen origin = td(1,1,2006)
stset faildate, fail(fail==1) entry(da) origin(origin) scale(365.25) id(eid)
stsplit age, at(0(10)100) after(time=dob)
gen py = _t-_t0
gen fMI = 1 if ((dod <= epien & epien!=. & MI == 1) | (dod==faildate & chddeath==1)) & fail == 1
gen nfMI = 1 if fail == 1 & fMI==. & MI == 1
gen fIS = 1 if ((dod <= epien & epien!=. & IS == 1) | (dod==faildate & isdeath==1)) & fail == 1
gen nfIS = 1 if fail == 1 & fIS==. & IS == 1
gen fICH = 1 if ((dod <= epien & epien!=. & ICH == 1) | (dod==faildate & ichdeath==1)) & fail == 1
gen nfICH = 1 if fail == 1 & fICH==. & ICH == 1
foreach var of varlist oesc-otherd {
replace `var´ = 0 if fail != 1
replace `var´ = 0 if epist==faildate | chddeath==1 | isdeath==1 | ichdeath==1
}
collapse (sum) py-nfICH oescdeath-otherdeath, by(sex age)
gen age2 = age+9
tostring sex-otherdeath age2, force format(%9.0fc) replace
foreach var of varlist py-otherdeath {
replace `var´ = "$<$6" if `var´ == "1" | `var´ == "2" | `var´ == "3" | `var´ == "4" | `var´ == "5"
}
replace age = age + "-" + age2 if age!="80"
replace age = age+ "+" if age == "80"
drop age2
bysort sex (age) : replace sex ="" if _n!=1
replace sex = "Females" if sex == "0"
replace sex = "Males" if sex == "1"
order sex age py
```

```
preserve
drop oescdeath-otherdeath
export delimited using CSV/miisichntable1.csv, novarnames delimiter(:) replace
restore
drop fMI-nfICH
export delimited using CSV/codtab.csv, novarnames delimiter(:) replace
```

Table 5.1: Incident MI, IS, and ICH counts.

Sex	Age	Person-years	Outcome					
			Fatal MI	Non-fatal MI	Fatal IS	Non-fatal IS	Fatal ICH	Non-fatal ICH
Females	30-39	0	0	0	0	0	0	0
	40-49	286,229	8	52	<6	32	<6	<6
	50-59	915,586	41	549	17	271	21	53
	60-69	1,290,813	204	1,582	68	859	67	156
	70-79	730,244	328	2,038	155	1,424	118	219
	80+	26,951	49	126	14	132	15	20
Males	30-39	<6	0	0	0	0	0	0
	40-49	239,404	29	190	<6	65	<6	16
	50-59	705,656	238	1,501	27	406	15	70
	60-69	986,737	755	3,656	109	1,313	67	181
	70-79	587,039	866	3,546	184	1,758	105	219
	80+	21,082	73	180	19	134	15	21

Table 5.2: Cause-specific mortality counts.

Sex	Age	Person-years	Cause of death											
			Oesophageal cancer	Colon cancer	Pancreatic cancer	Lung cancer	Ovarian cancer	Kidney cancer	Bladder cancer	Pneumonia	COPD	ALS	Parkinson's disease	Other
Females	30-39	0	0	0	0	0	0	0	0	0	0	0	0	0
	40-49	286,229	<6	8	6	10	8	<6	0	<6	0	<6	0	169
	50-59	915,586	14	99	63	109	75	9	11	9	14	16	<6	947
	60-69	1,290,813	74	233	205	514	231	57	34	33	79	68	10	2,499
	70-79	730,244	101	326	287	630	273	57	49	77	154	81	80	3,555
	80+	26,951	10	23	19	28	16	<6	<6	8	12	9	18	380
Males	30-39	<6	0	0	0	0	0	0	0	0	0	0	0	0
	40-49	239,404	<6	10	<6	12	0	<6	<6	<6	<6	<6	0	182
	50-59	705,656	60	121	50	115	0	31	13	19	13	17	<6	936
	60-69	986,737	238	325	228	519	0	106	79	70	115	63	33	2,734
	70-79	587,039	238	381	299	615	0	137	148	122	255	93	162	4,554
	80+	21,082	14	27	20	48	0	12	14	13	24	6	25	461

Other causes includes all other causes except CHD, IS, and ICH.
Abbreviations: COPD – Chronic obstructive pulmonary disease; ALS – amyotrophic lateral sclerosis.

```
*mkdir INC
*mkdir GPH
use mainset, clear
merge 1:1 eid using EV1
drop _merge
gen faildate = td(30,9,2021) if ac == "England"
replace faildate = td(31,7,2021) if ac == "Scotland"
replace faildate = td(28,2,2018) if ac == "Wales"
replace faildate = min(dod,epist,faildate)
gen fail = 1 if (epist==faildate | dod==faildate)
recode fail .=0
gen origin = td(1,1,2006)
stset faildate, fail(fail==1) entry(da) origin(origin) scale(365.25) id(eid)
stsplit age, at(0(0.5)100) after(time=dob)
stsplit year, at(0(0.5)20)
gen py = _t-_t0
gen fMI = 1 if ((dod <= epien & epien!=. & MI == 1) | (dod==faildate & chddeath==1)) & fail == 1
gen nfMI = 1 if fail == 1 & fMI==. & MI == 1
gen fIS = 1 if ((dod <= epien & epien!=. & IS == 1) | (dod==faildate & isdeath==1)) & fail == 1
gen nfIS = 1 if fail == 1 & fIS==. & IS == 1
gen fICH = 1 if ((dod <= epien & epien!=. & ICH == 1) | (dod==faildate & ichdeath==1)) & fail == 1
gen nfICH = 1 if fail == 1 & fICH==. & ICH == 1
foreach var of varlist oesc-otherd {
replace `var´ = 0 if fail != 1
replace `var´ = 0 if epist==faildate | chddeath==1 | isdeath==1 | ichdeath==1
}
collapse (sum) py-nfICH oescdeath-otherdeath, by(sex age year)
save collapseset, replace
use collapseset, clear
foreach var of varlist fMI-luncdeath kidcdeath-otherdeath {
forval s = 0/1 {
use collapseset, clear
keep if sex == `s´
gen coh = year-age
replace age = age+0.25
replace year = year+0.25
pctile AA=age [weight=`var´], nq(40)
foreach i in 2 11 20 29 38 {
local a`i´ = r(r`i´)
}
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
pctile BB=year [weight=`var´], nq(40)
foreach i in 2 11 20 29 38 {
local b`i´ = r(r`i´)
}
mkspline yearsp = year, cubic knots(`b2´ `b11´ `b20´ `b29´ `b38´)
pctile CC=coh [weight=`var´], nq(40)
foreach i in 2 11 20 29 38 {
local c`i´ = r(r`i´)
}
mkspline cohsp = coh, cubic knots(`c2´ `c11´ `c20´ `c29´ `c38´)
poisson `var´ agesp* yearsp* cohsp*, exposure(py)
clear
set obs 55
gen age = _n+29.5
gen year = 10
gen coh = year-age
gen py = 1
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
mkspline yearsp = year, cubic knots(`b2´ `b11´ `b20´ `b29´ `b38´)
mkspline cohsp = coh, cubic knots(`c2´ `c11´ `c20´ `c29´ `c38´)
predict rate, ir
predict errr, stdp
replace age = age-0.5
gen sex = `s´
keep age sex rate errr
save INC/`var´_`s´, replace
}
```

```
}
use collapseset, clear
keep if sex == 0
gen coh = year-age
replace age = age+0.25
replace year = year+0.25
pctile AA=age [weight=ovacdeath], nq(40)
foreach i in 2 11 20 29 38 {
local a`i´ = r(r`i´)
}
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
pctile BB=year [weight=ovacdeath], nq(40)
foreach i in 2 11 20 29 38 {
local b`i´ = r(r`i´)
}
mkspline yearsp = year, cubic knots(`b2´ `b11´ `b20´ `b29´ `b38´)
pctile CC=coh [weight=ovacdeath], nq(40)
foreach i in 2 11 20 29 38 {
local c`i´ = r(r`i´)
}
mkspline cohsp = coh, cubic knots(`c2´ `c11´ `c20´ `c29´ `c38´)
poisson ovacdeath agesp* yearsp* cohsp*, exposure(py)
clear
set obs 55
gen age = _n+29.5
gen year = 10
gen coh = year-age
gen py = 1
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
mkspline yearsp = year, cubic knots(`b2´ `b11´ `b20´ `b29´ `b38´)
mkspline cohsp = coh, cubic knots(`c2´ `c11´ `c20´ `c29´ `c38´)
predict rate, ir
predict errr, stdp
replace age = age-0.5
replace age = round(age,.1)
gen sex = 0
keep age sex rate errr
save INC/ovacdeath, replace
use collapseset, clear
foreach ii of varlist fMI-luncdeath kidcdeath-otherdeath {
clear
append using INC/`ii´_0 INC/`ii´_1
save INC/`ii´, replace
}
use collapseset, clear
foreach var of varlist fMI-nfICH {
if "`var´" == "fMI" {
local va = "Fatal MI"
}
if "`var´" == "nfMI" {
local va = "Non-fatal MI"
}
if "`var´" == "fIS" {
local va = "Fatal IS"
}
if "`var´" == "nfIS" {
local va = "Non-fatal IS"
}
if "`var´" == "fICH" {
local va = "Fatal ICH"
}
if "`var´" == "nfICH" {
local va = "Non-fatal ICH"
}
use INC/`var´, clear
replace rate = rate*1000
gen lb = exp(ln(rate)-1.96*errr)
gen ub = exp(ln(rate)+1.96*errr)
```

```
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 1, color(blue) lpattern(solid)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) yscale(log range(0.0001 15)) ///
graphregion(color(white)) ///
ylabel(0.0001 "0.0001" 0.001 "0.001" 0.01 "0.01" ///
0.1 "0.1"  1 "1" 10 "10" , angle(0)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
xtitle("Age (years)") title(`va´, placement(west) size(medium) color(black))
graph save "Graph" GPH/asr_`var´, replace
}
use collapseset, clear
foreach var of varlist oescdeath-luncdeath kidcdeath-otherdeath {
if "`var´" == "oescdeath" {
local va = "Oesophagageal cancer"
}
if "`var´" == "colcdeath" {
local va = "Colon cancer"
}
if "`var´" == "pancdeath" {
local va = "Pancreatic cancer"
}
if "`var´" == "luncdeath" {
local va = "Lung cancer"
}
if "`var´" == "kidcdeath" {
local va = "Kidney cancer"
}
if "`var´" == "blacdeath" {
local va = "Bladder cancer"
}
if "`var´" == "pneudeath" {
local va = "Pneumonia"
}
if "`var´" == "copddeath" {
local va = "COPD"
}
if "`var´" == "alsdeath" {
local va = "ALS"
}
if "`var´" == "pddeath" {
local va = "Parkinson´s disease"
}
if "`var´" == "otherdeath" {
local va = "Other"
}
use INC/`var´, clear
replace rate = rate*1000
gen lb = exp(ln(rate)-1.96*errr)
gen ub = exp(ln(rate)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 1, color(blue) lpattern(solid)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) yscale(log range(0.00001 100)) ///
graphregion(color(white)) ///
ylabel(0.00001 "0.00001" 0.0001 "0.0001" 0.001 "0.001" 0.01 "0.01" ///
0.1 "0.1"  1 "1" 10 "10" 100 "100", angle(0)) ///
```

```
xlabel(30(10)80, nogrid) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
xtitle("Age (years)") title(`va´, placement(west) size(medium) color(black))
graph save "Graph" GPH/asr_`var´, replace
}
clear
append using INC/ovacdeath
replace rate = rate*1000
gen lb = exp(ln(rate)-1.96*errr)
gen ub = exp(ln(rate)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 0, color(red) lpattern(solid)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(2 "Females") ///
cols(1)) yscale(log range(0.00001 100)) ///
graphregion(color(white)) ///
ylabel(0.00001 "0.00001" 0.0001 "0.0001" 0.001 "0.001" 0.01 "0.01" ///
0.1 "0.1"  1 "1" 10 "10" 100 "100", angle(0)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
xtitle("Age (years)") title(Ovarian cancer, placement(west) size(medium) color(black))
graph save "Graph" GPH/asr_ovacdeath, replace


graph combine ///
GPH/asr_nfMI.gph ///
GPH/asr_fMI.gph ///
GPH/asr_nfIS.gph ///
GPH/asr_fIS.gph ///
GPH/asr_nfICH.gph ///
GPH/asr_fICH.gph ///
, graphregion(color(white)) altshrink cols(2) xsize(4)
graph combine ///
GPH/asr_oescdeath.gph ///
GPH/asr_colcdeath.gph ///
GPH/asr_pancdeath.gph ///
GPH/asr_luncdeath.gph ///
GPH/asr_ovacdeath.gph ///
GPH/asr_kidcdeath.gph ///
GPH/asr_blacdeath.gph ///
GPH/asr_pneudeath.gph ///
GPH/asr_copddeath.gph ///
GPH/asr_alsdeath.gph ///
GPH/asr_pddeath.gph ///
GPH/asr_otherdeath.gph ///
, graphregion(color(white)) altshrink cols(3) xsize(4)
```

Age- and sex-specific incidence and mortality rates are presented in Figures 5.1 - 5.2.

Figure 5.1: Age- and sex-specific incidence of CVD, by CVD type.

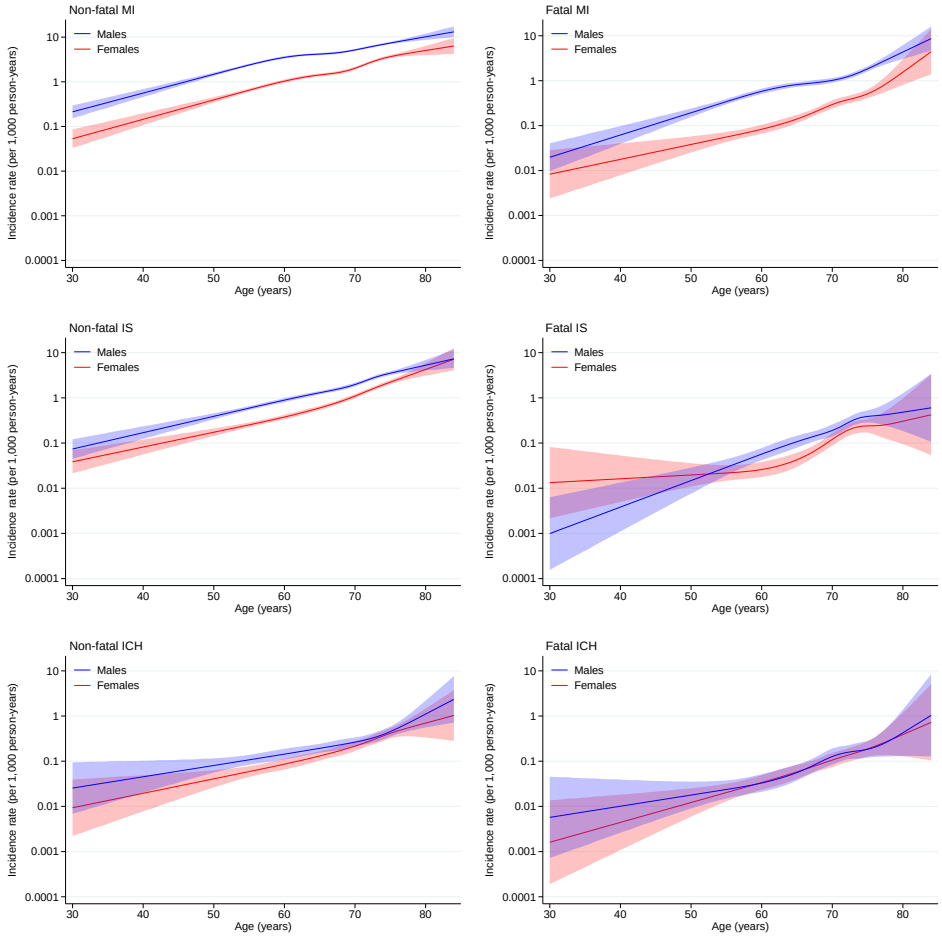
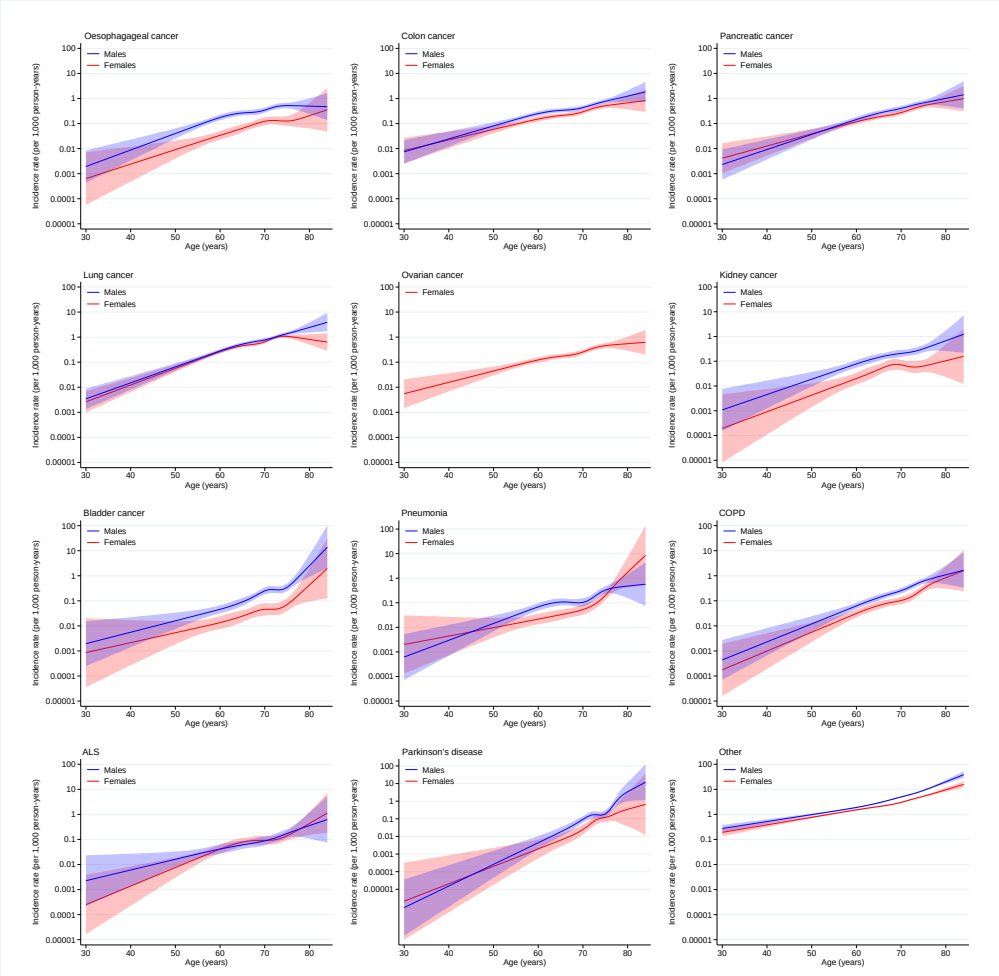


Figure 5.2: Age- and sex-specific mortality rate, by cause of death. Other causes includes all other causes except CHD, IS, and ICH.
Abbreviations: COPD – Chronic obstructive pulmonary disease; ALS – amyotrophic lateral sclerosis.



5.2 Repeat events and mortality following CVD

There are 15 transitions that fall into this category:

1. MI → non-fatal MI
2. MI → fatal MI
3. MI → non-fatal stroke
4. MI → fatal stroke
5. MI → other death
6. Stroke → non-fatal stroke
7. Stroke → fatal stroke
8. Stroke → non-fatal MI
9. Stroke → fatal MI
10. Stroke → other death
11. MI and stroke → non-fatal MI
12. MI and stroke → fatal MI
13. MI and stroke → non-fatal stroke
14. MI and stroke → fatal stroke
15. MI and stroke → other death

(Recall that the transition to type 2 diabetes in these health states is unaffected by the presence of CVD).

The definition of a fatal MI, IS, and ICH are as above. First, we must check how much power we have for each of these transitions before deciding how to analyse the data.

```
clear
append using All_MI All_IS All_ICH
bysort eid (epist) : drop if epist==epist[_n-1]
save REV, replace
use mainset, clear
merge 1:1 eid using EV1
drop _merge
drop if epist==.
drop if dod <= epien
drop MI IS ICH
rename (epist epien) (fepist fepien)
merge 1:m eid using REV
drop if _merge == 2
drop _merge
sort eid epist
gen faildate = td(30,9,2021) if ac == "England"
replace faildate = td(31,7,2021) if ac == "Scotland"
replace faildate = td(28,2,2018) if ac == "Wales"
drop if epist > faildate
replace epien = dod if dod < epien
```

```
bysort eid (epist) : replace faildate = min(dod,epist[_n+1],faildate)
bysort eid (epist) : gen entryvar = fepien if _n == 1
bysort eid (epist) : replace entryvar = epien if _n != 1
format faildate entryvar %td
gen HS =.
bysort eid (epist) : replace HS = 1 if MI == 1 & _n == 1
bysort eid (epist) : replace HS = 2 if MI == . & _n == 1
forval i = 2/8 {
bysort eid (epist) : replace HS = 1 if MI == 1 & HS[_n-1]==1 & _n == `i´
bysort eid (epist) : replace HS = 2 if MI == . & HS[_n-1]==2 & _n == `i´
bysort eid (epist) : replace HS = 3 if MI == 1 & HS[_n-1]==2 & _n == `i´
bysort eid (epist) : replace HS = 3 if MI == . & HS[_n-1]==1 & _n == `i´
bysort eid (epist) : replace HS = 3 if HS[_n-1]==3 & _n == `i´
}
bysort eid (epist) : gen fatal = 1 if dod <= epien[_n+1] & epien[_n+1]!=.
gen OC = 0
bysort eid (epist) : replace OC = 1 if MI[_n+1] == 1 & fatal==.
bysort eid (epist) : replace OC = 2 if (MI[_n+1] == 1 & fatal==1) | (dod==faildate & chddeath==1)
bysort eid (epist) : replace OC = 3 if (IS[_n+1] == 1 | ICH[_n+1] == 1) & fatal==.
bysort eid (epist) : replace OC = 4 if ((IS[_n+1] == 1 | ICH[_n+1] == 1) & fatal==1) | (dod==faildat
> e & (isdeath==1 | ichdeath==1))
bysort eid (epist) : replace OC = 5 if faildate==dod & chddeath==. & isdeath==. & ichdeath==.
drop if entryvar>=faildate
save reventset, replace
forval i = 1/3 {
use reventset, clear
keep if HS == `i´
forval ii = 1/5 {
preserve
gen fail = 1 if OC == `ii´
recode fail .=0
gen origin = td(1,1,2006)
gen id = _n
stset faildate, fail(fail==1) entry(entryvar) origin(origin) scale(365.25) id(id)
stsplit age, at(0(10)100) after(time=dob)
gen py = _t-_t0
collapse (sum) fail py, by(sex age)
rename fail fail`ii´
save tabbedrev_`i´_`ii´, replace
restore
}
use tabbedrev_`i´_1, clear
forval ii = 2/5 {
merge 1:1 sex age using tabbedrev_`i´_`ii´
drop _merge
}
if `i´ == 3 {
expand 2 if sex == 0 & age == 50
bysort sex age : replace age = 40 if _n == 1 & _n!=_N
foreach var of varlist fail1-fail5 {
replace `var´ = 0 if sex == 0 & age == 40
}
}
gen age2 = age+9
tostring py sex age fail1 fail2 fail3 fail4 fail5 age2, force format(%9.0fc) replace
forval ii = 1/5 {
replace fail`ii´ = "$<$6" if fail`ii´ == "1" |  fail`ii´ == "2" |  fail`ii´ == "3" |  fail`ii´ == "4
> " |  fail`ii´ == "5"
}
replace age = age + "-" + age2 if age!="80"
replace age = age+ "+" if age == "80"
drop age2
bysort sex (age) : replace sex ="" if _n!=1
replace sex = "Females" if sex == "0"
replace sex = "Males" if sex == "1"
order sex age py
export delimited using CSV/HS`i´pevtab.csv, novarnames delimiter(:) replace
}
```

The breakdowns of events by starting health state, sex, and age are shown in Tables 5.3 - 5.5. The best way to analyse this data, given the lack of power (discussed in Morton et al. [29]), will be to assume the incidence and mortality rates are proportional to age, and include factors for prior MI and stroke. Thus, data will be tabulated into 0.5-year intervals by age, with each unit containing the number of events and person-years of follow-up, and the value of age corresponding to the midpoint of the interval. A Poisson model is then fit on this tabulated data. The outcome is the number of events in the interval, the offset log-person-years, and predictors spline effects of age and factor effects of prior MI or stroke, with a log link function. Knot locations are as suggested by Frank Harrel [143]. Models will be fit separately for males and females. These models are then used to predict the incidence of each outcome by age in 1-year increments.

Table 5.3: MI, stroke, and other death counts following MI.

			Outcome				
Sex	Age	Person-years	Non-fatal MI	Fatal MI	Non-fatal stroke	Fatal stroke	Other death
Females	40-49	113	<6	0	0	0	0
	50-59	1,967	20	<6	8	<6	14
	60-69	6,368	99	16	18	<6	103
	70-79	10,284	168	40	55	16	282
	80+	895	9	<6	7	<6	53
Males	40-49	446	6	0	0	0	<6
	50-59	5,899	68	20	12	<6	34
	60-69	16,317	228	79	61	13	180
	70-79	21,034	300	167	132	32	578
	80+	1,522	22	13	7	<6	85

Table 5.4: MI, stroke, and other death counts following stroke.

			Outcome				
Sex	Age	Person-years	Non-fatal MI	Fatal MI	Non-fatal stroke	Fatal stroke	Other death
Females	40-49	110	0	0	<6	0	<6
	50-59	1,228	<6	<6	17	<6	24
	60-69	3,736	17	7	58	12	86
	70-79	7,444	61	22	209	71	248
	80+	759	7	<6	36	14	34
Males	40-49	183	<6	0	<6	0	6
	50-59	1,848	18	<6	41	<6	39
	60-69	5,541	59	11	123	26	121
	70-79	9,412	125	46	246	82	393
	80+	847	13	8	27	17	47

```
forval i = 1/5 {
use reventset, clear
gen fail = 1 if OC == `i´
ta OC
recode fail .=0
gen origin = td(1,1,2006)
gen id = _n
stset faildate, fail(fail==1) entry(entryvar) origin(origin) scale(365.25) id(id)
stsplit age, at(0(0.5)100) after(time=dob)
gen py = _t-_t0
collapse (sum) fail py, by(sex age HS)
if `i´ == 1 {
rename fail nfMI
}
if `i´ == 2 {
```

Table 5.5: MI, stroke, and other death counts following MI and stroke.

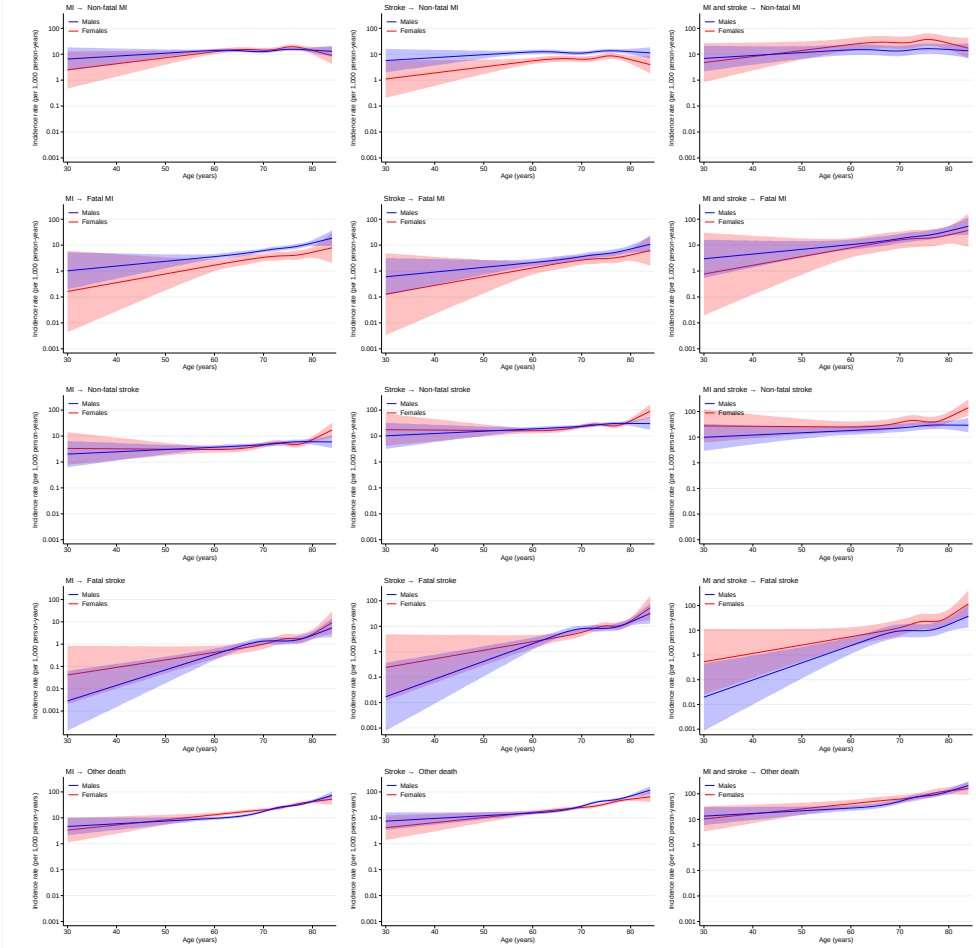
			Outcome				
Sex	Age	Person-years	Non-fatal MI	Fatal MI	Non-fatal stroke	Fatal stroke	Other death
Females	40-49	0	0	0	0	0	0
	50-59	36	0	0	<6	0	<6
	60-69	98	<6	<6	7	<6	<6
	70-79	295	7	7	8	<6	28
	80+	40	<6	0	<6	<6	<6
Males	40-49	3	0	0	0	0	0
	50-59	78	0	<6	0	0	<6
	60-69	312	8	9	10	<6	15
	70-79	794	11	12	21	9	60
	80+	83	0	<6	<6	<6	8

```
rename fail fMI
}
if `i´ == 3 {
rename fail nfS
}
if `i´ == 4 {
rename fail fS
}
if `i´ == 5 {
rename fail othd
}
save prevcol_`i´, replace
}
use prevcol_1, clear
forval i = 2/5 {
merge 1:1 sex HS age using prevcol_`i´
drop _merge
}
save pevcollapseset, replace
use pevcollapseset, clear
gen MI = 0
gen ST = 0
replace MI = 1 if HS == 1 | HS == 3
replace ST = 1 if HS == 2 | HS == 3
foreach var of varlist nfMI fMI-othd {
forval s = 0/1 {
preserve
keep if sex == `s´
replace age = age+0.25
pctile AA=age [weight=`var´], nq(40)
foreach i in 2 11 20 29 38 {
local a`i´ = r(r`i´)
}
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
poisson `var´ i.MI i.ST c.agesp*, exposure(py)
clear
set obs 55
gen age = _n+29.5
gen py = 1
expand 3
gen MI = 0
gen ST = 0
bysort age : replace MI = 1 if _n == 1 | _n == 3
bysort age : replace ST = 1 if _n == 2 | _n == 3
mkspline agesp = age, cubic knots(`a2´ `a11´ `a20´ `a29´ `a38´)
predict rate, ir
predict errr, stdp
replace age = age-0.5
replace age = round(age,.1)
gen sex = `s´
```

```
keep age sex rate errr MI ST
save INC/pev_`var´_`s´, replace
restore
}
}
clear
gen oc = ""
foreach i in nfMI fMI nfS fS othd {
append using INC/pev_`i´_0 INC/pev_`i´_1
replace oc = "`i´" if oc =""
}
recast double rate
recast double errr
reshape wide rate errr, i(age MI ST sex) j(oc) string
order age MI ST sex ratefMI ratenfMI ratefS ratenfS rateothd
preserve
drop errrfMI-errrothd
save pevtp, replace
restore
save pevtpe, replace
foreach i in nfMI fMI nfS fS othd {
if "`i´" == "nfMI" {
local d1 = "Non-fatal MI"
}
if "`i´" == "fMI" {
local d1 = "Fatal MI"
}
if "`i´" == "nfS" {
local d1 = "Non-fatal stroke"
}
if "`i´" == "fS" {
local d1 = "Fatal stroke"
}
if "`i´" == "othd" {
local d1 = "Other death"
}
forval ii = 1/3 {
clear
append using INC/pev_`i´_0 INC/pev_`i´_1
if `ii´ == 1 {
keep if MI == 1 & ST == 0
local d2 = "MI → "
}
if `ii´ == 2 {
keep if MI == 0 & ST == 1
local d2 = "Stroke → "
}
if `ii´ == 3 {
keep if MI == 1 & ST == 1
local d2 = "MI and stroke → "
}
replace rate = rate*1000
gen lb = exp(ln(rate)-1.96*errr)
gen ub = exp(ln(rate)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 1, color(blue) lpattern(solid)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) yscale(log range(0.001 260)) ///
graphregion(color(white)) ///
ylabel(0.001 "0.001" 0.01 "0.01" ///
0.1 "0.1"  1 "1" 10 "10" 100 "100", angle(0)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
```

```
xtitle("Age (years)") title(`d2´ `d1´, placement(west) size(medium) color(black))
graph save "Graph" GPH/pevasr_`i´_`ii´, replace
}
}
```

Figure 5.3: Age- and sex-specific incidence of recurrent CVD or death, by CVD type and starting health state.




```
graph combine ///
GPH/pevasr_nfMI_1.gph ///
GPH/pevasr_nfMI_2.gph ///
GPH/pevasr_nfMI_3.gph ///
GPH/pevasr_fMI_1.gph ///
GPH/pevasr_fMI_2.gph ///
GPH/pevasr_fMI_3.gph ///
GPH/pevasr_nfS_1.gph ///
GPH/pevasr_nfS_2.gph ///
GPH/pevasr_nfS_3.gph ///
GPH/pevasr_fS_1.gph ///
GPH/pevasr_fS_2.gph ///
GPH/pevasr_fS_3.gph ///
GPH/pevasr_othd_1.gph ///
GPH/pevasr_othd_2.gph ///
```

```
GPH/pevasr_othd_3.gph ///
, graphregion(color(white)) altshrink cols(3) xsize(4)
> tate.)
```

5.3 The incidence of type 2 diabetes

Pal et al [140] contains the most contemporary diabetes incidence data from the UK (between 1 January 2009 and 31 December 2018). Their results are summarised in Table 5.6.

Table 5.6: Incidence of type 2 diabetes by age and gender.

	Incidence rate (per 1,000 person-years)	
Age, years	Men	Women
0-19	0.09 (0.08 to 0.09)	0.15 (0.14 to 0.16)
20-29	0.40 (0.38 to 0.41)	0.92 (0.89 to 0.95)
30-39	1.47 (1.44 to 1.51)	1.69 (1.66 to 1.72)
40-49	4.28 (4.23 to 4.33)	3.16 (3.12 to 3.21)
50-59	8.31 (8.23 to 8.39)	5.82 (5.75 to 5.88)
60-69	12.49 (12.38 to 12.60)	8.85 (8.76 to 8.94)
70-79	13.69 (13.21 to 13.84)	11.01 (10.89 to 11.13)
80-89	10.55 (10.35 to 10.76)	8.86 (8.72 to 9.00)
90-99	7.02 (6.58 to 7.48)	5.39 (5.16 to 5.63)

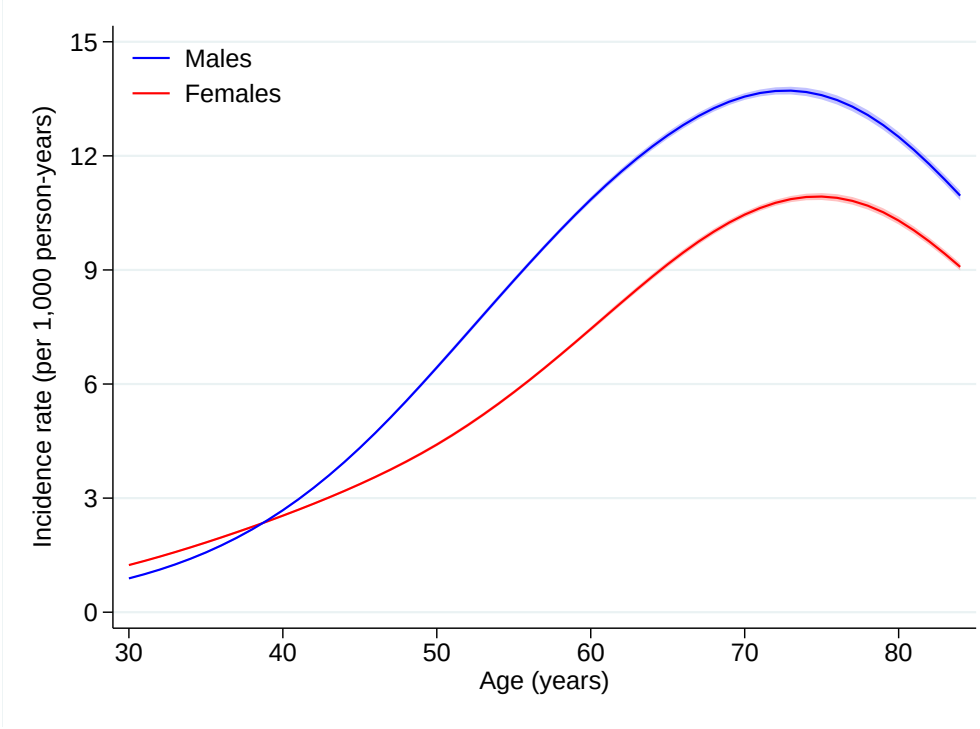
Now we just need to convert these to age-specific rates. To get proper estimates of uncertainty for these, it will be important to have reasonable estimates of the population numbers underlying the rate estimates in Pal et al. They present the overall number of cases of type 2 diabetes by sex (228,034 men and 198,683 women), and the crude incidence of type 2 diabetes by sex (4.51 in men, 3.88 in women, per 1,000 person-years), from which we can calculate the overall person-years of follow-up via: $py = \frac{N}{incidence}$ = 50,561,863 in men; 51,206,959 in women. To distribute this via age, we will assume the population distribution of age in their study was representative of the UK and thus use data from the Office for National Statistics for mid-2013 (to match the mid-point of Pal et al) [144]. With this data, it is possible to model the age-specific incidence of type 2 diabetes in the UK using Poisson regression – the model will use calculated counts as the outcome, calculated person-years of follow-up as the exposure (log), and include a spline effect of age (as the mid-point of the interval), fit separately by sex, with log-link. This model will then predict the age-specific incidence of type 2 diabetes in the UK (Figure 5.4).

```
copy /home/jed/Downloads/ukpop2013.dta ukpop2013.dta
*ONS data
use ukpop2013, clear
tostring age, replace
replace age = "0" + age if length(age)==1
replace age = substr(age,1,1)
replace age = "1" if age == "0"
collapse (sum) N, by(sex age)
bysort sex (age) : egen double sm = sum(N)
gen prp = N/sm
replace age = age+"5"
replace age = "10" if age == "15"
destring age, replace
keep age sex prp
save ukpopprp, replace
clear
set obs 2
gen sex = _n-1
expand 9
bysort sex : gen age = _n*10+5 if _n !=1
recode age .=10
gen inc=.
replace inc = 0.15 in 1
replace inc = 0.92 in 2
replace inc = 1.69 in 3
replace inc = 3.16 in 4
```

```
replace inc = 5.82 in 5
replace inc = 8.85 in 6
replace inc = 11.01 in 7
replace inc = 8.86 in 8
replace inc = 5.39 in 9
replace inc = 0.09 in 10
replace inc = 0.40 in 11
replace inc = 1.47 in 12
replace inc = 4.28 in 13
replace inc = 8.31 in 14
replace inc = 12.49 in 15
replace inc = 13.69 in 16
replace inc = 10.55 in 17
replace inc = 7.02 in 18
replace inc = (inc/1000)
save paletal, replace
gen double py = 51206959 if sex == 0
recode py .=50561863
merge 1:1 sex age using ukpopprp
drop _merge
replace py = py*prp
gen N = inc*py
mkspline agesp=age, cubic knots(20(15)95)
forval s = 0/1 {
preserve
poisson N agesp* if sex == `s´, exposure(py)
clear
set obs 55
gen age = _n+29.5
gen py = 1
mkspline agesp=age, cubic knots(20(15)95)
predict rate, ir
predict errr, stdp
replace age = age-0.5
replace age = round(age,.1)
gen sex = `s´
keep age sex rate errr
save INC/t2d_`s´, replace
restore
}


clear
append using INC/t2d_0 INC/t2d_1
save INC/t2d, replace
replace rate = rate*1000
gen lb = exp(ln(rate)-1.96*errr)
gen ub = exp(ln(rate)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line rate age if sex == 1, color(blue) lpattern(solid)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) yscale(nolog range(0 15)) ///
graphregion(color(white)) ///
ylabel(0(3)15, angle(0) format(%9.0f)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
xtitle("Age (years)")
```

Figure 5.4: Age- and sex-specific incidence of type 2 diabetes in the UK.



6 Risk factor trajectories

Given the way the model works – recall: $R_a = R \times M^{x-\mu}$ from section 3.2. We have just estimated R in the previous section. In this section, we need to estimate μ (x, and its dependent, R_a, need to be esimtated in the model, so that won't be done here).

It is important that μ, the time-weighted mean cumulative level of the risk factor (or LSI) across the population, matches the values in the UK Biobank sample used to estimate the transition probabilies in the previous section.

For all risk factors, these mean values will be informed by combination of published literature and the UK Biobank sample. Note that obviously there is individual variability in these trajectories, but here all that matters is the mean.

6.1 LDL-C

From age 40-69, the values can be estimated directly from UK Biobank. Note by using means from age 40-69, this incorporates the effect of lipid lowering therapy and the increasing prevalence of its use with age, as well as decreased survival in people with very high LDL-C. We will estimate the mean LDL-C by age and sex using using linear regression with a spline effect of age, stratified by sex. This model will then be used to project LDL-C beyond age 69.

Prior to age 40, we will assume that mean LDL-C is 0.75 mmol/L at birth [145]; that LDL-C increases linearly from birth to a mean of 2 mmol/L by age 5 [146]; and that after age 5 LDL-C increases linearly to the mean value of LDL-C in UK Biobank at age 40 (as defined by the linear regression model).

Following this, the cumulative LDL-C, and then mean cumulative LDL-C are estimated, using the lower and upper bounds of the 95%CIs estimated from the linear regression model. Recall that the cumulative mean needs to be time-weighted; calibration of time-weighting has been performed previously [29]. The equation that represents the time-weighting is: $w = (\frac{x-3}{3})^{-2}$, where w is the weighting factor and x is years before the current age for which mean cumulative LDL-C is being calculated.

The estimated age and sex-specific mean LDL-C and mean mean [two means intentional] cumulative LDL-C are shown in Figures 6.1 and 6.2. (We have excluded people with missing LDL-C values on the assumption that there is no major difference in the LDL-C of those missing and those with values available.)

```
forval s = 0/1 {
use mainset, clear
keep if sex == `s´
gen ageda = (da-dob)/365.25
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
reg ldl agesp*
clear
set obs 85
gen age = (_n-1)+0.5
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
predict ldl if inrange(age,40,85)
predict errr if inrange(age,40,85), stdp
replace age = age-0.5
replace ldl = 0.75 if age == 0
replace ldl = 2  if age == 5
replace ldl = (ldl[6]-ldl[1])/5 if inrange(age,1,4)
replace ldl = (ldl[41]-ldl[6])/35 if inrange(age,6,39)
replace ldl = sum(ldl) if inrange(age,0,4)
```

```
replace ldl = sum(ldl) if inrange(age,5,39)
gen sex = `s´
keep age sex ldl errr
save ldlmod_`s´, replace
}
clear
append using ldlmod_0 ldlmod_1
save ldlmod, replace
forval s = 0/1 {
use ldlmod_`s´, clear
gen lb = ldl-(1.96*errr)
gen ub = ldl+(1.96*errr)
replace lb = ldl if lb==.
replace ub = ldl if ub==.
gen cumldl=.
gen cumldllb=.
gen cumldlub=.
gen aveldl=.
gen aveldllb=.
gen aveldlub=.
forval ii = 0/84 {
gen logf = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog = ldl*logf
gen ldlloglb = lb*logf
gen ldllogub = ub*logf
gen cumldllog = sum(ldllog)
gen cumldlloglb = sum(ldlloglb)
gen cumldllogub = sum(ldllogub)
gen cumlog = sum(logf)
replace cumldl = cumldllog if age == `ii´
replace cumldllb = cumldlloglb if age == `ii´
replace cumldlub = cumldllogub if age == `ii´
replace aveldl = cumldllog/cumlog if age == `ii´
replace aveldllb = cumldlloglb/cumlog if age == `ii´
replace aveldlub = cumldllogub/cumlog if age == `ii´
drop logf ldllog ldlloglb ldllogub cumldllog cumldlloglb cumldllogub cumlog
}
keep sex age aveldl aveldllb aveldlub
keep if age >= 30
save aveldl_cal_`s´, replace
}
clear
append using aveldl_cal_0 aveldl_cal_1
save aveldl_cal, replace


use ldlmod, clear
gen lb = ldl-(1.96*errr)
gen ub = ldl+(1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line ldl age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line ldl age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(0(10)80, nogrid) ///
ytitle("LDL-C (mmol/L)") ///
xtitle("Age")
use aveldl_cal, clear
twoway ///
(rarea aveldlub aveldllb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line aveldl age if sex == 0, color(red) lpattern(solid)) ///
(rarea aveldlub aveldllb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line aveldl age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
```

Figure 6.1: Mean LDL-C by age and sex.

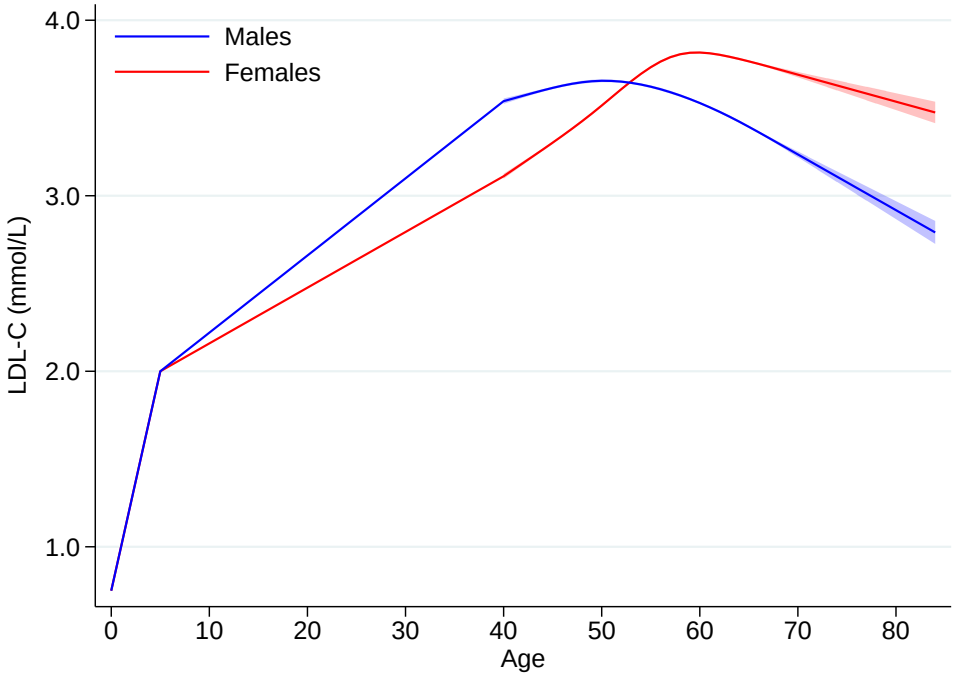
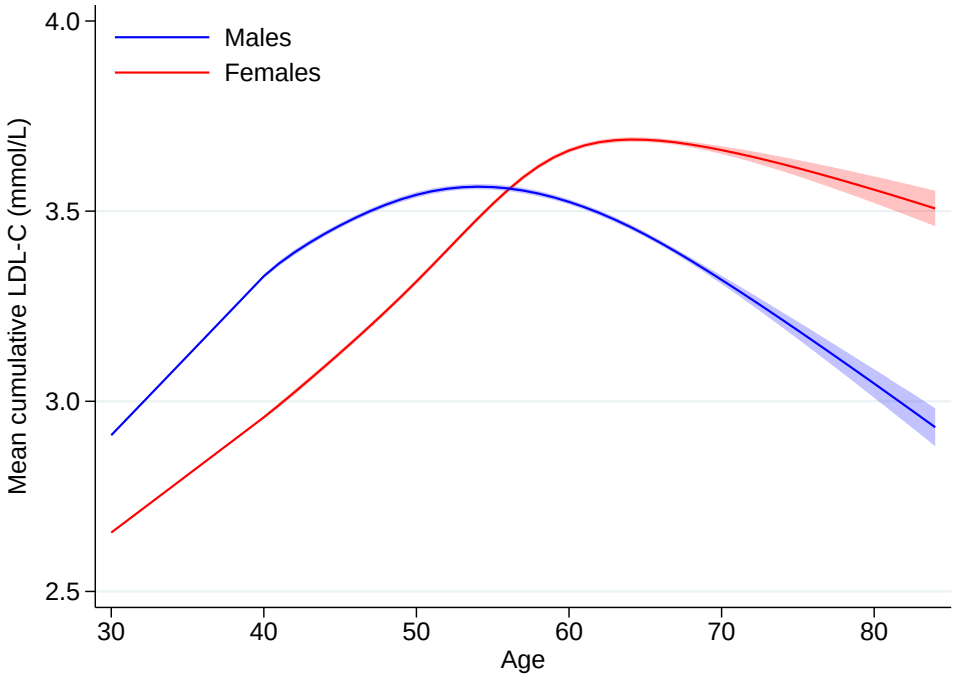


Figure 6.2: Time-weighted mean cumulative LDL-C by age and sex.



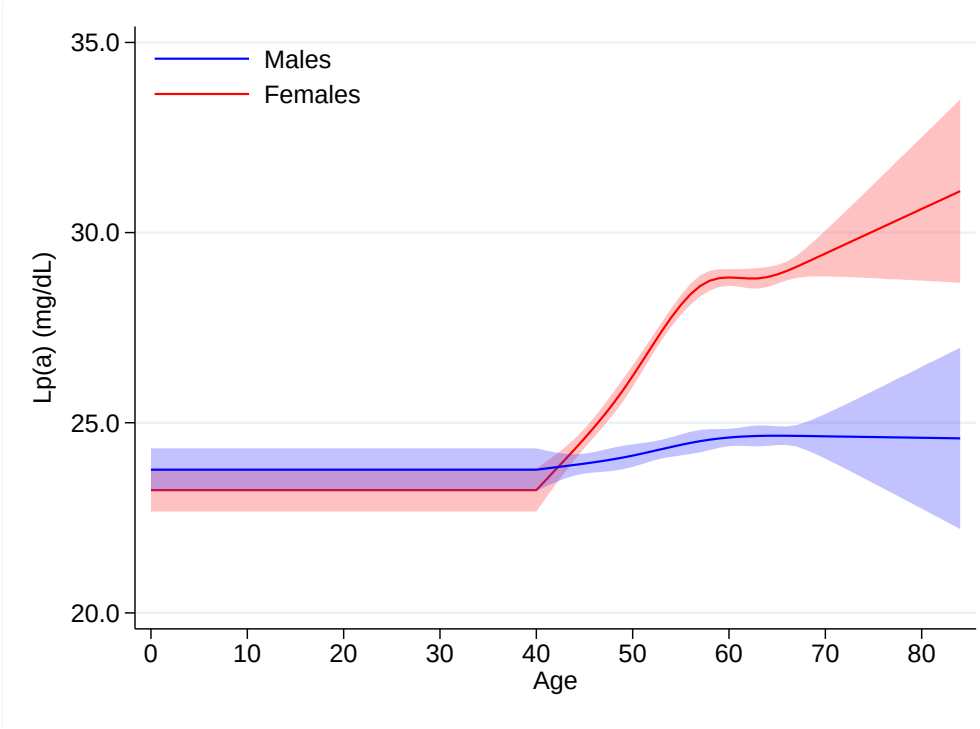
```
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Mean cumulative LDL-C (mmol/L)") ///
xtitle("Age")
```

6.2 Lp(a)

Lp(a) is fully expressed by the first year of life [7, 8, 9], and it is largely genetically determined [10], meaning the Lp(a) value measured in adulthood should effectively be the same across the lifespan. As with LDL-C, Lp(a) will be modelled using linear regression with a spline effect of age, stratified by sex. These models will then be used to project Lp(a) from age 70 to 85 years, and prior to age 40 Lp(a) will be assumed to be the same value as age 40. The projected mean Lp(a) across the lifespan is shown in Figure 6.3 and the mean mean cumulative Lp(a) in Figure 6.4. The time-weighting is the same as for LDL-C.

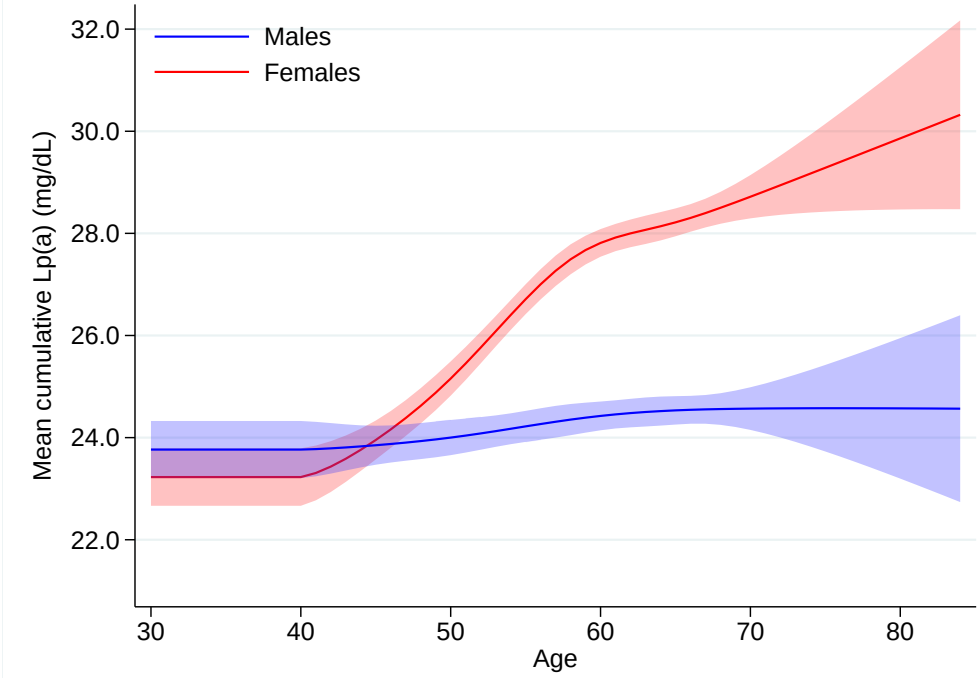
```
forval s = 0/1 {
use mainset, clear
keep if sex == `s´
gen ageda = (da-dob)/365.25
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
reg lpa agesp*
clear
set obs 85
gen age = (_n-1)+0.5
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
predict lpa if inrange(age,40,85)
predict errr if inrange(age,40,85), stdp
replace age = age-0.5
replace lpa = lpa[41] if lpa==.
replace errr = errr[41] if errr==.
gen sex = `s´
keep age sex lpa errr
save lpamod_`s´, replace
}
forval s = 0/1 {
use lpamod_`s´, clear
gen lb = lpa-(1.96*errr)
gen ub = lpa+(1.96*errr)
replace lb = lpa if lb==.
replace ub = lpa if ub==.
gen cumlpa=.
gen cumlpalb=.
gen cumlpaub=.
gen avelpa=.
gen avelpalb=.
gen avelpaub=.
forval ii = 0/84 {
gen logf = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog = lpa*logf
gen lpaloglb = lb*logf
gen lpalogub = ub*logf
gen cumlpalog = sum(lpalog)
gen cumlpaloglb = sum(lpaloglb)
gen cumlpalogub = sum(lpalogub)
gen cumlog = sum(logf)
replace cumlpa = cumlpalog if age == `ii´
replace cumlpalb = cumlpaloglb if age == `ii´
replace cumlpaub = cumlpalogub if age == `ii´
replace avelpa = cumlpalog/cumlog if age == `ii´
replace avelpalb = cumlpaloglb/cumlog if age == `ii´
replace avelpaub = cumlpalogub/cumlog if age == `ii´
drop logf lpalog lpaloglb lpalogub cumlpalog cumlpaloglb cumlpalogub cumlog
}
keep sex age avelpa avelpalb avelpaub
keep if age >= 30
save avelpa_cal_`s´, replace
}
clear
append using avelpa_cal_0 avelpa_cal_1
save avelpa_cal, replace
```

Figure 6.3: Mean Lp(a) by age and sex – projected.




```
clear
append using lpamod_0 lpamod_1
gen lb = lpa-(1.96*errr)
gen ub = lpa+(1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line lpa age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line lpa age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(0(10)80, nogrid) ///
ytitle("Lp(a) (mg/dL)") ///
xtitle("Age")
use avelpa_cal, clear
twoway ///
(rarea avelpaub avelpalb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line avelpa age if sex == 0, color(red) lpattern(solid)) ///
(rarea avelpaub avelpalb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line avelpa age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(30(10)80, nogrid) yscale(range(21.0 22.7)) ///
ytitle("Mean cumulative Lp(a) (mg/dL)") ///
xtitle("Age")
```

Figure 6.4: Mean cumulative Lp(a) by age and sex – projected.



6.3 SBP

As for LDL-C and Lp(a), a linear regression model with spline effects of age, stratified by sex will be used to estimate mean SBP for ages 40-69, and project SBP from 70 to 85.

The assumptions used to inform the projections prior to age 40 follow. SBP is 85 mmHg at age 1 and increases linearly to 118mmHg at age 17. This is derived from Table 3 in "The Fourth Report on The Diagnosis, Evaluation, and Treatment of High Blood Pressure in Children and Adolescents" [147]. From age 17 to 40, SBP increases linearly to age 40 (the value of which is derived from the linear regression model described above) [148]. Age 40 happens to be a point of inflection for SBP values over the lifetime (very fortunate for this dataset) [148]. (This paper also shows a lower SBP among females at age 40, with a faster increase from that age than males, eventually surpassing males, consistent with the UK Biobank data as seen in Figure 6.5).

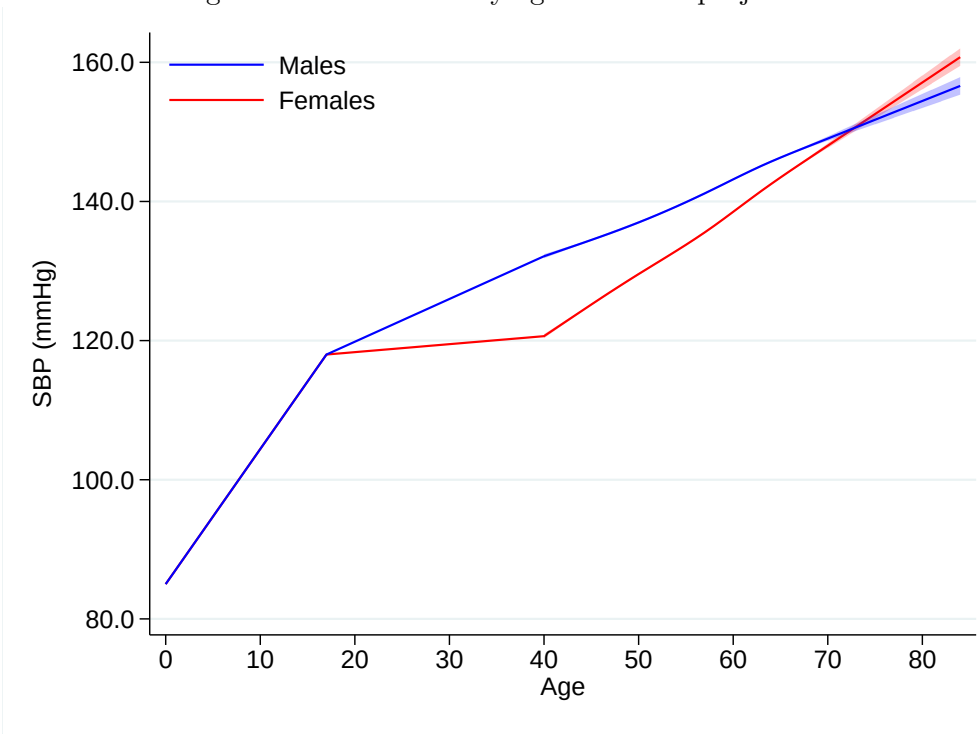
The time-weighting formula for SBP is $w = (\frac{x-2.1}{2.1})^{-2}$ [29].

The projected mean SBP across the lifespan is shown in Figure 6.5 and the mean mean cumulative SBP in Figure 6.6.

```
forval s = 0/1 {
use mainset, clear
keep if sex == `s´
gen ageda = (da-dob)/365.25
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
reg sbp agesp*
clear
set obs 85
gen age = (_n-1)+0.5
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
predict sbp if inrange(age,40,85)
predict errr if inrange(age,40,85), stdp
replace age = age-0.5
replace sbp = 85 if age == 0
replace sbp = 118  if age == 17
replace sbp = (sbp[18]-sbp[1])/17 if inrange(age,1,16)
replace sbp = (sbp[41]-sbp[18])/23 if inrange(age,18,39)
replace sbp = sum(sbp) if inrange(age,0,16)
replace sbp = sum(sbp) if inrange(age,17,39)
gen sex = `s´
keep age sex sbp errr
save sbpmod_`s´, replace
}
forval s = 0/1 {
use sbpmod_`s´, clear
gen lb = sbp-(1.96*errr)
gen ub = sbp+(1.96*errr)
replace lb = sbp if lb==.
replace ub = sbp if ub==.
gen cumsbp=.
gen cumsbplb=.
gen cumsbpub=.
gen avesbp=.
gen avesbplb=.
gen avesbpub=.
forval ii = 0/84 {
gen logf = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog = sbp*logf
gen sbploglb = lb*logf
gen sbplogub = ub*logf
gen cumsbplog = sum(sbplog)
gen cumsbploglb = sum(sbploglb)
gen cumsbplogub = sum(sbplogub)
gen cumlog = sum(logf)
replace cumsbp = cumsbplog if age == `ii´
replace cumsbplb = cumsbploglb if age == `ii´
```

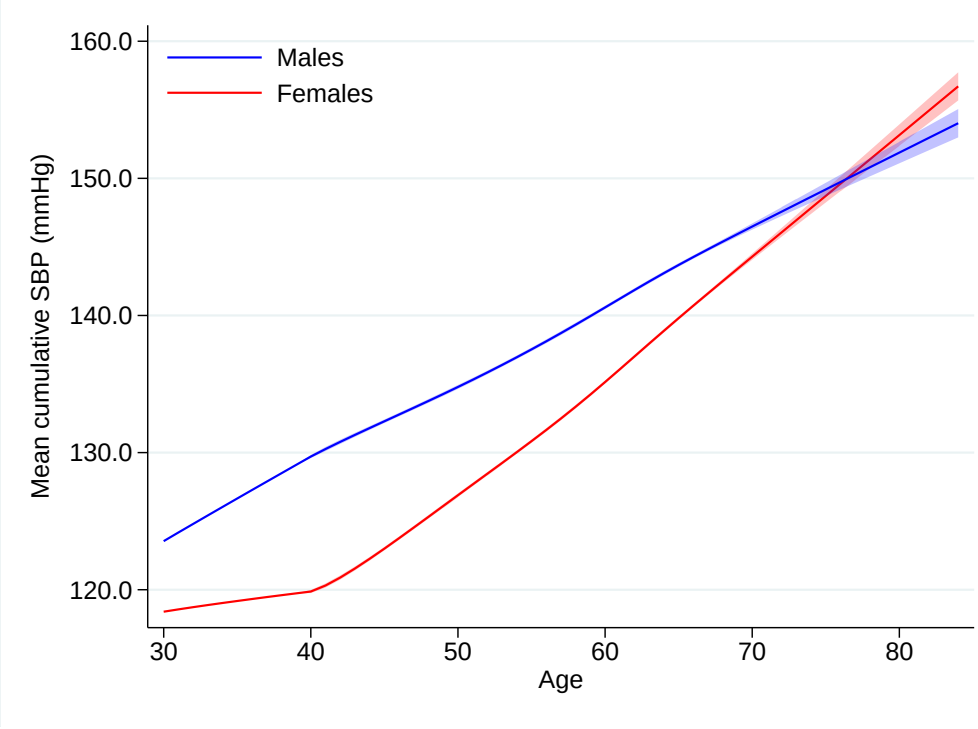
```
replace cumsbpub = cumsbplogub if age == `ii´
replace avesbp = cumsbplog/cumlog if age == `ii´
replace avesbplb = cumsbploglb/cumlog if age == `ii´
replace avesbpub = cumsbplogub/cumlog if age == `ii´
drop logf sbplog sbploglb sbplogub cumsbplog cumsbploglb cumsbplogub cumlog
}
keep sex age avesbp avesbplb avesbpub
keep if age >= 30
save avesbp_cal_`s´, replace
}
clear
append using avesbp_cal_0 avesbp_cal_1
save avesbp_cal, replace
```

Figure 6.5: Mean SBP by age and sex – projected.




```
clear
append using sbpmod_0 sbpmod_1
gen lb = sbp-(1.96*errr)
gen ub = sbp+(1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line sbp age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line sbp age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(0(10)80, nogrid) ///
ytitle("SBP (mmHg)") ///
xtitle("Age")
use avesbp_cal, clear
twoway ///
```

Figure 6.6: Mean cumulative SBP by age and sex – projected.




```
(rarea avesbpub avesbplb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line avesbp age if sex == 0, color(red) lpattern(solid)) ///
(rarea avesbpub avesbplb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line avesbp age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(, angle(0) format(%9.1f)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Mean cumulative SBP (mmHg)") ///
xtitle("Age")
```

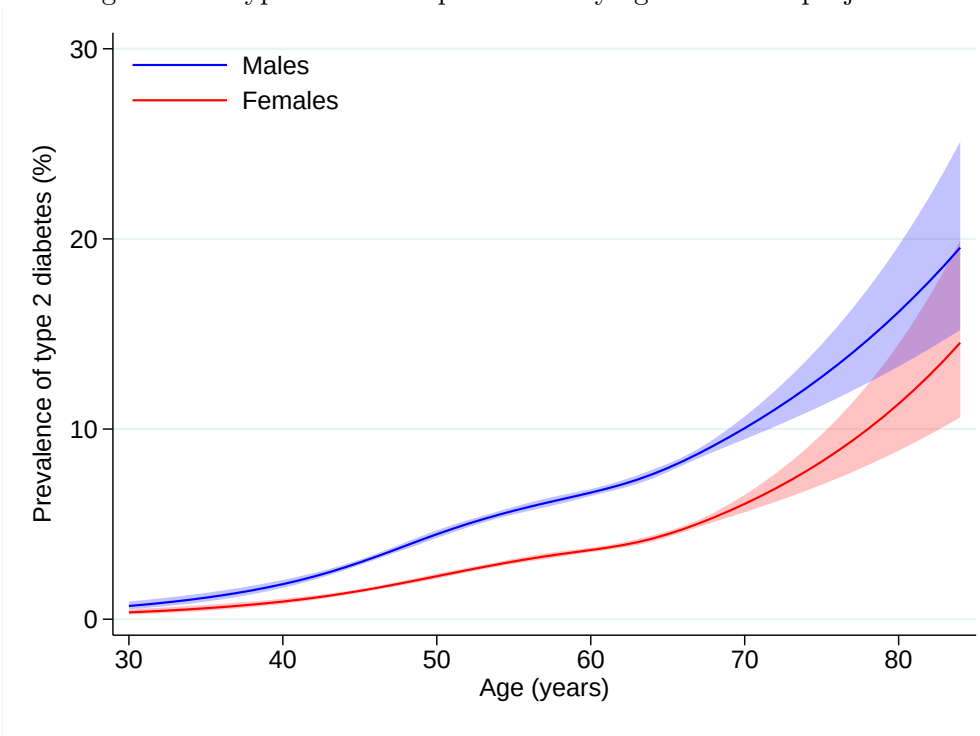
6.4 Type 2 diabetes

To derive the age-specific prevalence of type 2 diabetes, we will use a generalised linear model with the following specifications: a binomial outcome distribution; a log-link function; paramaterised with spline effects of age; and one observation per person. A limitation is that this projection (Figure 6.7) involves using data from people aged 40-69 to project the prevalence of diabetes out to age 85 years. This is not good, especially considering the prevalence of diabetes could plateau somewhere in this age range [149, 150]. Nevertheless, there isn't much else to do here without introducing other datasets (which come with their own limitations), so this seems like the least bad option. The projection to ages below 40 is less concerning, given the prevalence is just low and very unlikely to have a point of inflection in this age range [149, 150].

```
forval s = 0/1 {
use mainset, clear
keep if sex == `s´
recode DMC .=0
gen ageda = (da-dob)/365.25
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
glm DMC agesp*, family(binomial) link(log)
clear
set obs 55
gen age = (_n+29)+0.5
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
predict DMP
predict errr, stdp
replace age = age-0.5
gen sex = `s´
keep age sex DMP errr
save DMmod_`s´, replace
}


clear
append using DMmod_0 DMmod_1
save DMmod, replace
replace DMP = DMP*100
gen lb = exp(ln(DMP)-1.96*errr)
gen ub = exp(ln(DMP)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line DMP age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line DMP age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(0(10)30, angle(0) format(%9.0f)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Prevalence of type 2 diabetes (%)") ///
xtitle("Age (years)")
```

Figure 6.7: Type 2 diabetes prevalence by age and sex – projected.



6.5 LSI

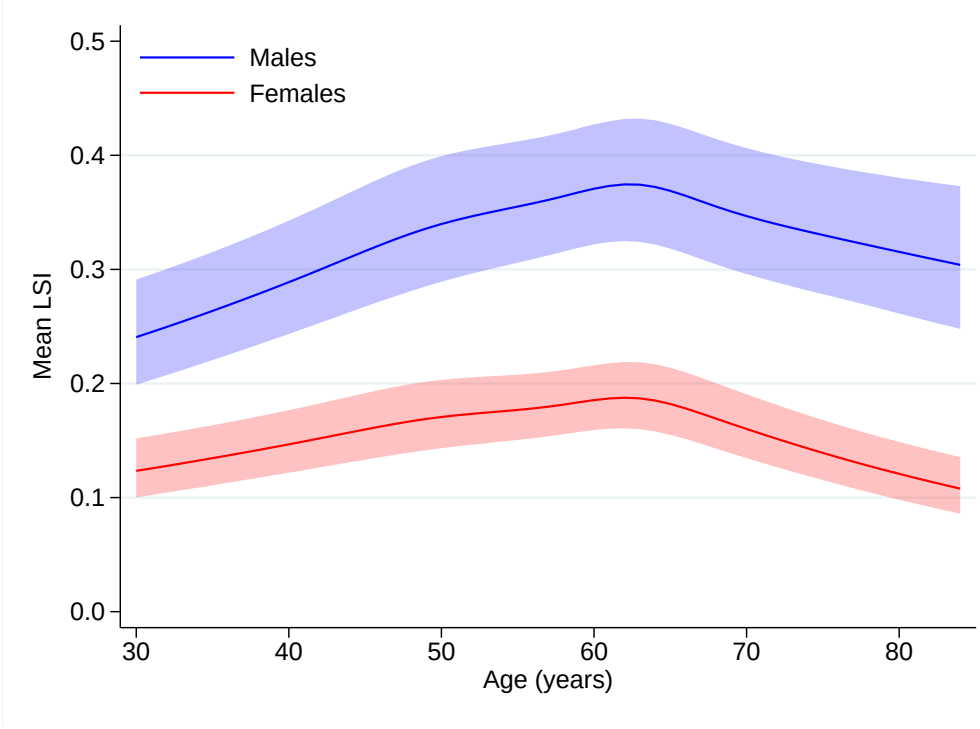
We will model the baseline LSI as a function of age, period, and cohort, and use this to estimate the mean LSI by age for the UK Biobank cohort in 2016 (to match the year used for transition probabilities). To do this we will use a generalised linear model with the following specifications: a gamma outcome distribution (the LSI is linear and non-negative); a log-link function; paramaterised with spline effects of age, period, and cohort; and one observation per person (i.e., no weighting). This will then be used to predict the age-specific LSI in 2016.

We have included period effects in this model because of the strong decline in smoking prevalence over time [151]. As with type 2 diabetes, this projection (Figure 6.8) involves using data from people aged 40-69 to project the mean LSI out to age 85 years. This is obviously not good – cigarettes have a negative impact on mortality, so we suspect the LSI could drop off much faster than the projections here. Nevertheless, as above, there isn't much to be done about this without going to other data sources, so we will accept this limitation.

```
forval s = 0/1 {
use mainset, clear
keep if sex == `s´
gen ageda = (da-dob)/365.25
gen year = year(da)
gen month = month(da)
gen dfu = (da-td(1,1,2009))/365.25
gen coh = dfu-age
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
mkspline dfusp=dfu, cubic knots(-1.5 0 1.5)
mkspline cohsp=coh, cubic knots(-68.5 -58 -40.5)
glm LSI agesp* dfusp* cohsp*, family(gamma) link(log)
clear
set obs 55
gen age = (_n+29)+0.5
gen dfu = 7
gen coh = dfu-age
mkspline agesp=age, cubic knots(42.5 50 58 63.5 68.5)
mkspline dfusp=dfu, cubic knots(-1.5 0 1.5)
mkspline cohsp=coh, cubic knots(-68.5 -58 -40.5)
predict LSI
predict errr, stdp
replace age = age-0.5
gen sex = `s´
keep age sex LSI errr
save LSImod_`s´, replace
}


clear
append using LSImod_0 LSImod_1
save LSImod, replace
gen lb = exp(ln(LSI)-1.96*errr)
gen ub = exp(ln(LSI)+1.96*errr)
twoway ///
(rarea ub lb age if sex == 0, color(red%30) fintensity(inten80) lwidth(none)) ///
(line LSI age if sex == 0, color(red) lpattern(solid)) ///
(rarea ub lb age if sex == 1, color(blue%30) fintensity(inten80) lwidth(none)) ///
(line LSI age if sex == 1, color(blue) lpattern(solid)) ///
, legend(position(11) ring(0) region(lcolor(white) color(none)) ///
order(4 "Males" ///
2 "Females") ///
cols(1)) graphregion(color(white)) ///
ylabel(0(0.1)0.5, angle(0) format(%9.1f)) ///
xlabel(30(10)80, nogrid) ///
ytitle("Mean LSI") ///
xtitle("Age (years)")
```

Figure 6.8: Mean LSI by age and sex – projected.



7 Base-case: study 1

Going back to our equation: $R_a = R \times M^{x-\mu}$ – we now have R and μ. Now we need to estimate x and R_a. However, they cannot be estimated outside of the model – they require interventions or control scenarios, which in turn require microsimulation. It is too computationally intensive to simulate the entire UK Biobank, so we will just select a random sample of 10,000 people to run the simulation on.

7.1 Setting up the model population

However, before the model begins, we will need to estimate risk factor trajectories (x) prior to model start point for each individual who enters the model. To do this, we need to make several assumptions about risk factor trajectories.

For LDL-C, the assumptions are as follows.

- LDL-C is constant from age 40 onwards in people who don't take lipid lowering therapy [152].
- Mean LDL-C is 0.75 mmol/L at birth [145], increases linearly to a mean of 2 mmol/L (standard deviation: 0.5) by age 5 (assumption based on [146]), and after this increases linearly to whatever value the individual has recorded in UK Biobank by age 40. The standard deviation for both these values is assumed to be 25% (so 0.1875 and 0.5, respectively).
- Where an individual sits on the LDL-C distribution is constant throughout life (i.e., someone in the 5th percentile of LDL-C will be in that percentile for life).

- People receiving lipid lowering therapy at baseline initiated therapy 5 years before their date of first assessment. Given how low lipid lowering therapy persistence is, [153, 154, 155] this is probably a reasonably conservative assumption.
- lipid lowering therapy lowers LDL-C by 30%. This is an assumption based on real-world studies of statin effectiveness [156, 157]. This is only prior to model start, after which the model assumptions and inputs take over.

For SBP, the assumptions are as follows.

- SBP is 85 mmHg at age 0 and increases linearly to 118 mmHg at age 17 [147].
- SBP increases linearly from age 17 until the model start age, after which it increase at the mean rate for the UK Biobank. The rate of increase is determined by calculating the linear rate of increase between the 118 mmHg value at age 17 and the value at whatever age the person enrols in UK Biobank.
- People receiving anti-hypertensive therapy at baseline had been on it for 10 years prior.
- One, two, and three antihypertensive drugs lead to SBP reductions of 6.7 mmHg, 13.3 mmHg, and 19.9 mmHg, respectively [158]. Again, this is only prior to model start, after which the model assumptions and inputs take over.

```
. use mainset, clear
. quietly su ldl
. di round(r(sd),0.01)
.86
. quietly su(sbp), detail
. di r(p99)
188


drop if ldl==.
drop if lpa ==.
drop if sbp ==.
drop if hdl ==.
gen daage = round((da-dob)/365.25,1)
keep if inrange(daage,40,69)
set seed 1312
gen rand = uniform()
sort rand
keep if _n<=10000
gen ind = _n
expand 85
bysort ind : gen age = _n-1
replace ldl = ldl/0.7 if LLT==1
rename ldl ldlm
merge m:1 sex age using ldlmod, keep(1 3) nogen
sort ind age
bysort ind (age) : gen sdldl = (ldlm-ldl[41])/0.86
replace ldl = ldl+0.1875*sdldl if age ==0
replace ldl = ldl+0.5*sdldl if age==5
replace ldl = ldlm if age >=40
bysort ind (age) : replace ldl = (ldl[6]-ldl[1])/5 if inrange(age,1,4)
bysort ind (age) : replace ldl = sum(ldl) if inrange(age,0,4)
bysort ind (age) : replace ldl = (ldl[41]-ldl[6])/35 if inrange(age,6,39)
bysort ind (age) : replace ldl = sum(ldl) if inrange(age,5,39)
replace ldl = ldl*0.7 if inrange(age,daage-6,daage-1) & LLT==1
replace ldl = . if age > daage
```

```
replace lpa =. if age > daage
replace sbp = sbp+6.7 if AHT == 1
replace sbp = sbp+13.3 if AHT == 2
replace sbp = sbp+19.9 if AHT >= 3
gen sbpo = sbp
replace sbp=. if age!=daage
replace sbp = 85 if age == 0
replace sbp = 118  if age == 17
bysort ind (age) : replace sbp = (sbp[18]-sbp[1])/17 if inrange(age,1,16)
bysort ind (age) : replace sbp = sum(sbp) if inrange(age,0,16)
bysort ind (age) : replace sbp = (sbpo-sbp[18])/(daage-17) if inrange(age,18,daage-1)
bysort ind (age) : replace sbp = sum(sbp) if inrange(age,17,daage-1)
replace sbp = sbp-6.7 if inrange(age,daage-11,daage-1) & AHT==1
replace sbp = sbp-13.3 if inrange(age,daage-11,daage-1) & AHT==2
replace sbp = sbp-19.9 if inrange(age,daage-11,daage-1) & AHT>=3
replace LSI = . if age!=daage
gen cs = 1 if tsc == 0
gen ps = 1 if tsc != 0 & tsc !=.
bysort ind (age) : replace dursmk = dursmk[_n-1]+1 if age > daage & cs == 1
bysort ind (age) : replace tsc = tsc[_n-1]+1 if age > daage & ps == 1
replace LSI = (1-(0.5^(dursmk/18)))*(0.5^(tsc/18))*ln(ncig+1) if cs == 1 | ps == 1
replace LSI = 0 if ps == . & cs == .
replace LSI = . if age < daage
gen agesta = daage-dursmk
gen agesto = daage-tsc if tsc!=0
replace dursmk = age-agesta if tsc != 0
replace tsc = age-agesto
gen cycle = age-daage if age >= daage
gen DM = 0 if cycle==0
replace DM = 1 if DMC == 1 & cycle==0
gen MI = 0 if cycle==0
gen ST = 0 if cycle==0
gen DT = 0 if cycle==0
gen DME = .
gen MIE = .
gen STE = .
gen DTE = .
replace LLT = 0
replace AHT = 0
replace LLT = . if cycle!=0
replace AHT = . if cycle!=0
replace hdl = . if age < daage
recode cs .=0
keep ind sex lpa ldl sbp LSI rand ind age cycle-DTE LLT AHT hdl cs
replace rand = uniform()
order ind sex age cs hdl ldl lpa sbp LSI cycle DM DME MI MIE ST STE DT DTE LLT AHT rand
rename LSI lsi
save modstart, replace
forval s = 0/2 {
use modstart, clear
keep if cycle == 0
drop if sex == `s´
count
matrix A = (r(N),.,.,.,.)
su age, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
count if DM==1
matrix A = (A\r(N),.,.,.,.)
su ldl, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su lpa, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
count if lpa >= 50
matrix A = (A\r(N),.,.,.,.)
count if lpa >= 70
matrix A = (A\r(N),.,.,.,.)
count if lpa >= 90
matrix A = (A\r(N),.,.,.,.)
```

```
count if lpa >= 110
matrix A = (A\r(N),.,.,.,.)
su sbp, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su lsi, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
mat A`s´ = (J(11,1,`s´),A)
}
mat A = (A0\A1\A2)
clear
svmat double A
gen njm = _n
bysort A1 (njm) : gen perc = 100*A2/A2[1] if _n == 3 | inrange(_n,6,9)
forval i = 2/6 {
bysort A1 (njm) : replace A`i´ = (A`i´*2.18)-3.83 if _n == 5
replace A`i´ = 0 if A`i´<0
}
tostring A2-A6, gen(B2 B3 B4 B5 B6) format(%9.0fc) force
tostring A2 A3 A4 A5 A6 perc, replace format(%9.1f) force
bysort A1 (njm) : gen A = B2 if _n == 1
bysort A1 (njm) : replace A = B2 + " (" + perc + "\%)" if _n == 3 | inrange(_n,6,9)
bysort A1 (njm) : replace A = B4 + " (" + B3 + ", " + B5 + "; " + B2 + ", " + B6 + ")" if _n == 2 |
> _n == 10
bysort A1 (njm) : replace A = A4 + " (" + A3 + ", " + A5 + "; " + A2 + ", " + A6 + ")" if A==""
bysort A1 (njm) : gen row = _n
keep A1 A row
reshape wide A, j(A1) i(row)
drop row
gen R1 = "N" if _n == 1
replace R1 = "Age" if _n == 2
replace R1 = "N (\%) with diabetes" if _n == 3
replace R1 = "LDL-C (mmol/L)" if _n == 4
replace R1 = "Lp(a) (nmol/L)" if _n == 5
replace R1 = "N (\%) with Lp(a) $\geq$105 nmol/L (50 mg/dL)" if _n == 6
replace R1 = "N (\%) with Lp(a) $\geq$149 nmol/L (70 mg/dL)" if _n == 7
replace R1 = "N (\%) with Lp(a) $\geq$192 nmol/L (90 mg/dL)" if _n == 8
replace R1 = "N (\%) with Lp(a) $\geq$236 nmol/L (110 mg/dL)" if _n == 9
replace R1 = "SBP (mmHg)" if _n == 10
replace R1 = "LSI" if _n == 11
order R1 A1 A0 A2
export delimited using CSV/popchar_MS.csv, delimiter(":") novarnames replace
```

Population characteristics for the random sample are shown in Table 7.1.

This data can now be put in excel and the model can be run on this population. Nevertheless, I will perform the rest of the analysis in Stata as well, so that it is reproducible.

We have also made a few other assumptions about what happens once the model begins:

- The LSI was calculated assuming people who are not smokers remain non-smokers, and smokers remain smokers.
- HDL-C remains constant over the lifetime (from the level at assessment onwards; this only effects calculation of SCORE-2 risk).
- After model start, SBP will increase at the mean rate for the sex, which are calculated below.

```
clear
append using sbpmod_0 sbpmod_1
keep if age >= 40
bysort sex (age) : gen dydx = (sbp-sbp[_n-1])
bysort sex : egen dydxm = mean(dydx)
keep if age == 40
format dydxm % 9.2f
```

Table 7.1: Baseline characteristics for the model starting population.

	Females	Males	
N	5,558	4,442	
Age	57 (50, 63; 40, 69)	58 (50, 64; 40, 69)	58 (50, 63;
N (%) with diabetes	185 (3.3%)	269 (6.1%)	454
LDL-C (mmol/L)	3.7 (3.2, 4.3; 0.9, 8.8)	3.7 (3.2, 4.3; 1.4, 8.9)	3.7 (3.2, 4.3; 0
Lp(a) (nmol/L)	21.4 (8.1, 75.9; 0.0, 551.0)	17.1 (7.0, 68.9; 0.0, 446.3)	19.3 (7.6, 72.7; 0.0
N (%) with Lp(a) ≥105 nmol/L (50 mg/dL)	1,144 (20.6%)	823 (18.5%)	1,967
N (%) with Lp(a) ≥149 nmol/L (70 mg/dL)	775 (13.9%)	521 (11.7%)	1,296
N (%) with Lp(a) ≥192 nmol/L (90 mg/dL)	445 (8.0%)	260 (5.9%)	705
N (%) with Lp(a) ≥236 nmol/L (110 mg/dL)	171 (3.1%)	98 (2.2%)	269
SBP (mmHg)	135 (122, 150; 89, 244)	142 (130, 156; 98, 226)	138 (126, 153; 8
LSI	0.0 (0.0, 0.2; 0.0, 3.1)	0.0 (0.0, 0.6; 0.0, 3.7)	0.0 (0.0, 0.4; 0

Numeric variables are presented as median (25th centile, 75th centile; minimum, maximum). Abbreviations: LDL-C – Low density lipoprotein-cholesterol; Lp(a) – Lipoprotein(a); SBP – Systolic blood pressure; LSI – Lifetime smoking index.

```
. list sex dydxm

     +-------------+
     | sex   dydxm |
     |-------------|
  1. |   0    0.91 |
  2. |   1    0.56 |
     +-------------+
```

7.2 Control scenario

First, the control scenario. Recall from section 2.1 that people are to undergo screening every 5 years, after which a treatment algorithm follows. We will start everyone at baseline with a screening check, and then repeat screening every 5 years. excel

```
use modstart, clear
quietly {
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 | `c´ == 45 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
```

```
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
merge m:1 sex age using DMmod, keep(1 3) nogen
```

```
merge m:1 sex age using LSImod, keep(1 3) nogen
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge errr
rename (rate) (rate_`i´)
}
replace rate_t2d = 0 if DM == 1
recast double rate_t2d-rate_nfICH
*replace rate_t2d = rate_t2d*(0.763^(mcldl-aveldl))
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*(2.083^(mcldl-aveldl))
replace rate_`i´MI=rate_`i´MI*(1.0054^(mclpa-avelpa))
replace rate_`i´MI=rate_`i´MI*(1.058^(mcsbp-avesbp))
replace rate_`i´MI=rate_`i´MI*(1.43^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(0.26*DMP)) if DM == 0
replace rate_`i´MI=1.26*rate_`i´MI/(1+(0.26*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*(1.08^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*(1.0035^(mclpa-avelpa))
replace rate_`i´IS=rate_`i´IS*(1.027^(mcsbp-avesbp))
replace rate_`i´IS=rate_`i´IS*(1.33^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(0.74*DMP)) if DM == 0
replace rate_`i´IS=1.74*rate_`i´IS/(1+(0.74*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*(1.039^(mcsbp-avesbp))
}
replace rate_blacdeath = rate_blacdeath*(2.52^(lsi-LSI))
replace rate_colcdeath = rate_colcdeath*(1.24^(lsi-LSI))
replace rate_oescdeath = rate_oescdeath*(3.67^(lsi-LSI))
replace rate_kidcdeath = rate_kidcdeath*(1.69^(lsi-LSI))
replace rate_luncdeath = rate_luncdeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*(37.6) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*(1.27^(lsi-LSI))
replace rate_pancdeath = rate_pancdeath*(2.13^(lsi-LSI))
replace rate_pneudeath = rate_pneudeath*(1.016^(mcsbp-avesbp))
replace rate_pneudeath = rate_pneudeath*(4.03^(lsi-LSI))
replace rate_copddeath = rate_copddeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*(37.6) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*(1.09^(mcldl-aveldl))
replace rate_pddeath = rate_pddeath*(0.48^(lsi-LSI))
recode rate_ovacdeath .=0
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
```

```
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand
}
}
gen LPAT=.
save modend0, replace
```

7.2.1 Model validation

Before moving on to the intervention, we should test that this model more or less approximates reality. To do this, we will compare the model results with those of the actual follow-up of the UK Biobank participants. While we don't expect perfect concordance (as that would require every one of our assumptions to be correct), we should come close.

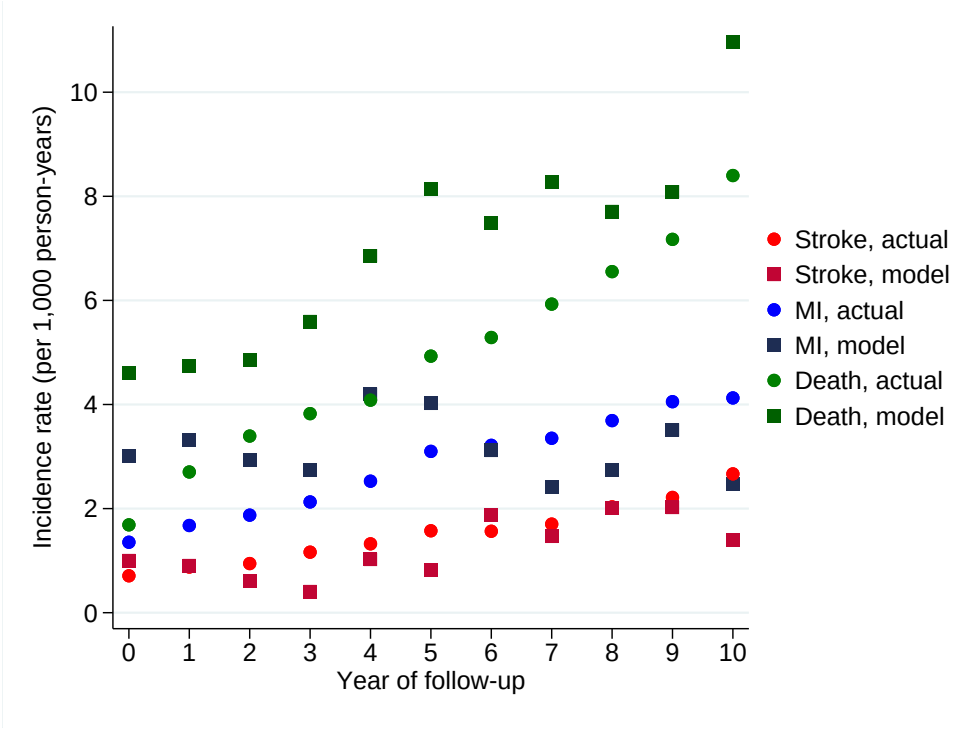
First, I will generate the actual event counts from the UK Biobank that I want to recapitulate: the incidence of MI, stroke, and death each year (measured in time since baseline assessment, to match the model). Then, I will generate the same incidence rates from the model.

```
use mainset, clear
gen faildate = td(30,9,2021) if ac == "England"
replace faildate = td(31,7,2021) if ac == "Scotland"
replace faildate = td(28,2,2018) if ac == "Wales"
replace faildate = min(dod,faildate)
gen fail = 0
stset faildate, fail(fail==1) entry(da) origin(da) scale(365.25) id(eid)
stsplit yfu, at(0(1)16)
keep eid sex da faildate dod _st-yfu
gen t0 = da+(_t0*365.25)
gen t = da+(_t*365.25)
save MT1_intervals, replace
use All_MI, clear
bysort eid (epist) : gen njm = _n
ta njm
forval i = 1/8 {
preserve
keep if njm == `i´
save MIn_`i´, replace
restore
}
```

```
forval i = 1/8 {
use MT1_intervals, clear
merge m:1 eid using MIn_`i´
keep if _merge == 3
keep if inrange(epist,t0,t)
keep eid _t0
gen MI = 1
save MIna_`i´, replace
}
clear
forval i = 1/8 {
append using MIna_`i´
}
collapse (sum) MI, by(eid _t0)
save MIa, replace
use All_IS, clear
append using All_ICH
bysort eid (epist) : gen njm = _n
ta njm
forval i = 1/6 {
preserve
keep if njm == `i´
save STn_`i´, replace
restore
}
forval i = 1/6 {
use MT1_intervals, clear
merge m:1 eid using STn_`i´
keep if _merge == 3
keep if inrange(epist,t0,t)
keep eid _t0
gen ST = 1
save STna_`i´, replace
}
clear
forval i = 1/6 {
append using STna_`i´
}
collapse (sum) ST, by(eid _t0)
save STa, replace
use mainset, clear
keep if chddeath==1
keep eid chddeath
save chddeath, replace
use mainset, clear
keep if isdeath==1 | ichdeath==1
gen stdeath = 1
keep eid stdeath
save stdeath, replace
use MT1_intervals, clear
merge 1:1 eid _t0 using MIa
drop _merge
merge 1:1 eid _t0 using STa
drop _merge
merge m:1 eid using chddeath
drop _merge
merge m:1 eid using stdeath
drop _merge
bysort eid (_t) : replace MI = 1 if _n==_N & chddeath==1
bysort eid (_t) : replace ST = 1 if _n==_N & stdeath==1
gen DT = 1 if faildate == dod
gen PY = _t-_t0
collapse (sum) MI ST DT PY, by(yfu)
gen MIa = 1000*MI/PY
gen STa = 1000*ST/PY
gen DTa = 1000*DT/PY
drop PY
drop if yfu > 10
```

```
save MT1a, replace
use modend0, clear
gen PY = 1 if DT == 0
replace PY = 0.5 if DTE==1
collapse (sum) MIE STE DTE PY, by(cycle)
rename cycle yfu
merge 1:1 yfu using MT1a, keep(3) nogen
gen MIm = 1000*MIE/PY
gen STm = 1000*STE/PY
gen DTm = 1000*DTE/PY
save MT1am, replace
```

Figure 7.1: Actual results in the UK Biobank vs. modeled results for the control scenario for the incidence of MI, stroke, and death.




```
use MT1am, clear
twoway ///
(scatter MIa yfu, col(blue) msymbol(O)) ///
(scatter MIm yfu, col(dknavy) msymbol(S)) ///
(scatter STa yfu, col(red) msymbol(O)) ///
(scatter STm yfu, col(cranberry) msymbol(S)) ///
(scatter DTa yfu, col(green) msymbol(O)) ///
(scatter DTm yfu, col(dkgreen) msymbol(S)) ///
, graphregion(color(white)) ///
ylabel(0(2)10, angle(0) format(%9.0f)) ///
ytitle("Incidence rate (per 1,000 person-years)") ///
xtitle(Year of follow-up) xlabel(0(1)10) ///
legend(order( ///
3 "Stroke, actual" ///
4 "Stroke, model" ///
1 "MI, actual" ///
2 "MI, model" ///
5 "Death, actual" ///
6 "Death, model" ///
```

```
) position(3) cols(1) region(col(none) lcolor(none)))
```

Figure 7.1 shows that our model performs pretty well, given we only have a random sample of 10,000 and necessarily made a number of assumptions that won't be reflective of reality (particularly the assumption that everyone gets treatment concordant with guidelines).

7.3 Intervention scenario

This is the same as the intervention scenario, except we add an Lp(a) test at each screening for those not at high risk.

```
use modstart, clear
quietly {
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
if `c´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70)
gen LPAT=1 if vhr!=1 & cycle == `c´
replace vhr = 1 if lpa >= 90 & cycle == `c´
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70 | lpa >= 90) & MI == 0 & ST == 0
}
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
```

```
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
merge m:1 sex age using DMmod, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge errr
rename (rate) (rate_`i´)
}
replace rate_t2d = 0 if DM == 1
recast double rate_t2d-rate_nfICH
*replace rate_t2d = rate_t2d*(0.763^(mcldl-aveldl))
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*(2.083^(mcldl-aveldl))
replace rate_`i´MI=rate_`i´MI*(1.0054^(mclpa-avelpa))
replace rate_`i´MI=rate_`i´MI*(1.058^(mcsbp-avesbp))
replace rate_`i´MI=rate_`i´MI*(1.43^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(0.26*DMP)) if DM == 0
replace rate_`i´MI=1.26*rate_`i´MI/(1+(0.26*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*(1.08^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*(1.0035^(mclpa-avelpa))
replace rate_`i´IS=rate_`i´IS*(1.027^(mcsbp-avesbp))
replace rate_`i´IS=rate_`i´IS*(1.33^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(0.74*DMP)) if DM == 0
replace rate_`i´IS=1.74*rate_`i´IS/(1+(0.74*DMP)) if DM == 1
```

```
replace rate_`i´ICH=rate_`i´ICH*(1.039^(mcsbp-avesbp))
}
replace rate_blacdeath = rate_blacdeath*(2.52^(lsi-LSI))
replace rate_colcdeath = rate_colcdeath*(1.24^(lsi-LSI))
replace rate_oescdeath = rate_oescdeath*(3.67^(lsi-LSI))
replace rate_kidcdeath = rate_kidcdeath*(1.69^(lsi-LSI))
replace rate_luncdeath = rate_luncdeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*(37.6) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*(1.27^(lsi-LSI))
replace rate_pancdeath = rate_pancdeath*(2.13^(lsi-LSI))
replace rate_pneudeath = rate_pneudeath*(1.016^(mcsbp-avesbp))
replace rate_pneudeath = rate_pneudeath*(4.03^(lsi-LSI))
replace rate_copddeath = rate_copddeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*(37.6) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*(1.09^(mcldl-aveldl))
replace rate_pddeath = rate_pddeath*(0.48^(lsi-LSI))
recode rate_ovacdeath .=0
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
```

```
}
keep ind-rand LPAT
}
}
save modend1, replace
```

7.4 Results

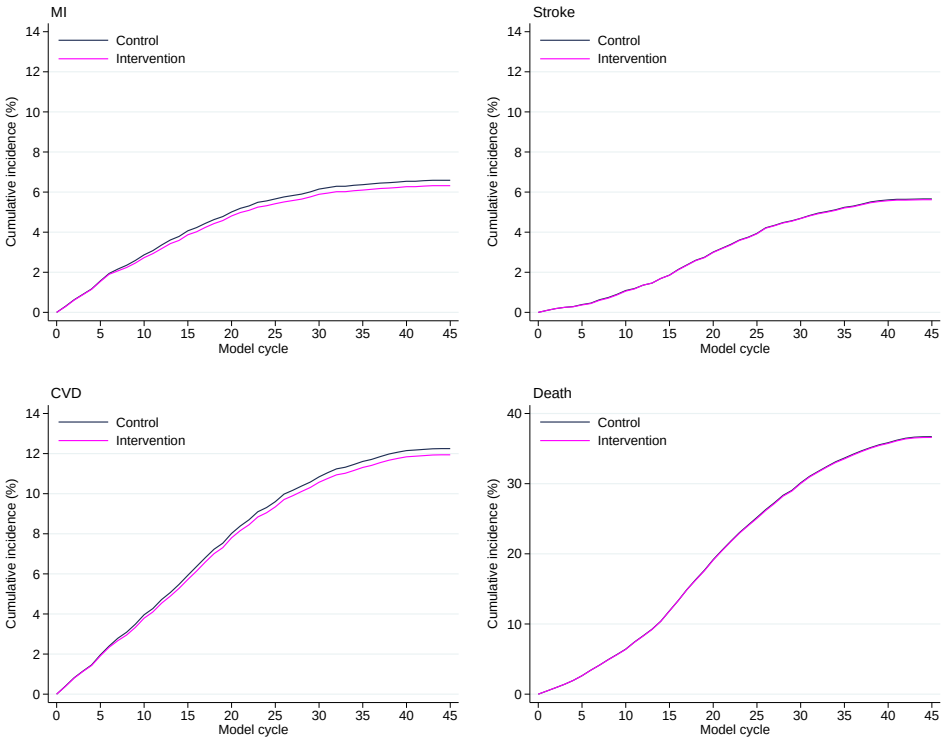
With that, we can now calculate some results. First, it's worth calculating the cumulative incidence of MI, stroke, CVD, and death under both conditions to determine the effectiveness of the intervention.

```
forval ii = 0/1 {
use modend`ii´, clear
drop if cycle ==.
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
gen CVD1 = 1 if MI1 == 1 | ST1 == 1
collapse (sum) MI1 ST1 CVD1 DTE, by(cycle)
rename DTE DT1
replace cycle = cycle+1
expand 2 if cycle == 1
bysort cycle : replace cycle = 0 if _n == 1 & cycle == 1
foreach i in MI ST CVD DT {
bysort cycle : replace `i´ = 0 if cycle == 0
gen `i´ = sum(`i´1)/100
}
gen a = `ii´
save cuminc_`ii´, replace
}
clear
append using cuminc_0 cuminc_1
rename ST Stroke
rename DT Death
foreach i in MI Stroke CVD Death {
if "`i´" == "Death" {
local ylab = "0(10)40"
}
else {
local ylab = "0(2)14"
}
twoway ///
(line `i´ cycle if a == 0, col(dknavy)) ///
(line `i´ cycle if a == 1, col(magenta)) ///
, graphregion(color(white)) ///
ylabel(`ylab´, angle(0) format(%9.0f)) ///
ytitle("Cumulative incidence (%)") ///
xtitle(Model cycle) xlabel(0(5)45) ///
legend(order( ///
1 "Control" ///
2 "Intervention" ///
) position(11) cols(1) ring(0) region(col(none) lcolor(none))) ///
title(`i´, placement(west) size(medium) col(black))
graph save "Graph" GPH/cuminc_`i´, replace
}


graph combine ///
GPH/cuminc_MI.gph ///
GPH/cuminc_Stroke.gph ///
GPH/cuminc_CVD.gph ///
GPH/cuminc_Death.gph ///
, graphregion(color(white)) cols(2) altshrink xsize(5)

quietly {
forval c = 1/2 {
forval i = 0/1 {
use modend`i´, clear
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= 90
keep if cycle!=.
```

Figure 7.2: Cumulative incidence of MI, stroke, CVD, and death in the control and invervention scenarios. Study 1.




```
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
```

```
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
```

```
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}
gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
```

```
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}
gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabres_`i´_`c´, replace
}
use tabres_0_`c´, clear
merge 1:1 _n using tabres_1_`c´, nogen
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 41
replace stat = "SICER" if _n == 42
replace val0 = . if _n>40
replace val1 = . if _n>40
replace diff = diff[38]/diff[37] if _n==41
replace diff = diff[40]/diff[37] if _n==42
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
```

```
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "YLL no CVD or diabetes" if _n == 9
replace row = "YLL with MI" if _n == 10
replace row = "YLL with stroke" if _n == 11
replace row = "YLL with diabetes" if _n == 12
replace row = "YLL with MI and stroke" if _n == 13
replace row = "YLL with MI and diabetes" if _n == 14
replace row = "YLL with stroke and diabetes" if _n == 15
replace row = "YLL with MI, stroke, and diabetes" if _n == 16
replace row = "QALY with no CVD or diabetes" if _n == 17
replace row = "QALY with MI" if _n == 18
replace row = "QALY with stroke" if _n == 19
replace row = "QALY with diabetes" if _n == 20
replace row = "QALY with MI and stroke" if _n == 21
replace row = "QALY with MI and diabetes" if _n == 22
replace row = "QALY with stroke and diabetes" if _n == 23
replace row = "QALY with MI, stroke, and diabetes" if _n == 24
replace row = "Chronic MI healthcare costs (\textsterling)" if _n == 25
replace row = "Chronic stroke healthcare costs (\textsterling)" if _n == 26
replace row = "Chronic diabetes healthcare costs (\textsterling)" if _n == 27
replace row = "Chronic MI and stroke healthcare costs (\textsterling)" if _n == 28
replace row = "Chronic MI and diabetes healthcare costs (\textsterling)" if _n == 29
replace row = "Chronic stroke and diabetes healthcare costs (\textsterling)" if _n == 30
replace row = "Chronic MI, diabetes, and stroke healthcare costs (\textsterling)" if _n == 31
replace row = "Acute MI costs (\textsterling)" if _n == 32
replace row = "Acute stroke costs (\textsterling)" if _n == 33
replace row = "Medication costs (\textsterling)" if _n == 34
replace row = "Lp(a) test costs (\textsterling)" if _n == 35
replace row = "Total YLL" if _n == 36
replace row = "Total QALY" if _n == 37
replace row = "Total healthcare costs (\textsterling)" if _n == 38
replace row = "Total indirect costs (\textsterling)" if _n == 39
replace row = "Total costs (\textsterling)" if _n == 40
replace row = "ICER (\textsterling \ per QALY)" if _n == 41
replace row = "SICER (\textsterling \ per QALY)" if _n == 42
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestab_`c´.csv, delimiter(":") novarnames replace
use tabres_0_`c´, clear
merge 1:1 _n using tabres_1_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
```

```
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
save tempsumtab_`c´, replace
export delimited using CSV/BCrestabsum_`c´.csv, delimiter(":") novarnames replace
}
clear
append using tempsumtab_1 tempsumtab_2
drop if inrange(_n,20,27)
gen C = "Overall" if _n == 1
replace C = "Australia" if _n == 9
replace C = "UK" if _n == 20
order C
export delimited using CSV/BCrestabsumpaper.csv, delimiter(":") novarnames replace
}
```

The intervention is only slightly effective for preventing MI, and far less so for stroke (Figure 7.2). This is not surprising given how few people actually receive therapy in response to Lp(a) testing (Table 7.2).

Table 7.2: Base case results, summary. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs ($)	86,896,417	85,947,239	-949,178
Acute event costs ($)	10,274,131	10,024,083	-250,048
Medication costs ($)	43,645,563	45,554,123	1,908,561
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,392	60
Total QALY	117,965	118,047	82
Total healthcare costs ($)	140,816,110	141,750,270	934,160
Total indirect costs ($)	114,253,596	112,721,465	-1,532,131
Total costs ($)	255,069,707	254,471,735	-597,972
ICER ($ per QALY)			11,369
SICER ($ per QALY)			-7,277

Table 7.3: Base case results, summary. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs (£)	74,895,806	74,089,353	-806,454
Acute event costs (£)	2,738,237	2,688,591	-49,647
Medication costs (£)	4,004,436	4,172,008	167,572
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,427	78
Total QALY	128,184	128,282	98
Total healthcare costs (£)	81,638,479	81,309,672	-328,808
Total indirect costs (£)	55,683,736	54,945,980	-737,756
Total costs (£)	137,322,215	136,255,651	-1,066,564
ICER (£ per QALY)			-3,356
SICER (£ per QALY)			-10,886

Table 7.4: Base case results, detailed. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
YLL no CVD or diabetes	120,190	120,404	214
YLL with MI	2,397	2,259	-138
YLL with stroke	893	872	-21
YLL with diabetes	17,025	17,036	12
YLL with MI and stroke	138	130	-8
YLL with MI and diabetes	418	415	-3
YLL with stroke and diabetes	252	256	4
YLL with MI, stroke, and diabetes	21	21	-0
QALY with no CVD or diabetes	103,956	104,140	184
QALY with MI	1,622	1,529	-93
QALY with stroke	484	473	-12
QALY with diabetes	11,426	11,434	8
QALY with MI and stroke	75	71	-4
QALY with MI and diabetes	260	258	-2
QALY with stroke and diabetes	130	132	2
QALY with MI, stroke, and diabetes	11	11	-0
Chronic MI healthcare costs ($)	14,911,439	14,055,071	-856,368
Chronic stroke healthcare costs ($)	3,916,576	3,824,568	-92,008
Chronic diabetes healthcare costs ($)	61,084,875	61,126,620	41,745
Chronic MI and stroke healthcare costs ($)	857,119	808,104	-49,015
Chronic MI and diabetes healthcare costs ($)	3,705,896	3,679,562	-26,333
Chronic stroke and diabetes healthcare costs ($)	2,231,903	2,268,003	36,100
Chronic MI, diabetes, and stroke healthcare costs ($)	188,609	185,310	-3,299
Acute MI costs ($)	5,586,722	5,366,888	-219,835
Acute stroke costs ($)	4,687,408	4,657,195	-30,213
Medication costs ($)	43,645,563	45,554,123	1,908,561
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,392	60
Total QALY	117,965	118,047	82
Total healthcare costs ($)	140,816,110	141,750,270	934,160
Total indirect costs ($)	114,253,596	112,721,465	-1,532,131
Total costs ($)	255,069,707	254,471,735	-597,972
ICER ($ per QALY)			11,369
SICER ($ per QALY)			-7,277

Table 7.5: Base case results, detailed. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
YLL no CVD or diabetes	138,284	138,556	272
YLL with MI	2,953	2,779	-174
YLL with stroke	1,137	1,112	-24
YLL with diabetes	20,884	20,899	16
YLL with MI and stroke	178	166	-12
YLL with MI and diabetes	548	543	-5
YLL with stroke and diabetes	337	343	6
YLL with MI, stroke, and diabetes	29	28	-1
QALY with no CVD or diabetes	112,229	112,447	218
QALY with MI	1,850	1,740	-110
QALY with stroke	566	553	-12
QALY with diabetes	12,967	12,977	10
QALY with MI and stroke	89	83	-6
QALY with MI and diabetes	311	308	-3
QALY with stroke and diabetes	158	161	2
QALY with MI, stroke, and diabetes	13	13	-0
Chronic MI healthcare costs (£)	9,063,333	8,517,679	-545,653
Chronic stroke healthcare costs (£)	8,141,867	7,971,286	-170,581
Chronic diabetes healthcare costs (£)	49,226,184	49,263,088	36,905
Chronic MI and stroke healthcare costs (£)	2,369,115	2,218,410	-150,706
Chronic MI and diabetes healthcare costs (£)	2,231,721	2,211,145	-20,575
Chronic stroke and diabetes healthcare costs (£)	3,464,934	3,520,937	56,002
Chronic MI, diabetes, and stroke healthcare costs (£)	398,652	386,807	-11,845
Acute MI costs (£)	991,235	951,665	-39,569
Acute stroke costs (£)	1,747,003	1,736,926	-10,077
Medication costs (£)	4,004,436	4,172,008	167,572
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,427	78
Total QALY	128,184	128,282	98
Total healthcare costs (£)	81,638,479	81,309,672	-328,808
Total indirect costs (£)	55,683,736	54,945,980	-737,756
Total costs (£)	137,322,215	136,255,651	-1,066,564
ICER (£ per QALY)			-3,356
SICER (£ per QALY)			-10,886

7.5 Results by age and sex.

It's worth repeating the results by age and sex.

```
quietly {
forval c = 1/2 {
forval a = 1/5 {
forval i = 0/1 {
use modend`i´, clear
gen dage = age if cycle == 0
bysort ind (age) : egen daage = min(dage)
if `a´ == 1 {
keep if sex == 0
}
if `a´ == 2 {
keep if sex == 1
}
if `a´ == 3 {
keep if inrange(daage,40,49)
}
if `a´ == 4 {
keep if inrange(daage,50,59)
}
if `a´ == 5 {
keep if inrange(daage,60,69)
}
drop dage daage
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= 90
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
```

```
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
```

```
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}


gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
```

```
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}


gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabres_`i´_`a´_`c´, replace
}
use tabres_0_`a´_`c´, clear
merge 1:1 _n using tabres_1_`a´_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
```

```
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestabsum_`a´_`c´.csv, delimiter(":") novarnames replace
}
}
}
```

Table 7.6: Base case results, summary, females. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,749	1,749
Incident MI (N)	659	585	-74
Total MIs (N)	817	715	-102
Incident stroke (N)	566	564	-2
Total strokes (N)	728	727	-1
Deaths (N)	3,670	3,642	-28
Chronic healthcare costs ($)	86,896,417	83,909,545	-2,986,872
Acute event costs ($)	10,274,131	9,463,757	-810,373
Medication costs ($)	43,645,563	49,302,438	5,656,875
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,526	193
Total QALY	117,965	118,215	251
Total healthcare costs ($)	140,816,110	142,900,565	2,084,455
Total indirect costs ($)	114,253,596	108,913,785	-5,339,811
Total costs ($)	255,069,707	251,814,351	-3,255,356
ICER ($ per QALY)			8,316
SICER ($ per QALY)			-12,987

Table 7.7: Base case results, summary, males. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,146	1,146
Incident MI (N)	659	600	-59
Total MIs (N)	817	737	-80
Incident stroke (N)	566	565	-1
Total strokes (N)	728	730	2
Deaths (N)	3,670	3,650	-20
Chronic healthcare costs ($)	86,896,417	84,254,388	-2,642,029
Acute event costs ($)	10,274,131	9,647,976	-626,154
Medication costs ($)	43,645,563	47,385,391	3,739,829
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,436	103
Total QALY	117,965	118,130	166
Total healthcare costs ($)	140,816,110	141,512,580	696,470
Total indirect costs ($)	114,253,596	110,674,253	-3,579,343
Total costs ($)	255,069,707	252,186,833	-2,882,873
ICER ($ per QALY)			4,205
SICER ($ per QALY)			-17,407

Table 7.8: Base case results, summary, people aged 40-49 at baseline. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	231	231
Incident MI (N)	659	649	-10
Total MIs (N)	817	805	-12
Incident stroke (N)	566	563	-3
Total strokes (N)	728	724	-4
Deaths (N)	3,670	3,668	-2
Chronic healthcare costs ($)	86,896,417	86,553,472	-342,945
Acute event costs ($)	10,274,131	10,181,117	-93,014
Medication costs ($)	43,645,563	44,290,247	644,684
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,363	30
Total QALY	117,965	118,002	37
Total healthcare costs ($)	140,816,110	141,249,660	433,550
Total indirect costs ($)	114,253,596	113,752,750	-500,847
Total costs ($)	255,069,707	255,002,410	-67,297
ICER ($ per QALY)			11,608
SICER ($ per QALY)			-1,802

Table 7.9: Base case results, summary, people aged 50-59 at baseline. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	690	658	-32
Total MIs (N)	833	791	-42
Incident stroke (N)	557	559	2
Total strokes (N)	708	705	-3
Deaths (N)	3,651	3,639	-12
Chronic healthcare costs ($)	91,903,921	91,822,753	-81,168
Acute event costs ($)	10,454,073	10,162,462	-291,612
Medication costs ($)	43,649,452	45,568,819	1,919,366
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,456	141,540	84
Total QALY	117,855	117,910	54
Total healthcare costs ($)	146,007,447	147,778,858	1,771,411
Total indirect costs ($)	113,957,379	112,642,383	-1,314,996
Total costs ($)	259,964,825	260,421,241	456,416
ICER ($ per QALY)			32,602
SICER ($ per QALY)			8,400

Table 7.10: Base case results, summary, people aged 60-69 at baseline. Study 1. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs ($)	191,893,379	189,599,330	-2,294,049
Acute event costs ($)	20,874,303	20,414,260	-460,043
Medication costs ($)	92,483,758	95,308,285	2,824,528
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	251,939	252,098	158
Total QALY	206,933	207,140	207
Total healthcare costs ($)	305,251,440	305,546,700	295,260
Total indirect costs ($)	187,251,908	184,593,557	-2,658,351
Total costs ($)	492,503,347	490,140,257	-2,363,090
ICER ($ per QALY)			1,425
SICER ($ per QALY)			-11,401

Table 7.11: Base case results, summary, females. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,749	1,749
Incident MI (N)	659	585	-74
Total MIs (N)	817	715	-102
Incident stroke (N)	566	564	-2
Total strokes (N)	728	727	-1
Deaths (N)	3,670	3,642	-28
Chronic healthcare costs (£)	74,895,806	72,916,796	-1,979,010
Acute event costs (£)	2,738,237	2,595,630	-142,607
Medication costs (£)	4,004,436	4,503,425	498,989
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,597	249
Total QALY	128,184	128,480	296
Total healthcare costs (£)	81,638,479	80,375,571	-1,262,909
Total indirect costs (£)	55,683,736	53,041,134	-2,642,602
Total costs (£)	137,322,215	133,416,704	-3,905,511
ICER (£ per QALY)			-4,268
SICER (£ per QALY)			-13,198

Table 7.12: Base case results, summary, males. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,146	1,146
Incident MI (N)	659	600	-59
Total MIs (N)	817	737	-80
Incident stroke (N)	566	565	-1
Total strokes (N)	728	730	2
Deaths (N)	3,670	3,650	-20
Chronic healthcare costs (£)	74,895,806	73,082,233	-1,813,573
Acute event costs (£)	2,738,237	2,633,366	-104,871
Medication costs (£)	4,004,436	4,333,221	328,785
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,484	136
Total QALY	128,184	128,380	197
Total healthcare costs (£)	81,638,479	80,408,540	-1,229,940
Total indirect costs (£)	55,683,736	53,942,937	-1,740,798
Total costs (£)	137,322,215	134,351,477	-2,970,738
ICER (£ per QALY)			-6,257
SICER (£ per QALY)			-15,112

Table 7.13: Base case results, summary, people aged 40-49 at baseline. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	231	231
Incident MI (N)	659	649	-10
Total MIs (N)	817	805	-12
Incident stroke (N)	566	563	-3
Total strokes (N)	728	724	-4
Deaths (N)	3,670	3,668	-2
Chronic healthcare costs (£)	74,895,806	74,569,510	-326,296
Acute event costs (£)	2,738,237	2,717,351	-20,887
Medication costs (£)	4,004,436	4,061,558	57,122
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,388	39
Total QALY	128,184	128,228	45
Total healthcare costs (£)	81,638,479	81,708,139	69,659
Total indirect costs (£)	55,683,736	55,437,412	-246,324
Total costs (£)	137,322,215	137,145,550	-176,665
ICER (£ per QALY)			1,564
SICER (£ per QALY)			-3,967

Table 7.14: Base case results, summary, people aged 50-59 at baseline. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	690	658	-32
Total MIs (N)	833	791	-42
Incident stroke (N)	557	559	2
Total strokes (N)	708	705	-3
Deaths (N)	3,651	3,639	-12
Chronic healthcare costs (£)	78,549,242	78,750,865	201,624
Acute event costs (£)	2,749,070	2,696,169	-52,901
Medication costs (£)	4,006,680	4,175,050	168,370
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,527	164,634	106
Total QALY	128,063	128,126	63
Total healthcare costs (£)	85,304,992	85,981,804	676,812
Total indirect costs (£)	55,608,484	55,023,043	-585,442
Total costs (£)	140,913,477	141,004,847	91,371
ICER (£ per QALY)			10,800
SICER (£ per QALY)			1,458

Table 7.15: Base case results, summary, people aged 60-69 at baseline. Study 1. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs (£)	135,096,606	133,524,412	-1,572,195
Acute event costs (£)	4,776,538	4,698,293	-78,245
Medication costs (£)	6,955,149	7,178,084	222,936
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	251,939	252,098	158
Total QALY	193,228	193,419	190
Total healthcare costs (£)	146,828,293	145,760,509	-1,067,784
Total indirect costs (£)	78,868,386	77,791,024	-1,077,362
Total costs (£)	225,696,679	223,551,533	-2,145,146
ICER (£ per QALY)			-5,608
SICER (£ per QALY)			-11,266

8 Base-case: study 2

This model will be similar to the last, except that this time we must factor in the effect of Lp(a) lowering for secondary prevention of CVD. We will assume that the risk of CVD following an event is also proportional to the cumulative mean Lp(a) over the lifetime. However, clearly this population is higher risk, so the baseline rate of CVD will be as for the secondary prevention population.

Because this model includes the secondary prevention population, we will also run it separately for people in the primary and secondary prevention populations. For the primary prevention population, we will use the same model starting population as above; for secondary prevention, we will define a new model population.

```
use fullset, clear
keep eid sex da ac dob dod ucod chddeath-otherdeath DMC ldl hdl LLT lpa lpar sbp AHT LSI dursmk tsc
> ncig mid std
drop if lpa ==.
gen daage = round((da-dob)/365.25,1)
keep if inrange(daage,40,69)
keep if mid < da | std < da
set seed 1311
gen rand = uniform()
sort rand
keep if _n<=10000
gen ind = _n
expand 85
bysort ind : gen age = _n-1
replace lpa =. if age > daage
replace LSI = . if age!=daage
gen cs = 1 if tsc == 0
gen ps = 1 if tsc != 0 & tsc !=.
bysort ind (age) : replace dursmk = dursmk[_n-1]+1 if age > daage & cs == 1
bysort ind (age) : replace tsc = tsc[_n-1]+1 if age > daage & ps == 1
replace LSI = (1-(0.5^(dursmk/18)))*(0.5^(tsc/18))*ln(ncig+1) if cs == 1 | ps == 1
replace LSI = 0 if ps == . & cs == .
replace LSI = . if age < daage
gen agesta = daage-dursmk
gen agesto = daage-tsc if tsc!=0
replace dursmk = age-agesta if tsc != 0
replace tsc = age-agesto
gen cycle = age-daage if age >= daage
gen DM = 0 if cycle==0
replace DM = 1 if DMC == 1 & cycle==0
gen MI = 0 if cycle==0
replace MI = 1 if mid < da & cycle == 0
gen ST = 0 if cycle==0
replace ST = 1 if std < da & cycle == 0
gen DT = 0 if cycle==0
gen DME = .
gen MIE = .
gen STE = .
gen DTE = .
rename LSI lsi
save modstartspbloated, replace
keep ind sex lpa lsi rand ind age cycle-DTE
replace rand = uniform()
order ind sex age lpa lsi cycle DM DME MI MIE ST STE DT DTE rand
save modstartsp, replace
forval s = 0/2 {
use modstartspbloated, clear
keep if cycle == 0
drop if sex == `s´
count
matrix A = (r(N),.,.,.,.)
su age, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
```

```
count if DM==1
matrix A = (A\r(N),.,.,.,.)
su ldl, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su lpa, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
count if lpa >= 50
matrix A = (A\r(N),.,.,.,.)
count if lpa >= 70
matrix A = (A\r(N),.,.,.,.)
count if lpa >= 90
matrix A = (A\r(N),.,.,.,.)
count if lpa >= 110
matrix A = (A\r(N),.,.,.,.)
su sbp, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
su lsi, detail
matrix A = (A\r(min),r(p25),r(p50),r(p75),r(max))
mat A`s´ = (J(11,1,`s´),A)
}
mat A = (A0\A1\A2)
clear
svmat double A
gen njm = _n
bysort A1 (njm) : gen perc = 100*A2/A2[1] if _n == 3 | inrange(_n,6,9)
forval i = 2/6 {
bysort A1 (njm) : replace A`i´ = (A`i´*2.18)-3.83 if _n == 5
replace A`i´ = 0 if A`i´<0
}
tostring A2-A6, gen(B2 B3 B4 B5 B6) format(%9.0fc) force
tostring A2 A3 A4 A5 A6 perc, replace format(%9.1f) force
bysort A1 (njm) : gen A = B2 if _n == 1
bysort A1 (njm) : replace A = B2 + " (" + perc + "\%)" if _n == 3 | inrange(_n,6,9)
bysort A1 (njm) : replace A = B4 + " (" + B3 + ", " + B5 + "; " + B2 + ", " + B6 + ")" if _n == 2 |
> _n == 10
bysort A1 (njm) : replace A = A4 + " (" + A3 + ", " + A5 + "; " + A2 + ", " + A6 + ")" if A==""
bysort A1 (njm) : gen row = _n
keep A1 A row
reshape wide A, j(A1) i(row)
drop row
gen R1 = "N" if _n == 1
replace R1 = "Age" if _n == 2
replace R1 = "N (\%) with diabetes" if _n == 3
replace R1 = "LDL-C (mmol/L)" if _n == 4
replace R1 = "Lp(a) (nmol/L)" if _n == 5
replace R1 = "N (\%) with Lp(a) $\geq$105 nmol/L (50 mg/dL)" if _n == 6
replace R1 = "N (\%) with Lp(a) $\geq$149 nmol/L (70 mg/dL)" if _n == 7
replace R1 = "N (\%) with Lp(a) $\geq$192 nmol/L (90 mg/dL)" if _n == 8
replace R1 = "N (\%) with Lp(a) $\geq$236 nmol/L (110 mg/dL)" if _n == 9
replace R1 = "SBP (mmHg)" if _n == 10
replace R1 = "LSI" if _n == 11
order R1 A1 A0 A2
export delimited using CSV/popcharsp_MS.csv, delimiter(":") novarnames replace
rename (R1 A1 A0 A2) (v1 v2 v3 v4)
save sppopchar, replace
import delimited using CSV/popchar_MS.csv, delimiter(":") clear
append using sppopchar
gen C = "Primary prevention" if _n == 1
replace C = "Secondary prevention" if _n == 12
order C
export delimited using CSV/popchar_MS2comb.csv, delimiter(":") novarnames replace
```

Population characteristics for the random sample for the secondary prevention population are shown in Table 8.1.

```
use modstart, clear
set seed 78932173
replace rand = runiform()
```

Table 8.1: Baseline characteristics for the model starting population, secondary prevention.

	Females	Males	
N	2,904	7,096	
Age	62 (57, 66; 40, 69)	63 (58, 66; 40, 69)	63 (58,
N (%) with diabetes	338 (11.6%)	1,180 (16.6%)	1,5
LDL-C (mmol/L)	2.9 (2.4, 3.5; 1.2, 8.8)	2.6 (2.2, 3.1; 0.9, 7.8)	2.7 (2.2, 3.
Lp(a) (nmol/L)	28.6 (9.2, 120.9; 0.0, 418.5)	24.5 (8.1, 108.7; 0.0, 470.3)	25.8 (8.4, 111.9;
N (%) with Lp(a) ≥105 nmol/L (50 mg/dL)	786 (27.1%)	1,824 (25.7%)	2,6
N (%) with Lp(a) ≥149 nmol/L (70 mg/dL)	606 (20.9%)	1,246 (17.6%)	1,8
N (%) with Lp(a) ≥192 nmol/L (90 mg/dL)	379 (13.1%)	699 (9.9%)	1,0
N (%) with Lp(a) ≥236 nmol/L (110 mg/dL)	125 (4.3%)	256 (3.6%)	
SBP (mmHg)	136 (124, 148; 87, 241)	137 (126, 150; 86, 232)	136 (125, 15
LSI	0.0 (0.0, 1.3; 0.0, 3.7)	0.4 (0.0, 1.6; 0.0, 3.8)	0.2 (0.0, 1.

Numeric variables are presented as median (25th centile, 75th centile; minimum, maximum).
Abbreviations: LDL-C – Low density lipoprotein-cholesterol; Lp(a) – Lipoprotein(a); SBP – Systolic blood pressure; LSI – Lifetime smoking index.

```
save modstart2, replace
quietly {
forval m = 0/1 {
use modstart2, clear
gen LPAT=.
gen LPT=.
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
```

```
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
replace vhr = 1 if MI == 1 | ST == 1
if `m´ == 1 {
replace LPAT=1 if vhr==1 & cycle == `c´
replace LPT=1 if lpa >= 90 & LPAT==1 & cycle == `c´ & vhr == 1
bysort ind (age) : replace lpa = lpa*(1-0.975) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
merge m:1 sex age using DMmod, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge errr
```

```
rename (rate) (rate_`i´)
}
replace rate_t2d = 0 if DM == 1
recast double rate_t2d-rate_nfICH
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
sort ind age
*replace rate_t2d = rate_t2d*(0.763^(mcldl-aveldl))
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*(2.083^(mcldl-aveldl))
replace rate_`i´MI=rate_`i´MI*(1.0054^(mclpa-avelpa))
replace rate_`i´MI=rate_`i´MI*(1.058^(mcsbp-avesbp))
replace rate_`i´MI=rate_`i´MI*(1.43^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(0.26*DMP)) if DM == 0
replace rate_`i´MI=1.26*rate_`i´MI/(1+(0.26*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*(1.08^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*(1.0035^(mclpa-avelpa))
replace rate_`i´IS=rate_`i´IS*(1.027^(mcsbp-avesbp))
replace rate_`i´IS=rate_`i´IS*(1.33^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(0.74*DMP)) if DM == 0
replace rate_`i´IS=1.74*rate_`i´IS/(1+(0.74*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*(1.039^(mcsbp-avesbp))
replace rate`i´MI=rate`i´MI*(1.0054^(mclpa-avelpa))
replace rate`i´S=rate`i´S*(1.0035^(mclpa-avelpa))
}
replace rate_blacdeath = rate_blacdeath*(2.52^(lsi-LSI))
replace rate_colcdeath = rate_colcdeath*(1.24^(lsi-LSI))
replace rate_oescdeath = rate_oescdeath*(3.67^(lsi-LSI))
replace rate_kidcdeath = rate_kidcdeath*(1.69^(lsi-LSI))
replace rate_luncdeath = rate_luncdeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*(37.6) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*(1.27^(lsi-LSI))
replace rate_pancdeath = rate_pancdeath*(2.13^(lsi-LSI))
replace rate_pneudeath = rate_pneudeath*(1.016^(mcsbp-avesbp))
replace rate_pneudeath = rate_pneudeath*(4.03^(lsi-LSI))
replace rate_copddeath = rate_copddeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*(37.6) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*(1.09^(mcldl-aveldl))
replace rate_pddeath = rate_pddeath*(0.48^(lsi-LSI))
recode rate_ovacdeath .=0
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
```

```
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace LPT = LPT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT LPT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save modend2_`m´, replace
}
forval m = 0/1 {
use modstartsp, clear
if `m´ == 0 {
gen LPAT=.
gen LPT=.
}
if `m´ == 1 {
gen LPAT=1 if cycle == 0
gen LPT=1 if lpa >= 90 & cycle == 0
}
forval c = 0/44 {
if `m´ == 1 {
bysort ind (age) : replace lpa = lpa*(1-0.975) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
replace mclpa = . if cycle!=`c´
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
merge m:1 age sex using INC/t2d
drop _merge errr
rename (rate) (rate_t2d)
replace rate_t2d = 0 if DM == 1
recast double rate_t2d
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate`i´MI=rate`i´MI*(1.0054^(mclpa-avelpa))
replace rate`i´S=rate`i´S*(1.0035^(mclpa-avelpa))
}
sort ind age
gen ratesum1 = 0
```

```
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
foreach var of varlist lpa lsi {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save modendsp`m´, replace
}
}
```

8.1 Results

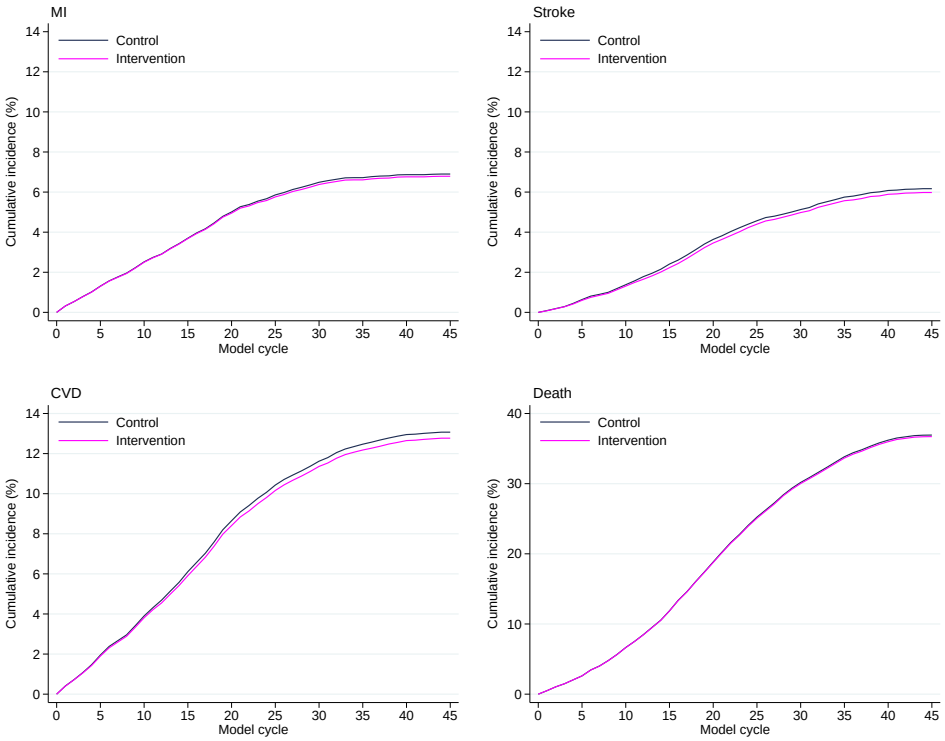
First, the cumulative incidence of MI, stroke, CVD, and death under both conditions to determine the effectiveness of the intervention.

```
forval ii = 0/1 {
use modend2_`ii´, clear
drop if cycle ==.
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
gen CVD1 = 1 if MI1 == 1 | ST1 == 1
collapse (sum) MI1 ST1 CVD1 DTE, by(cycle)
rename DTE DT1
replace cycle = cycle+1
expand 2 if cycle == 1
bysort cycle : replace cycle = 0 if _n == 1 & cycle == 1
foreach i in MI ST CVD DT {
bysort cycle : replace `i´ = 0 if cycle == 0
gen `i´ = sum(`i´1)/100
}
gen a = `ii´
save cuminc2_`ii´, replace
}
clear
append using cuminc2_0 cuminc2_1
rename ST Stroke
rename DT Death
foreach i in MI Stroke CVD Death {
if "`i´" == "Death" {
local ylab = "0(10)40"
}
else {
local ylab = "0(2)14"
}
twoway ///
(line `i´ cycle if a == 0, col(dknavy)) ///
(line `i´ cycle if a == 1, col(magenta)) ///
, graphregion(color(white)) ///
ylabel(`ylab´, angle(0) format(%9.0f)) ///
ytitle("Cumulative incidence (%)") ///
xtitle(Model cycle) xlabel(0(5)45) ///
legend(order( ///
1 "Control" ///
2 "Intervention" ///
) position(11) cols(1) ring(0) region(col(none) lcolor(none))) ///
title(`i´, placement(west) size(medium) col(black))
graph save "Graph" GPH/cuminc2_`i´, replace
}


graph combine ///
GPH/cuminc2_MI.gph ///
GPH/cuminc2_Stroke.gph ///
GPH/cuminc2_CVD.gph ///
GPH/cuminc2_Death.gph ///
, graphregion(color(white)) cols(2) altshrink xsize(5)

quietly {
forval c = 1/2 {
forval ii = 0/1 {
use modend2_`ii´, clear
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
```

Figure 8.1: Cumulative incidence of MI, stroke, CVD, and death in the control and invervention scenarios. Study 2.




```
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
```

```
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
```

```
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
```

```
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
save threshtemp_`ii´_`c´, replace
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save tabres2_`ii´_`c´, replace
}
use tabres2_0_`c´, clear
merge 1:1 _n using tabres2_1_`c´, nogen
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 41
replace stat = "SICER" if _n == 42
replace val0 = . if _n>40
replace val1 = . if _n>40
replace diff = diff[38]/diff[37] if _n==41
replace diff = diff[40]/diff[37] if _n==42
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
```

```
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "YLL no CVD or diabetes" if _n == 9
replace row = "YLL with MI" if _n == 10
replace row = "YLL with stroke" if _n == 11
replace row = "YLL with diabetes" if _n == 12
replace row = "YLL with MI and stroke" if _n == 13
replace row = "YLL with MI and diabetes" if _n == 14
replace row = "YLL with stroke and diabetes" if _n == 15
replace row = "YLL with MI, stroke, and diabetes" if _n == 16
replace row = "QALY with no CVD or diabetes" if _n == 17
replace row = "QALY with MI" if _n == 18
replace row = "QALY with stroke" if _n == 19
replace row = "QALY with diabetes" if _n == 20
replace row = "QALY with MI and stroke" if _n == 21
replace row = "QALY with MI and diabetes" if _n == 22
replace row = "QALY with stroke and diabetes" if _n == 23
replace row = "QALY with MI, stroke, and diabetes" if _n == 24
replace row = "Chronic MI healthcare costs (\textsterling)" if _n == 25
replace row = "Chronic stroke healthcare costs (\textsterling)" if _n == 26
replace row = "Chronic diabetes healthcare costs (\textsterling)" if _n == 27
replace row = "Chronic MI and stroke healthcare costs (\textsterling)" if _n == 28
replace row = "Chronic MI and diabetes healthcare costs (\textsterling)" if _n == 29
replace row = "Chronic stroke and diabetes healthcare costs (\textsterling)" if _n == 30
replace row = "Chronic MI, diabetes, and stroke healthcare costs (\textsterling)" if _n == 31
replace row = "Acute MI costs (\textsterling)" if _n == 32
replace row = "Acute stroke costs (\textsterling)" if _n == 33
replace row = "Medication costs (\textsterling)" if _n == 34
replace row = "Lp(a) test costs (\textsterling)" if _n == 35
replace row = "Total YLL" if _n == 36
replace row = "Total QALY" if _n == 37
replace row = "Total healthcare costs (\textsterling)" if _n == 38
replace row = "Total indirect costs (\textsterling)" if _n == 39
replace row = "Total costs (\textsterling)" if _n == 40
replace row = "ICER (\textsterling \ per QALY)" if _n == 41
replace row = "SICER (\textsterling \ per QALY)" if _n == 42
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestab2_`c´.csv, delimiter(":") novarnames replace
use tabres2_0_`c´, clear
merge 1:1 _n using tabres2_1_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
```

```
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
save tempsumtab2_`c´, replace
export delimited using CSV/BCrestabsum2_`c´.csv, delimiter(":") novarnames replace
forval i = 0/1 {
use modendsp`i´, clear
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
```

```
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}

if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
```

```
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}


gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
```

```
> ==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
save threshtempsp_`i´_`c´, replace
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 38
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabressp_`i´_`c´, replace
}
use tabressp_0_`c´, clear
merge 1:1 _n using tabressp_1_`c´, nogen
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 39
replace stat = "SICER" if _n == 40
replace val0 = . if _n>38
replace val1 = . if _n>38
replace diff = diff[36]/diff[35] if _n==39
replace diff = diff[38]/diff[35] if _n==40
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Total MIs (N)" if _n == 4
replace row = "Total strokes (N)" if _n == 5
replace row = "Deaths (N)" if _n == 6
replace row = "YLL no CVD or diabetes" if _n == 7
replace row = "YLL with MI" if _n == 8
replace row = "YLL with stroke" if _n == 9
replace row = "YLL with diabetes" if _n == 10
replace row = "YLL with MI and stroke" if _n == 11
replace row = "YLL with MI and diabetes" if _n == 12
```

```
replace row = "YLL with stroke and diabetes" if _n == 13
replace row = "YLL with MI, stroke, and diabetes" if _n == 14
replace row = "QALY with no CVD or diabetes" if _n == 15
replace row = "QALY with MI" if _n == 16
replace row = "QALY with stroke" if _n == 17
replace row = "QALY with diabetes" if _n == 18
replace row = "QALY with MI and stroke" if _n == 19
replace row = "QALY with MI and diabetes" if _n == 20
replace row = "QALY with stroke and diabetes" if _n == 21
replace row = "QALY with MI, stroke, and diabetes" if _n == 22
replace row = "Chronic MI healthcare costs (\textsterling)" if _n == 23
replace row = "Chronic stroke healthcare costs (\textsterling)" if _n == 24
replace row = "Chronic diabetes healthcare costs (\textsterling)" if _n == 25
replace row = "Chronic MI and stroke healthcare costs (\textsterling)" if _n == 26
replace row = "Chronic MI and diabetes healthcare costs (\textsterling)" if _n == 27
replace row = "Chronic stroke and diabetes healthcare costs (\textsterling)" if _n == 28
replace row = "Chronic MI, diabetes, and stroke healthcare costs (\textsterling)" if _n == 29
replace row = "Acute MI costs (\textsterling)" if _n == 30
replace row = "Acute stroke costs (\textsterling)" if _n == 31
replace row = "Medication costs (\textsterling)" if _n == 32
replace row = "Lp(a) test costs (\textsterling)" if _n == 33
replace row = "Total YLL" if _n == 34
replace row = "Total QALY" if _n == 35
replace row = "Total healthcare costs (\textsterling)" if _n == 36
replace row = "Total indirect costs (\textsterling)" if _n == 37
replace row = "Total costs (\textsterling)" if _n == 38
replace row = "ICER (\textsterling \ per QALY)" if _n == 39
replace row = "SICER (\textsterling \ per QALY)" if _n == 40
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestabsp_`c´.csv, delimiter(":") novarnames replace
use tabressp_0_`c´, clear
merge 1:1 _n using tabressp_1_`c´, nogen
drop if inrange(_n,7,22)
replace val0 = val0[7]+val0[8]+val0[9]+val0[10]+val0[11]+val0[12]+val0[13] if _n == 13
replace val0 = val0[14]+val0[15] if _n == 15
replace val1 = val1[7]+val1[8]+val1[9]+val1[10]+val1[11]+val1[12]+val1[13] if _n == 13
replace val1 = val1[14]+val1[15] if _n == 15
drop if inrange(_n,7,12) | _n == 14
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 17
replace stat = "SICER" if _n == 18
replace val0 = . if _n>15
replace val1 = . if _n>15
replace diff = diff[13]/diff[12] if _n==16
replace diff = diff[15]/diff[12] if _n==17
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Total MIs (N)" if _n == 4
replace row = "Total strokes (N)" if _n == 5
replace row = "Deaths (N)" if _n == 6
replace row = "Chronic healthcare costs (\textsterling)" if _n == 7
replace row = "Acute event costs (\textsterling)" if _n == 8
replace row = "Medication costs (\textsterling)" if _n == 9
replace row = "Lp(a) test costs (\textsterling)" if _n == 10
replace row = "Total YLL" if _n == 11
replace row = "Total QALY" if _n == 12
replace row = "Total healthcare costs (\textsterling)" if _n == 13
replace row = "Total indirect costs (\textsterling)" if _n == 14
```

```
replace row = "Total costs (\textsterling)" if _n == 15
replace row = "ICER (\textsterling \ per QALY)" if _n == 16
replace row = "SICER (\textsterling \ per QALY)" if _n == 17
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
save tempsumtab2sp_`c´, replace
export delimited using CSV/BCrestabspsum_`c´.csv, delimiter(":") novarnames replace
}
clear
append using tempsumtab2_1 tempsumtab2_2
drop if inrange(_n,20,27)
gen C = "Overall" if _n == 1
replace C = "Australia" if _n == 9
replace C = "UK" if _n == 20
order C
export delimited using CSV/BCrestabsumpaper2_PP.csv, delimiter(":") novarnames replace
clear
append using tempsumtab2sp_1 tempsumtab2sp_2
drop if inrange(_n,18,23)
gen C = "Overall" if _n == 1
replace C = "Australia" if _n == 7
replace C = "UK" if _n == 18
order C
export delimited using CSV/BCrestabsumpaper2_SP.csv, delimiter(":") novarnames replace
}
```

The intervention is only slightly effective for preventing MI, and far less so for stroke (Figure 8.1). This is not surprising given how few people actually receive therapy in response to Lp(a) testing (Table 8.2).

Table 8.2: Base case results, summary. Primary prevention population. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs ($)	87,578,365	87,210,619	-367,746
Acute event costs ($)	10,911,055	10,510,540	-400,515
Medication costs ($)	43,333,417	64,383,010	21,049,593
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,297	39
Total QALY	117,856	117,903	48
Total healthcare costs ($)	141,822,837	162,252,428	20,429,591
Total indirect costs ($)	116,754,059	116,430,698	-323,361
Total costs ($)	258,576,897	278,683,126	20,106,229
ICER ($ per QALY)			429,756
SICER ($ per QALY)			422,953

Table 8.3: Base case results, detailed. Primary prevention population. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
YLL no CVD or diabetes	120,114	120,187	73
YLL with MI	2,426	2,409	-17
YLL with stroke	1,080	1,044	-37
YLL with diabetes	16,751	16,800	50
YLL with MI and stroke	86	90	5
YLL with MI and diabetes	408	402	-6
YLL with stroke and diabetes	371	342	-29
YLL with MI, stroke, and diabetes	22	22	-0
QALY with no CVD or diabetes	103,876	103,938	61
QALY with MI	1,643	1,631	-11
QALY with stroke	590	570	-20
QALY with diabetes	11,242	11,275	33
QALY with MI and stroke	47	49	2
QALY with MI and diabetes	253	249	-4
QALY with stroke and diabetes	193	178	-15
QALY with MI, stroke, and diabetes	12	12	-0
Chronic MI healthcare costs ($)	15,092,439	14,987,585	-104,853
Chronic stroke healthcare costs ($)	4,740,806	4,579,883	-160,923
Chronic diabetes healthcare costs ($)	60,101,156	60,279,124	177,967
Chronic MI and stroke healthcare costs ($)	533,085	561,337	28,252
Chronic MI and diabetes healthcare costs ($)	3,622,093	3,567,796	-54,297
Chronic stroke and diabetes healthcare costs ($)	3,291,184	3,037,858	-253,326
Chronic MI, diabetes, and stroke healthcare costs ($)	197,602	197,035	-567
Acute MI costs ($)	5,478,867	5,383,356	-95,511
Acute stroke costs ($)	5,432,188	5,127,183	-305,004
Medication costs ($)	43,333,417	64,383,010	21,049,593
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,297	39
Total QALY	117,856	117,903	48
Total healthcare costs ($)	141,822,837	162,252,428	20,429,591
Total indirect costs ($)	116,754,059	116,430,698	-323,361
Total costs ($)	258,576,897	278,683,126	20,106,229
ICER ($ per QALY)			429,756
SICER ($ per QALY)			422,953

Table 8.4: Base case results, summary. Secondary prevention population. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	1,078	1,078
Total MIs (N)	3,275	3,099	-176
Total strokes (N)	2,976	2,910	-66
Deaths (N)	6,725	6,664	-61
Chronic healthcare costs ($)	685,367,549	687,843,080	2,475,531
Acute event costs ($)	49,064,556	47,487,200	-1,577,356
Medication costs ($)	0	4,700,080	4,700,080
Lp(a) test costs ($)	0	250,000	250,000
Total YLL	108,441	108,826	385
Total QALY	67,737	67,982	245
Total healthcare costs ($)	734,432,105	740,280,360	5,848,254
Total indirect costs ($)	914,129,163	910,390,824	-3,738,339
Total costs ($)	1,648,561,269	1,650,671,184	2,109,915
ICER ($ per QALY)			23,870
SICER ($ per QALY)			8,612

Table 8.5: Base case results, detailed. Secondary prevention population. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	1,078	1,078
Total MIs (N)	3,275	3,099	-176
Total strokes (N)	2,976	2,910	-66
Deaths (N)	6,725	6,664	-61
YLL no CVD or diabetes	0	0	0
YLL with MI	47,743	48,017	274
YLL with stroke	30,185	30,282	97
YLL with diabetes	0	0	0
YLL with MI and stroke	5,587	5,502	-85
YLL with MI and diabetes	16,105	16,102	-3
YLL with stroke and diabetes	6,837	6,910	73
YLL with MI, stroke, and diabetes	1,985	2,013	28
QALY with no CVD or diabetes	0	0	0
QALY with MI	32,796	32,981	185
QALY with stroke	16,922	16,977	55
QALY with diabetes	0	0	0
QALY with MI and stroke	3,133	3,086	-47
QALY with MI and diabetes	10,172	10,171	-2
QALY with stroke and diabetes	3,649	3,687	38
QALY with MI, stroke, and diabetes	1,064	1,079	14
Chronic MI healthcare costs ($)	297,054,282	298,759,989	1,705,707
Chronic stroke healthcare costs ($)	132,450,065	132,877,496	427,432
Chronic diabetes healthcare costs ($)	0	0	0
Chronic MI and stroke healthcare costs ($)	34,760,552	34,232,824	-527,728
Chronic MI and diabetes healthcare costs ($)	142,849,088	142,826,153	-22,935
Chronic stroke and diabetes healthcare costs ($)	60,646,699	61,294,542	647,844
Chronic MI, diabetes, and stroke healthcare costs ($)	17,606,864	17,852,075	245,211
Acute MI costs ($)	22,410,100	21,384,683	-1,025,417
Acute stroke costs ($)	26,654,456	26,102,517	-551,939
Medication costs ($)	0	4,700,080	4,700,080
Lp(a) test costs ($)	0	250,000	250,000
Total YLL	108,441	108,826	385
Total QALY	67,737	67,982	245
Total healthcare costs ($)	734,432,105	740,280,360	5,848,254
Total indirect costs ($)	914,129,163	910,390,824	-3,738,339
Total costs ($)	1,648,561,269	1,650,671,184	2,109,915
ICER ($ per QALY)			23,870
SICER ($ per QALY)			8,612

Table 8.6: Base case results, summary. Primary prevention population. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs (£)	76,542,513	76,011,410	-531,103
Acute event costs (£)	2,975,549	2,849,364	-126,184
Medication costs (£)	3,975,212	27,523,857	23,548,645
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,307	52
Total QALY	128,058	128,112	55
Total healthcare costs (£)	83,493,274	106,653,325	23,160,051
Total indirect costs (£)	57,084,779	56,925,554	-159,225
Total costs (£)	140,578,053	163,578,879	23,000,826
ICER (£ per QALY)			422,978
SICER (£ per QALY)			420,070

Table 8.7: Base case results, detailed. Primary prevention population. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
YLL no CVD or diabetes	138,150	138,240	90
YLL with MI	3,019	2,995	-23
YLL with stroke	1,353	1,309	-44
YLL with diabetes	20,579	20,642	63
YLL with MI and stroke	115	122	7
YLL with MI and diabetes	532	524	-8
YLL with stroke and diabetes	478	444	-34
YLL with MI, stroke, and diabetes	30	30	0
QALY with no CVD or diabetes	112,110	112,179	69
QALY with MI	1,889	1,875	-14
QALY with stroke	677	656	-21
QALY with diabetes	12,783	12,822	39
QALY with MI and stroke	57	61	4
QALY with MI and diabetes	301	297	-4
QALY with stroke and diabetes	226	209	-17
QALY with MI, stroke, and diabetes	14	14	0
Chronic MI healthcare costs (£)	9,212,916	9,141,350	-71,566
Chronic stroke healthcare costs (£)	9,742,831	9,428,164	-314,667
Chronic diabetes healthcare costs (£)	48,582,834	48,722,582	139,749
Chronic MI and stroke healthcare costs (£)	1,491,468	1,590,418	98,950
Chronic MI and diabetes healthcare costs (£)	2,192,746	2,162,983	-29,763
Chronic stroke and diabetes healthcare costs (£)	4,931,579	4,578,211	-353,368
Chronic MI, diabetes, and stroke healthcare costs (£)	388,140	387,703	-436
Acute MI costs (£)	986,970	968,765	-18,205
Acute stroke costs (£)	1,988,578	1,880,599	-107,980
Medication costs (£)	3,975,212	27,523,857	23,548,645
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,307	52
Total QALY	128,058	128,112	55
Total healthcare costs (£)	83,493,274	106,653,325	23,160,051
Total indirect costs (£)	57,084,779	56,925,554	-159,225
Total costs (£)	140,578,053	163,578,879	23,000,826
ICER (£ per QALY)			422,978
SICER (£ per QALY)			420,070

Table 8.8: Base case results, summary. Secondary prevention population. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	1,078	1,078
Total MIs (N)	3,275	3,099	-176
Total strokes (N)	2,976	2,910	-66
Deaths (N)	6,725	6,664	-61
Chronic healthcare costs (£)	667,416,844	669,372,846	1,956,001
Acute event costs (£)	12,911,141	12,535,272	-375,868
Medication costs (£)	0	4,285,050	4,285,050
Lp(a) test costs (£)	0	400,000	400,000
Total YLL	121,593	122,056	464
Total QALY	70,620	70,892	272
Total healthcare costs (£)	680,327,985	686,593,168	6,265,183
Total indirect costs (£)	416,096,120	414,397,455	-1,698,665
Total costs (£)	1,096,424,105	1,100,990,623	4,566,518
ICER (£ per QALY)			23,056
SICER (£ per QALY)			16,805

Table 8.9: Base case results, detailed. Secondary prevention population. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	1,078	1,078
Total MIs (N)	3,275	3,099	-176
Total strokes (N)	2,976	2,910	-66
Deaths (N)	6,725	6,664	-61
YLL no CVD or diabetes	0	0	0
YLL with MI	53,116	53,440	324
YLL with stroke	33,477	33,593	115
YLL with diabetes	0	0	0
YLL with MI and stroke	6,342	6,237	-104
YLL with MI and diabetes	18,494	18,496	3
YLL with stroke and diabetes	7,857	7,950	93
YLL with MI, stroke, and diabetes	2,307	2,340	33
QALY with no CVD or diabetes	0	0	0
QALY with MI	33,924	34,126	202
QALY with stroke	17,573	17,634	61
QALY with diabetes	0	0	0
QALY with MI and stroke	3,300	3,248	-52
QALY with MI and diabetes	10,790	10,791	1
QALY with stroke and diabetes	3,890	3,935	45
QALY with MI, stroke, and diabetes	1,143	1,158	15
Chronic MI healthcare costs (£)	159,036,022	160,033,098	997,076
Chronic stroke healthcare costs (£)	240,482,667	241,322,681	840,014
Chronic diabetes healthcare costs (£)	0	0	0
Chronic MI and stroke healthcare costs (£)	82,677,119	81,381,425	-1,295,694
Chronic MI and diabetes healthcare costs (£)	74,443,556	74,458,826	15,271
Chronic stroke and diabetes healthcare costs (£)	80,982,430	81,941,314	958,885
Chronic MI, diabetes, and stroke healthcare costs (£)	29,795,051	30,235,500	440,449
Acute MI costs (£)	3,869,598	3,687,097	-182,501
Acute stroke costs (£)	9,041,543	8,848,176	-193,367
Medication costs (£)	0	4,285,050	4,285,050
Lp(a) test costs (£)	0	400,000	400,000
Total YLL	121,593	122,056	464
Total QALY	70,620	70,892	272
Total healthcare costs (£)	680,327,985	686,593,168	6,265,183
Total indirect costs (£)	416,096,120	414,397,455	-1,698,665
Total costs (£)	1,096,424,105	1,100,990,623	4,566,518
ICER (£ per QALY)			23,056
SICER (£ per QALY)			16,805

8.2 Results by age and sex

```
quietly {
forval c = 1/2 {
forval a = 1/5 {
forval ii = 0/1 {
use modend2_`ii´, clear
gen dage = age if cycle == 0
bysort ind (age) : egen daage = min(dage)
if `a´ == 1 {
keep if sex == 0
}
if `a´ == 2 {
keep if sex == 1
}
if `a´ == 3 {
keep if inrange(daage,40,49)
}
if `a´ == 4 {
keep if inrange(daage,50,59)
}
if `a´ == 5 {
keep if inrange(daage,60,69)
}
drop dage daage
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}

sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
```

```
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
```

```
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
```

```
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save tabres2_`ii´_`a´_`c´, replace
}
use tabres2_0_`a´_`c´, clear
merge 1:1 _n using tabres2_1_`a´_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
```

```
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestabsum2_`a´_`c´.csv, delimiter(":") novarnames replace
}
forval a = 1/5 {
forval i = 0/1 {
use modendsp`i´, clear
gen dage = age if cycle == 0
bysort ind (age) : egen daage = min(dage)
if `a´ == 1 {
keep if sex == 0
}
if `a´ == 2 {
keep if sex == 1
}
if `a´ == 3 {
keep if inrange(daage,40,49)
}
if `a´ == 4 {
keep if inrange(daage,50,59)
}
if `a´ == 5 {
keep if inrange(daage,60,69)
}
drop dage daage
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.

if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}

sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
```

```
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
```

```
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
```

```
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 38
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabressp_`i´_`a´_`c´, replace
}
use tabressp_0_`a´_`c´, clear
merge 1:1 _n using tabressp_1_`a´_`c´, nogen
drop if inrange(_n,7,22)
```

```
replace val0 = val0[7]+val0[8]+val0[9]+val0[10]+val0[11]+val0[12]+val0[13] if _n == 13
replace val0 = val0[14]+val0[15] if _n == 15
replace val1 = val1[7]+val1[8]+val1[9]+val1[10]+val1[11]+val1[12]+val1[13] if _n == 13
replace val1 = val1[14]+val1[15] if _n == 15
drop if inrange(_n,7,12) | _n == 14
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 17
replace stat = "SICER" if _n == 18
replace val0 = . if _n>15
replace val1 = . if _n>15
replace diff = diff[13]/diff[12] if _n==16
replace diff = diff[15]/diff[12] if _n==17
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Total MIs (N)" if _n == 4
replace row = "Total strokes (N)" if _n == 5
replace row = "Deaths (N)" if _n == 6
replace row = "Chronic healthcare costs (\textsterling)" if _n == 7
replace row = "Acute event costs (\textsterling)" if _n == 8
replace row = "Medication costs (\textsterling)" if _n == 9
replace row = "Lp(a) test costs (\textsterling)" if _n == 10
replace row = "Total YLL" if _n == 11
replace row = "Total QALY" if _n == 12
replace row = "Total healthcare costs (\textsterling)" if _n == 13
replace row = "Total indirect costs (\textsterling)" if _n == 14
replace row = "Total costs (\textsterling)" if _n == 15
replace row = "ICER (\textsterling \ per QALY)" if _n == 16
replace row = "SICER (\textsterling \ per QALY)" if _n == 17
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/BCrestabspsum_`a´_`c´.csv, delimiter(":") novarnames replace
}
}
}
```

Table 8.10: Base case results, summary. Primary prevention population, females. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,888	1,888
Incident MI (N)	690	659	-31
Total MIs (N)	844	793	-51
Incident stroke (N)	617	590	-27
Total strokes (N)	797	741	-56
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs ($)	87,578,365	86,865,345	-713,021
Acute event costs ($)	10,911,055	10,290,702	-620,353
Medication costs ($)	43,333,417	100,135,563	56,802,146
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,293	36
Total QALY	117,856	117,907	51
Total healthcare costs ($)	141,822,837	197,439,869	55,617,032
Total indirect costs ($)	116,754,059	116,550,952	-203,107
Total costs ($)	258,576,897	313,990,822	55,413,925
ICER ($ per QALY)			1,086,931
SICER ($ per QALY)			1,082,962

Table 8.11: Base case results, summary. Primary prevention population, males. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,240	1,240
Incident MI (N)	690	665	-25
Total MIs (N)	844	799	-45
Incident stroke (N)	617	600	-17
Total strokes (N)	797	755	-42
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs ($)	87,578,365	86,990,370	-587,996
Acute event costs ($)	10,911,055	10,388,414	-522,641
Medication costs ($)	43,333,417	81,257,717	37,924,300
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,298	40
Total QALY	117,856	117,905	50
Total healthcare costs ($)	141,822,837	178,784,760	36,961,923
Total indirect costs ($)	116,754,059	116,514,675	-239,384
Total costs ($)	258,576,897	295,299,435	36,722,539
ICER ($ per QALY)			744,864
SICER ($ per QALY)			740,040

Table 8.12: Base case results, summary. Primary prevention population, people aged 40-49 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	251	251
Incident MI (N)	690	684	-6
Total MIs (N)	844	834	-10
Incident stroke (N)	617	600	-17
Total strokes (N)	797	765	-32
Deaths (N)	3,693	3,683	-10
Chronic healthcare costs ($)	87,578,365	87,350,375	-227,991
Acute event costs ($)	10,911,055	10,647,689	-263,366
Medication costs ($)	43,333,417	51,202,572	7,869,155
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,277	19
Total QALY	117,856	117,882	27
Total healthcare costs ($)	141,822,837	149,348,895	7,526,058
Total indirect costs ($)	116,754,059	116,616,577	-137,482
Total costs ($)	258,576,897	265,965,473	7,388,576
ICER ($ per QALY)			280,503
SICER ($ per QALY)			275,378

Table 8.13: Base case results, summary. Primary prevention population, people aged 50-59 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs ($)	194,258,303	193,637,820	-620,483
Acute event costs ($)	22,267,643	21,467,868	-799,775
Medication costs ($)	91,737,770	137,004,338	45,266,568
Lp(a) test costs ($)	0	237,975	237,975
Total YLL	251,750	251,860	110
Total QALY	206,652	206,766	113
Total healthcare costs ($)	308,263,716	352,348,000	44,084,284
Total indirect costs ($)	191,005,719	190,560,137	-445,582
Total costs ($)	499,269,435	542,908,138	43,638,702
ICER ($ per QALY)			389,458
SICER ($ per QALY)			385,521

Table 8.14: Base case results, summary. Primary prevention population, people aged 60-69 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	4,384	4,384	0
Lp(a) tests	0	4,238	4,238
Treatment modified in response to Lp(a) test	0	337	337
Incident MI (N)	263	260	-3
Total MIs (N)	300	292	-8
Incident stroke (N)	279	265	-14
Total strokes (N)	351	324	-27
Deaths (N)	1,564	1,555	-9
Chronic healthcare costs ($)	36,173,349	35,917,612	-255,737
Acute event costs ($)	5,414,657	5,138,097	-276,561
Medication costs ($)	23,362,673	35,725,383	12,362,710
Lp(a) test costs ($)	0	85,434	85,434
Total YLL	52,505	52,527	22
Total QALY	42,886	42,918	32
Total healthcare costs ($)	64,950,679	76,866,525	11,915,847
Total indirect costs ($)	14,739,219	14,689,809	-49,410
Total costs ($)	79,689,898	91,556,334	11,866,437
ICER ($ per QALY)			377,884
SICER ($ per QALY)			376,317

Table 8.15: Base case results, summary. Secondary prevention population, females. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	2,904	2,904	0
Lp(a) tests	0	2,904	2,904
Treatment modified in response to Lp(a) test	0	379	379
Total MIs (N)	684	660	-24
Total strokes (N)	1,252	1,214	-38
Deaths (N)	1,933	1,908	-25
Chronic healthcare costs ($)	186,232,356	187,456,266	1,223,909
Acute event costs ($)	15,629,287	15,241,612	-387,674
Medication costs ($)	0	1,652,440	1,652,440
Lp(a) test costs ($)	0	72,600	72,600
Total YLL	31,913	32,099	185
Total QALY	18,714	18,828	113
Total healthcare costs ($)	201,861,643	204,422,918	2,561,275
Total indirect costs ($)	330,162,370	328,618,316	-1,544,054
Total costs ($)	532,024,014	533,041,234	1,017,221
ICER ($ per QALY)			22,608
SICER ($ per QALY)			8,979

Table 8.16: Base case results, summary. Secondary prevention population, males. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	7,096	7,096	0
Lp(a) tests	0	7,096	7,096
Treatment modified in response to Lp(a) test	0	699	699
Total MIs (N)	2,591	2,439	-152
Total strokes (N)	1,724	1,696	-28
Deaths (N)	4,792	4,756	-36
Chronic healthcare costs ($)	499,135,193	500,386,814	1,251,621
Acute event costs ($)	33,435,269	32,245,588	-1,189,682
Medication costs ($)	0	3,047,640	3,047,640
Lp(a) test costs ($)	0	177,400	177,400
Total YLL	76,527	76,727	199
Total QALY	49,022	49,154	132
Total healthcare costs ($)	532,570,462	535,857,442	3,286,979
Total indirect costs ($)	583,966,793	581,772,508	-2,194,285
Total costs ($)	1,116,537,255	1,117,629,950	1,092,695
ICER ($ per QALY)			24,956
SICER ($ per QALY)			8,296

Table 8.17: Base case results, summary. Secondary prevention population, people aged 40-49 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	641	641	0
Lp(a) tests	0	641	641
Treatment modified in response to Lp(a) test	0	49	49
Total MIs (N)	265	251	-14
Total strokes (N)	270	272	2
Deaths (N)	487	486	-1
Chronic healthcare costs ($)	54,465,571	54,576,709	111,138
Acute event costs ($)	3,078,607	3,012,221	-66,386
Medication costs ($)	0	213,640	213,640
Lp(a) test costs ($)	0	16,025	16,025
Total YLL	9,134	9,152	18
Total QALY	5,749	5,761	11
Total healthcare costs ($)	57,544,178	57,818,595	274,417
Total indirect costs ($)	170,654,178	169,952,692	-701,485
Total costs ($)	228,198,356	227,771,287	-427,069
ICER ($ per QALY)			24,129
SICER ($ per QALY)			-37,552

Table 8.18: Base case results, summary. Secondary prevention population, people aged 50-59 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	2,446	2,446	0
Lp(a) tests	0	2,446	2,446
Treatment modified in response to Lp(a) test	0	251	251
Total MIs (N)	945	880	-65
Total strokes (N)	799	784	-15
Deaths (N)	1,787	1,764	-23
Chronic healthcare costs ($)	188,583,491	189,648,869	1,065,378
Acute event costs ($)	12,662,113	12,235,029	-427,084
Medication costs ($)	0	1,094,360	1,094,360
Lp(a) test costs ($)	0	61,150	61,150
Total YLL	30,053	30,219	165
Total QALY	18,802	18,905	104
Total healthcare costs ($)	201,245,604	203,039,407	1,793,804
Total indirect costs ($)	455,165,201	452,288,153	-2,877,048
Total costs ($)	656,410,805	655,327,561	-1,083,244
ICER ($ per QALY)			17,282
SICER ($ per QALY)			-10,437

Table 8.19: Base case results, summary. Secondary prevention population, people aged 60-69 at baseline. Study 2. Australia.

Outcome	Control	Intervention	Difference
Population size	6,913	6,913	0
Lp(a) tests	0	6,913	6,913
Treatment modified in response to Lp(a) test	0	778	778
Total MIs (N)	2,065	1,968	-97
Total strokes (N)	1,907	1,854	-53
Deaths (N)	4,451	4,414	-37
Chronic healthcare costs ($)	442,318,488	443,617,502	1,299,015
Acute event costs ($)	33,323,836	32,239,950	-1,083,886
Medication costs ($)	0	3,392,080	3,392,080
Lp(a) test costs ($)	0	172,825	172,825
Total YLL	69,254	69,455	201
Total QALY	43,186	43,316	130
Total healthcare costs ($)	475,642,323	479,422,357	3,780,034
Total indirect costs ($)	288,309,785	288,149,979	-159,806
Total costs ($)	763,952,108	767,572,336	3,620,228
ICER ($ per QALY)			29,115
SICER ($ per QALY)			27,884

Table 8.20: Base case results, summary. Primary prevention population, females. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,888	1,888
Incident MI (N)	690	659	-31
Total MIs (N)	844	793	-51
Incident stroke (N)	617	590	-27
Total strokes (N)	797	741	-56
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs (£)	76,542,513	76,006,043	-536,470
Acute event costs (£)	2,975,549	2,791,686	-183,863
Medication costs (£)	3,975,212	67,832,977	63,857,764
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,300	45
Total QALY	128,058	128,115	57
Total healthcare costs (£)	83,493,274	146,899,398	63,406,124
Total indirect costs (£)	57,084,779	56,975,521	-109,258
Total costs (£)	140,578,053	203,874,919	63,296,866
ICER (£ per QALY)			1,106,309
SICER (£ per QALY)			1,104,402

Table 8.21: Base case results, summary. Primary prevention population, males. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,240	1,240
Incident MI (N)	690	665	-25
Total MIs (N)	844	799	-45
Incident stroke (N)	617	600	-17
Total strokes (N)	797	755	-42
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs (£)	76,542,513	76,168,081	-374,432
Acute event costs (£)	2,975,549	2,825,543	-150,005
Medication costs (£)	3,975,212	46,488,760	42,513,548
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,305	51
Total QALY	128,058	128,113	56
Total healthcare costs (£)	83,493,274	125,751,078	42,257,804
Total indirect costs (£)	57,084,779	56,965,476	-119,303
Total costs (£)	140,578,053	182,716,554	42,138,501
ICER (£ per QALY)			756,844
SICER (£ per QALY)			754,707

Table 8.22: Base case results, summary. Primary prevention population, people aged 40-49 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	251	251
Incident MI (N)	690	684	-6
Total MIs (N)	844	834	-10
Incident stroke (N)	617	600	-17
Total strokes (N)	797	765	-32
Deaths (N)	3,693	3,683	-10
Chronic healthcare costs (£)	76,542,513	76,091,908	-450,605
Acute event costs (£)	2,975,549	2,887,510	-88,038
Medication costs (£)	3,975,212	12,714,409	8,739,197
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,280	26
Total QALY	128,058	128,089	31
Total healthcare costs (£)	83,493,274	91,962,521	8,469,247
Total indirect costs (£)	57,084,779	57,022,721	-62,058
Total costs (£)	140,578,053	148,985,242	8,407,189
ICER (£ per QALY)			272,127
SICER (£ per QALY)			270,133

Table 8.23: Base case results, summary. Primary prevention population, people aged 50-59 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs (£)	137,933,493	137,273,026	-660,468
Acute event costs (£)	5,139,048	4,931,709	-207,339
Medication costs (£)	6,900,217	48,019,223	41,119,005
Lp(a) test costs (£)	0	380,760	380,760
Total YLL	251,750	251,860	110
Total QALY	192,977	193,079	103
Total healthcare costs (£)	149,972,758	190,604,717	40,631,959
Total indirect costs (£)	80,701,506	80,492,292	-209,214
Total costs (£)	230,674,265	271,097,010	40,422,745
ICER (£ per QALY)			396,326
SICER (£ per QALY)			394,285

Table 8.24: Base case results, summary. Primary prevention population, people aged 60-69 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	4,384	4,384	0
Lp(a) tests	0	4,238	4,238
Treatment modified in response to Lp(a) test	0	337	337
Incident MI (N)	263	260	-3
Total MIs (N)	300	292	-8
Incident stroke (N)	279	265	-14
Total strokes (N)	351	324	-27
Deaths (N)	1,564	1,555	-9
Chronic healthcare costs (£)	30,681,818	30,240,345	-441,473
Acute event costs (£)	1,461,219	1,372,860	-88,359
Medication costs (£)	2,018,870	15,081,456	13,062,586
Lp(a) test costs (£)	0	145,259	145,259
Total YLL	58,841	58,868	27
Total QALY	44,375	44,409	35
Total healthcare costs (£)	34,161,906	46,839,919	12,678,013
Total indirect costs (£)	5,255,002	5,247,250	-7,751
Total costs (£)	39,416,908	52,087,170	12,670,262
ICER (£ per QALY)			365,118
SICER (£ per QALY)			364,895

Table 8.25: Base case results, summary. Secondary prevention population, females. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	2,904	2,904	0
Lp(a) tests	0	2,904	2,904
Treatment modified in response to Lp(a) test	0	379	379
Total MIs (N)	684	660	-24
Total strokes (N)	1,252	1,214	-38
Deaths (N)	1,933	1,908	-25
Chronic healthcare costs (£)	226,784,006	228,113,078	1,329,072
Acute event costs (£)	4,521,248	4,402,822	-118,426
Medication costs (£)	0	1,506,525	1,506,525
Lp(a) test costs (£)	0	116,160	116,160
Total YLL	35,857	36,081	223
Total QALY	19,687	19,814	127
Total healthcare costs (£)	231,305,254	234,138,585	2,833,331
Total indirect costs (£)	149,128,869	148,343,677	-785,192
Total costs (£)	380,434,123	382,482,262	2,048,139
ICER (£ per QALY)			22,293
SICER (£ per QALY)			16,115

Table 8.26: Base case results, summary. Secondary prevention population, males. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	7,096	7,096	0
Lp(a) tests	0	7,096	7,096
Treatment modified in response to Lp(a) test	0	699	699
Total MIs (N)	2,591	2,439	-152
Total strokes (N)	1,724	1,696	-28
Deaths (N)	4,792	4,756	-36
Chronic healthcare costs (£)	440,632,838	441,259,768	626,930
Acute event costs (£)	8,389,893	8,132,450	-257,443
Medication costs (£)	0	2,778,525	2,778,525
Lp(a) test costs (£)	0	283,840	283,840
Total YLL	85,735	85,976	240
Total QALY	50,933	51,078	145
Total healthcare costs (£)	449,022,731	452,454,583	3,431,852
Total indirect costs (£)	266,967,251	266,053,779	-913,473
Total costs (£)	715,989,982	718,508,361	2,518,379
ICER (£ per QALY)			23,727
SICER (£ per QALY)			17,411

Table 8.27: Base case results, summary. Secondary prevention population, people aged 40-49 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	641	641	0
Lp(a) tests	0	641	641
Treatment modified in response to Lp(a) test	0	49	49
Total MIs (N)	265	251	-14
Total strokes (N)	270	272	2
Deaths (N)	487	486	-1
Chronic healthcare costs (£)	64,511,053	64,602,752	91,699
Acute event costs (£)	892,690	881,013	-11,677
Medication costs (£)	0	194,775	194,775
Lp(a) test costs (£)	0	25,640	25,640
Total YLL	10,794	10,813	20
Total QALY	6,444	6,456	12
Total healthcare costs (£)	65,403,743	65,704,180	300,437
Total indirect costs (£)	93,952,423	93,655,309	-297,114
Total costs (£)	159,356,166	159,359,489	3,323
ICER (£ per QALY)			25,441
SICER (£ per QALY)			281

Table 8.28: Base case results, summary. Secondary prevention population, people aged 50-59 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	2,446	2,446	0
Lp(a) tests	0	2,446	2,446
Treatment modified in response to Lp(a) test	0	251	251
Total MIs (N)	945	880	-65
Total strokes (N)	799	784	-15
Deaths (N)	1,787	1,764	-23
Chronic healthcare costs (£)	196,072,227	197,146,791	1,074,564
Acute event costs (£)	3,346,401	3,250,287	-96,113
Medication costs (£)	0	997,725	997,725
Lp(a) test costs (£)	0	97,840	97,840
Total YLL	34,327	34,532	205
Total QALY	20,149	20,268	119
Total healthcare costs (£)	199,418,627	201,492,644	2,074,016
Total indirect costs (£)	217,094,832	215,682,209	-1,412,623
Total costs (£)	416,513,459	417,174,852	661,393
ICER (£ per QALY)			17,419
SICER (£ per QALY)			5,555

Table 8.29: Base case results, summary. Secondary prevention population, people aged 60-69 at baseline. Study 2. UK.

Outcome	Control	Intervention	Difference
Population size	6,913	6,913	0
Lp(a) tests	0	6,913	6,913
Treatment modified in response to Lp(a) test	0	778	778
Total MIs (N)	2,065	1,968	-97
Total strokes (N)	1,907	1,854	-53
Deaths (N)	4,451	4,414	-37
Chronic healthcare costs (£)	406,833,564	407,623,302	789,738
Acute event costs (£)	8,672,050	8,403,972	-268,078
Medication costs (£)	0	3,092,550	3,092,550
Lp(a) test costs (£)	0	276,520	276,520
Total YLL	76,472	76,711	239
Total QALY	44,026	44,167	141
Total healthcare costs (£)	415,505,614	419,396,344	3,890,730
Total indirect costs (£)	105,048,866	105,059,938	11,072
Total costs (£)	520,554,480	524,456,282	3,901,802
ICER (£ per QALY)			27,621
SICER (£ per QALY)			27,699

8.3 Threshold analyses

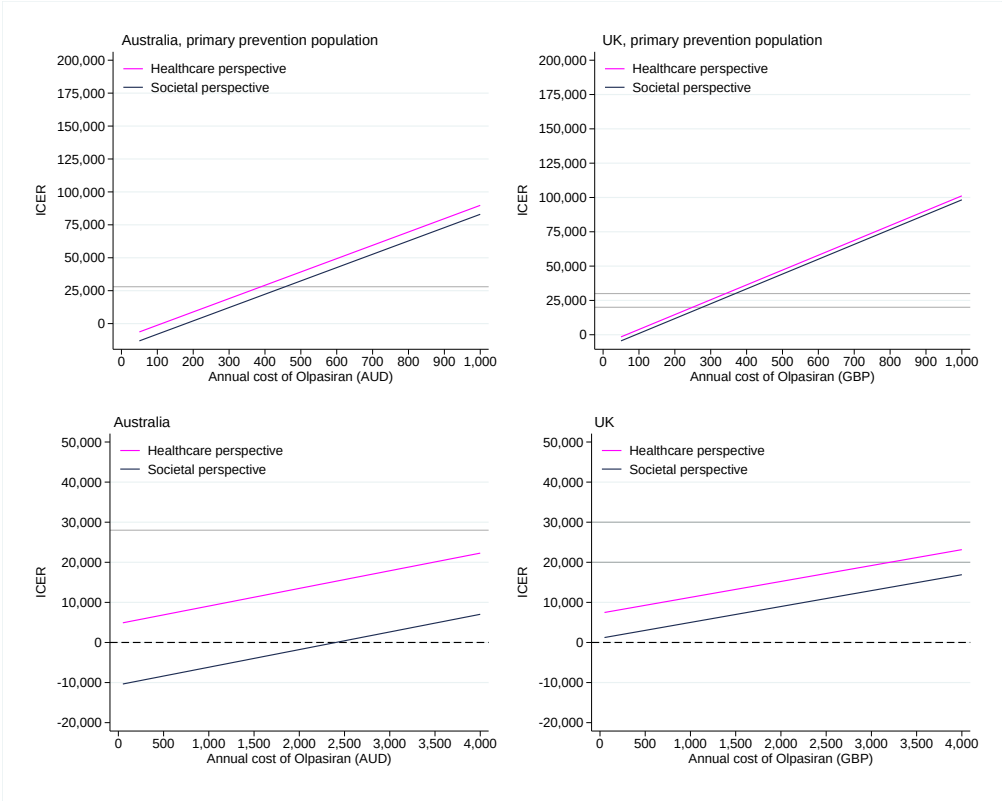
Given that Olpasiran is not cost-effective at the estimated price, it is worth preforming a threshold analysis to work out the price at which it would be cost-effective in both the primary and secondary prevention populations.

```
*mkdir thresh
quietly {
forval c = 1/2 {
forval t = 50(50)4000 {
forval ii = 0/1 {
use threshtemp_`ii´_`c´, clear
if `c´ == 1 {
replace DRUGSHC = DRUGSHC-4360 if LPT == 1
}
if `c´ == 2 {
replace DRUGSHC = DRUGSHC-3975 if LPT == 1
}
replace DRUGSHC = DRUGSHC+`t´ if LPT == 1
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
keep QLY HCC TSC
expand 3
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save thresh/tabres2_`ii´_c`t´_`c´, replace
}
use thresh/tabres2_0_c`t´_`c´, clear
merge 1:1 _n using thresh/tabres2_1_c`t´_`c´, nogen
gen double diff = val1-val0
gen ICER = diff[2]/diff[1]
gen SICER = diff[3]/diff[1]
keep ICER SICER
gen DC = `t´
gen pop = 1
gen country = `c´
keep if _n == 1
save thresh/PP_`t´_`c´, replace
forval ii = 0/1 {
use threshtempsp_`ii´_`c´, clear
if `c´ == 1 {
replace DRUGSHC = DRUGSHC-4360 if LPT == 1
}
if `c´ == 2 {
replace DRUGSHC = DRUGSHC-3975 if LPT == 1
}
replace DRUGSHC = DRUGSHC+`t´ if LPT == 1
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
```

```
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
keep QLY HCC TSC
expand 3
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save thresh/tabres2sp_`ii´_c`t´_`c´, replace
}
use thresh/tabres2sp_0_c`t´_`c´, clear
merge 1:1 _n using thresh/tabres2sp_1_c`t´_`c´, nogen
gen double diff = val1-val0
gen ICER = diff[2]/diff[1]
gen SICER = diff[3]/diff[1]
keep ICER SICER
gen DC = `t´
gen pop = 2
gen country = `c´
keep if _n == 1
save thresh/SP_`t´_`c´, replace
}
}
}
forval c = 1/2 {
if `c´ == 1 {
local cc = "Australia"
local cur = "AUD"
local y = 28000
}
if `c´ == 2 {
local cc = "UK"
local cur = "GBP"
local y = "20000 30000"
}
clear
forval t = 50(50)4000 {
append using thresh/PP_`t´_`c´
}
drop if DC > 1000
twoway ///
(line ICER DC, color(magenta)) ///
(line SICER DC, color(dknavy)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(1 "Healthcare perspective" ///
2 "Societal perspective") ///
cols(1)) yscale(nolog) xscale(nolog) ///
graphregion(color(white)) ///
ylabel(0(25000)200000, format(%9.0fc) angle(0)) ///
xlabel(0(100)1000, nogrid format(%9.0fc)) yline(`y´, lcol(black%30)) ///
ytitle("ICER") title("`cc´, primary prevention population", size(medium) placement(west) col(black))
>  ///
xtitle("Annual cost of Olpasiran (`cur´)")
graph save "Graph" GPH/treshPP_`c´, replace
clear
forval t = 50(50)4000 {
append using thresh/SP_`t´_`c´
```

```
}
twoway ///
(line ICER DC, color(magenta)) ///
(line SICER DC, color(dknavy)) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(1 "Healthcare perspective" ///
2 "Societal perspective") ///
cols(1)) yscale(nolog) xscale(nolog) ///
graphregion(color(white)) ///
ylabel(-20000(10000)50000, format(%9.0fc) angle(0)) ///
xlabel(0(500)4000, nogrid format(%9.0fc)) yline(0, lcol(black) lpattern(dash)) yline(`y´, lcol(black
> %30)) ///
ytitle("ICER") title("`cc´", size(medium) placement(west) col(black)) ///
xtitle("Annual cost of Olpasiran (`cur´)")
graph save "Graph" GPH/treshSP_`c´, replace
}
```

Figure 8.2: Threshold analysis results. Grey lines show the willingness-to-pay thresholds for each country.




```
graph combine ///
GPH/treshPP_1.gph ///
GPH/treshPP_2.gph ///
GPH/treshSP_1.gph ///
GPH/treshSP_2.gph ///
, graphregion(color(white)) altshrink cols(2) xsize(5)
> ry.)
```

9 Scenario analyses

9.1 Study 1

We will conduct the following scenario analyses for study 1:

1. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL).
2. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL) .
3. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL).
4. Assuming LDL-C lowering via statins causally increases the risk of diabetes.
5. Vary the LDL-C threshold to initiate treatment from 5.0 mmol/L to 3.0 mmol/L.
6. Vary the SBP threshold to initiate treatment from 160 mmHg to 140 mmHg.
7. Using a moderate intensity statin, rather than a high intensity statin. In Australia, the most common moderate intensity statin was Rosuvastatin 10 mg once per day (see below), which leads to an LDL-C reduction of 45.6% [159] and costs $194 per year (see below) in Australia [118] and £13.44 per year in the UK [122].
8. 0% discounting.
9. Changing the cost of the blood-pressure lowering therapy to that of a triple combination pill. In Australia, this lowers the annual cost from $555 to $340 (see below). Given the already low medication costs in the UK, this scenario analysis was not performed for the UK.

Note that scenario 4 is not applicable in the secondary prevention population – all individuals with a prior event should have been on statin therapy during the estimation of transition probabilities for this population, meaning the effects of statins are already factored in.

```
quietly {
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"NYSTATIN")
drop if pos !=0
drop pos
gen pos = strpos(drug_name,"STATIN")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
collapse (sum) prescriptions, by(drug)
sort prescriptions
keep if _n == _N
noisily list drug
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
```

```
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
keep if drug == "ROSUVASTATIN Tablet 10 mg (as calcium)"
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Triple combination " 365.25*r(mean)/30
}
quietly {
import delimited "pbs-item-drug-map.csv", varnames(1) clear
gen pos = strpos(drug_name,"VALSARTAN")
keep if pos !=0
save PBSpos, replace
import delimited "dos-jul-2019-to-sep-2023.csv", clear
merge m:1 item_code using PBSpos
keep if _merge == 3
drop _merge
keep if month == 202306
gen drug = drug_name + " " + formstrength
keep if drug == "AMLODIPINE + VALSARTAN + HYDROCHLOROTHIAZIDE Tablet 10 mg (as besilate)-320 mg-25 m
> g"
gen unitcost = total_cost/prescriptions
su unitcost [aweight=prescriptions]
noisily di "Triple combination " 365.25*r(mean)/30
}


. quietly {

     +-----------------------------------------+
     |                                    drug |
     |-----------------------------------------|
  1. | ROSUVASTATIN Tablet 10 mg (as calcium)  |
     +-----------------------------------------+

Rosuvastatin 10 mg 193.99697
. quietly {
Rosuvastatin 10 mg 339.59957


quietly {
forval s = 1/9 {
if `s´ == 1 {
local lpa = 50
}
else if `s´ == 2 {
local lpa = 70
}
else if `s´ == 3 {
local lpa = 110
}
else {
local lpa = 90
}
if `s´ <= 7 {
forval m = 0/1 {
use modstart, clear
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
```

```
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
if `m´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
if `s´ == 5 {
replace LLT = 1 if cycle == `c´ & ldl >= 3 & DT!=1 & MI == 0 & ST == 0
}
else {
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 6 {
replace AHT = 1 if cycle == `c´ & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
}
else {
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 7 {
bysort ind (age) : replace ldl = ldl*(1-0.456) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
else {
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
if `c´ == 0 {
gen LPAT=.
}
}
if `m´ == 1 {
if `c´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70)
gen LPAT=1 if vhr!=1 & cycle == `c´
replace vhr = 1 if lpa >= `lpa´ & cycle == `c´
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70 | lpa >= `lpa´) & MI == 0 & ST == 0
}
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
if `s´ == 5 {
replace LLT = 1 if cycle == `c´ & ldl >= 3 & DT!=1 & MI == 0 & ST == 0
}
else {
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 6 {
replace AHT = 1 if cycle == `c´ & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
}
else {
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 7 {
```

```
bysort ind (age) : replace ldl = ldl*(1-0.456) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
else {
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
if `s´ == 5 {
replace LLT = 1 if cycle == `c´ & ldl >= 3 & DT!=1 & MI == 0 & ST == 0
}
else {
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 6 {
replace AHT = 1 if cycle == `c´ & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
}
else {
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
}
if `s´ == 7 {
bysort ind (age) : replace ldl = ldl*(1-0.456) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
else {
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
}
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
```

```
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
merge m:1 sex age using DMmod, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge errr
rename (rate) (rate_`i´)
}
replace rate_t2d = 0 if DM == 1
recast double rate_t2d-rate_nfICH
if `s´ == 4 {
replace rate_t2d = rate_t2d*(0.763^(mcldl-aveldl))
}
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*(2.083^(mcldl-aveldl))
replace rate_`i´MI=rate_`i´MI*(1.0054^(mclpa-avelpa))
replace rate_`i´MI=rate_`i´MI*(1.058^(mcsbp-avesbp))
replace rate_`i´MI=rate_`i´MI*(1.43^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(0.26*DMP)) if DM == 0
replace rate_`i´MI=1.26*rate_`i´MI/(1+(0.26*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*(1.08^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*(1.0035^(mclpa-avelpa))
replace rate_`i´IS=rate_`i´IS*(1.027^(mcsbp-avesbp))
replace rate_`i´IS=rate_`i´IS*(1.33^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(0.74*DMP)) if DM == 0
replace rate_`i´IS=1.74*rate_`i´IS/(1+(0.74*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*(1.039^(mcsbp-avesbp))
}
replace rate_blacdeath = rate_blacdeath*(2.52^(lsi-LSI))
replace rate_colcdeath = rate_colcdeath*(1.24^(lsi-LSI))
replace rate_oescdeath = rate_oescdeath*(3.67^(lsi-LSI))
replace rate_kidcdeath = rate_kidcdeath*(1.69^(lsi-LSI))
replace rate_luncdeath = rate_luncdeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*(37.6) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*(1.27^(lsi-LSI))
replace rate_pancdeath = rate_pancdeath*(2.13^(lsi-LSI))
replace rate_pneudeath = rate_pneudeath*(1.016^(mcsbp-avesbp))
replace rate_pneudeath = rate_pneudeath*(4.03^(lsi-LSI))
replace rate_copddeath = rate_copddeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*(37.6) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*(1.09^(mcldl-aveldl))
replace rate_pddeath = rate_pddeath*(0.48^(lsi-LSI))
recode rate_ovacdeath .=0
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
```

```
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT
}
save modend`m´_`s´, replace
}
}
forval c = 1/2 {
forval i = 0/1 {
if `s´ <= 7 {
use modend`i´_`s´, clear
}
else {
use modend`i´, clear
}
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= `lpa´
keep if cycle!=.

if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}

sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
```

```
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.
if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
if `s´ == 7 {
replace DRUGSHC = DRUGSHC+194 if LLT == 1 & MI == 0 & ST == 0
}
else {
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
```

```
}
if `s´ == 9 {
replace DRUGSHC = DRUGSHC+340 if AHT == 1 & MI == 0 & ST == 0
}
else {
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
}
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
if `s´ == 7 {
replace DRUGSHC = DRUGSHC+13.44 if LLT == 1 & MI == 0 & ST == 0
}
else {
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
}
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}
gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
```

```
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
if `s´ == 8 {
gen DC = 1
}
else {
gen DC = 1/((1.05)^(cycle))
}
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
if `s´ == 8 {
gen DC = 1
}
else {
gen DC = 1/((1.035)^(cycle))
}
```

```
}
gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabres_`i´_`s´_`c´, replace
}
use tabres_0_`s´_`c´, clear
merge 1:1 _n using tabres_1_`s´_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
```

```
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/SCErestabsum_`s´_`c´.csv, delimiter(":") novarnames replace
}
}
}
```

9.1.1 Results

Table 9.1: Scenario analysis results, summary. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,749	1,749
Incident MI (N)	659	585	-74
Total MIs (N)	817	715	-102
Incident stroke (N)	566	564	-2
Total strokes (N)	728	727	-1
Deaths (N)	3,670	3,642	-28
Chronic healthcare costs ($)	86,896,417	83,909,545	-2,986,872
Acute event costs ($)	10,274,131	9,463,757	-810,373
Medication costs ($)	43,645,563	49,302,438	5,656,875
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,526	193
Total QALY	117,965	118,215	251
Total healthcare costs ($)	140,816,110	142,900,565	2,084,455
Total indirect costs ($)	114,253,596	108,913,785	-5,339,811
Total costs ($)	255,069,707	251,814,351	-3,255,356
ICER ($ per QALY)			8,316
SICER ($ per QALY)			-12,987

Table 9.2: Scenario analysis results, summary. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,146	1,146
Incident MI (N)	659	600	-59
Total MIs (N)	817	737	-80
Incident stroke (N)	566	565	-1
Total strokes (N)	728	730	2
Deaths (N)	3,670	3,650	-20
Chronic healthcare costs ($)	86,896,417	84,254,388	-2,642,029
Acute event costs ($)	10,274,131	9,647,976	-626,154
Medication costs ($)	43,645,563	47,385,391	3,739,829
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,436	103
Total QALY	117,965	118,130	166
Total healthcare costs ($)	140,816,110	141,512,580	696,470
Total indirect costs ($)	114,253,596	110,674,253	-3,579,343
Total costs ($)	255,069,707	252,186,833	-2,882,873
ICER ($ per QALY)			4,205
SICER ($ per QALY)			-17,407

Table 9.3: Scenario analysis results, summary. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	231	231
Incident MI (N)	659	649	-10
Total MIs (N)	817	805	-12
Incident stroke (N)	566	563	-3
Total strokes (N)	728	724	-4
Deaths (N)	3,670	3,668	-2
Chronic healthcare costs ($)	86,896,417	86,553,472	-342,945
Acute event costs ($)	10,274,131	10,181,117	-93,014
Medication costs ($)	43,645,563	44,290,247	644,684
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,363	30
Total QALY	117,965	118,002	37
Total healthcare costs ($)	140,816,110	141,249,660	433,550
Total indirect costs ($)	114,253,596	113,752,750	-500,847
Total costs ($)	255,069,707	255,002,410	-67,297
ICER ($ per QALY)			11,608
SICER ($ per QALY)			-1,802

Table 9.4: Scenario analysis results, summary. Scenario 4: LDL-C lowering with statins causes diabetes. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	690	658	-32
Total MIs (N)	833	791	-42
Incident stroke (N)	557	559	2
Total strokes (N)	708	705	-3
Deaths (N)	3,651	3,639	-12
Chronic healthcare costs ($)	91,903,921	91,822,753	-81,168
Acute event costs ($)	10,454,073	10,162,462	-291,612
Medication costs ($)	43,649,452	45,568,819	1,919,366
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,456	141,540	84
Total QALY	117,855	117,910	54
Total healthcare costs ($)	146,007,447	147,778,858	1,771,411
Total indirect costs ($)	113,957,379	112,642,383	-1,314,996
Total costs ($)	259,964,825	260,421,241	456,416
ICER ($ per QALY)			32,602
SICER ($ per QALY)			8,400

Table 9.5: Scenario analysis results, summary. Scenario 5: LDL-C lowering occurs at 3.0 mmol/L, not 5.0 mmol/L. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	477	472	-5
Total MIs (N)	579	573	-6
Incident stroke (N)	554	549	-5
Total strokes (N)	722	715	-7
Deaths (N)	3,612	3,609	-3
Chronic healthcare costs ($)	80,510,653	80,321,322	-189,331
Acute event costs ($)	8,402,687	8,317,037	-85,650
Medication costs ($)	55,572,785	56,609,167	1,036,382
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,694	141,719	25
Total QALY	118,456	118,486	31
Total healthcare costs ($)	144,486,125	145,472,352	986,226
Total indirect costs ($)	105,212,113	104,624,004	-588,109
Total costs ($)	249,698,239	250,096,356	398,117
ICER ($ per QALY)			32,223
SICER ($ per QALY)			13,008

Table 9.6: Scenario analysis results, summary. Scenario 6: SBP lowering occurs at 140 mmHg, not 160 mmHg. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	585	570	-15
Total MIs (N)	723	704	-19
Incident stroke (N)	542	534	-8
Total strokes (N)	697	687	-10
Deaths (N)	3,648	3,636	-12
Chronic healthcare costs ($)	83,905,960	83,388,093	-517,867
Acute event costs ($)	9,290,809	9,112,618	-178,191
Medication costs ($)	58,262,058	59,270,324	1,008,266
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,558	141,610	52
Total QALY	118,251	118,318	67
Total healthcare costs ($)	151,458,827	151,995,861	537,034
Total indirect costs ($)	108,458,631	107,152,218	-1,306,413
Total costs ($)	259,917,458	259,148,079	-769,379
ICER ($ per QALY)			7,989
SICER ($ per QALY)			-11,445

Table 9.7: Scenario analysis results, summary. Scenario 7: Moderate intensity statin. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	731	703	-28
Total MIs (N)	898	864	-34
Incident stroke (N)	554	550	-4
Total strokes (N)	707	706	-1
Deaths (N)	3,687	3,677	-10
Chronic healthcare costs ($)	87,705,963	86,813,096	-892,867
Acute event costs ($)	10,569,722	10,322,908	-246,814
Medication costs ($)	43,165,244	45,048,164	1,882,920
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,320	141,390	70
Total QALY	117,938	118,029	91
Total healthcare costs ($)	141,440,930	142,408,994	968,064
Total indirect costs ($)	114,401,620	112,852,540	-1,549,081
Total costs ($)	255,842,550	255,261,533	-581,017
ICER ($ per QALY)			10,634
SICER ($ per QALY)			-6,383

Table 9.8: Scenario analysis results, summary. Scenario 8: 0% discounting. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs ($)	191,893,379	189,599,330	-2,294,049
Acute event costs ($)	20,874,303	20,414,260	-460,043
Medication costs ($)	92,483,758	95,308,285	2,824,528
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	251,939	252,098	158
Total QALY	206,933	207,140	207
Total healthcare costs ($)	305,251,440	305,546,700	295,260
Total indirect costs ($)	187,251,908	184,593,557	-2,658,351
Total costs ($)	492,503,347	490,140,257	-2,363,090
ICER ($ per QALY)			1,425
SICER ($ per QALY)			-11,401

Table 9.9: Scenario analysis results, summary. Scenario 9: Combination price for anti-hypertensive therapy. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs ($)	86,896,417	85,947,239	-949,178
Acute event costs ($)	10,274,131	10,024,083	-250,048
Medication costs ($)	31,692,960	33,225,832	1,532,872
Lp(a) test costs ($)	0	224,825	224,825
Total YLL	141,333	141,392	60
Total QALY	117,965	118,047	82
Total healthcare costs ($)	128,863,508	129,421,979	558,471
Total indirect costs ($)	114,253,596	112,721,465	-1,532,131
Total costs ($)	243,117,105	242,143,444	-973,660
ICER ($ per QALY)			6,797
SICER ($ per QALY)			-11,849

Table 9.10: Scenario analysis results, summary. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,749	1,749
Incident MI (N)	659	585	-74
Total MIs (N)	817	715	-102
Incident stroke (N)	566	564	-2
Total strokes (N)	728	727	-1
Deaths (N)	3,670	3,642	-28
Chronic healthcare costs (£)	74,895,806	72,916,796	-1,979,010
Acute event costs (£)	2,738,237	2,595,630	-142,607
Medication costs (£)	4,004,436	4,503,425	498,989
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,597	249
Total QALY	128,184	128,480	296
Total healthcare costs (£)	81,638,479	80,375,571	-1,262,909
Total indirect costs (£)	55,683,736	53,041,134	-2,642,602
Total costs (£)	137,322,215	133,416,704	-3,905,511
ICER (£ per QALY)			-4,268
SICER (£ per QALY)			-13,198

Table 9.11: Scenario analysis results, summary. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	1,146	1,146
Incident MI (N)	659	600	-59
Total MIs (N)	817	737	-80
Incident stroke (N)	566	565	-1
Total strokes (N)	728	730	2
Deaths (N)	3,670	3,650	-20
Chronic healthcare costs (£)	74,895,806	73,082,233	-1,813,573
Acute event costs (£)	2,738,237	2,633,366	-104,871
Medication costs (£)	4,004,436	4,333,221	328,785
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,484	136
Total QALY	128,184	128,380	197
Total healthcare costs (£)	81,638,479	80,408,540	-1,229,940
Total indirect costs (£)	55,683,736	53,942,937	-1,740,798
Total costs (£)	137,322,215	134,351,477	-2,970,738
ICER (£ per QALY)			-6,257
SICER (£ per QALY)			-15,112

Table 9.12: Scenario analysis results, summary. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	231	231
Incident MI (N)	659	649	-10
Total MIs (N)	817	805	-12
Incident stroke (N)	566	563	-3
Total strokes (N)	728	724	-4
Deaths (N)	3,670	3,668	-2
Chronic healthcare costs (£)	74,895,806	74,569,510	-326,296
Acute event costs (£)	2,738,237	2,717,351	-20,887
Medication costs (£)	4,004,436	4,061,558	57,122
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,349	164,388	39
Total QALY	128,184	128,228	45
Total healthcare costs (£)	81,638,479	81,708,139	69,659
Total indirect costs (£)	55,683,736	55,437,412	-246,324
Total costs (£)	137,322,215	137,145,550	-176,665
ICER (£ per QALY)			1,564
SICER (£ per QALY)			-3,967

Table 9.13: Scenario analysis results, summary. Scenario 4: LDL-C lowering with statins causes diabetes. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	690	658	-32
Total MIs (N)	833	791	-42
Incident stroke (N)	557	559	2
Total strokes (N)	708	705	-3
Deaths (N)	3,651	3,639	-12
Chronic healthcare costs (£)	78,549,242	78,750,865	201,624
Acute event costs (£)	2,749,070	2,696,169	-52,901
Medication costs (£)	4,006,680	4,175,050	168,370
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,527	164,634	106
Total QALY	128,063	128,126	63
Total healthcare costs (£)	85,304,992	85,981,804	676,812
Total indirect costs (£)	55,608,484	55,023,043	-585,442
Total costs (£)	140,913,477	141,004,847	91,371
ICER (£ per QALY)			10,800
SICER (£ per QALY)			1,458

Table 9.14: Scenario analysis results, summary. Scenario 5: LDL-C lowering occurs at 3.0 mmol/L, not 5.0 mmol/L. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	477	472	-5
Total MIs (N)	579	573	-6
Incident stroke (N)	554	549	-5
Total strokes (N)	722	715	-7
Deaths (N)	3,612	3,609	-3
Chronic healthcare costs (£)	70,679,725	70,400,900	-278,825
Acute event costs (£)	2,403,502	2,377,442	-26,059
Medication costs (£)	5,182,488	5,264,311	81,823
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,828	164,862	33
Total QALY	128,771	128,809	37
Total healthcare costs (£)	78,265,714	78,402,373	136,659
Total indirect costs (£)	51,331,163	51,021,252	-309,911
Total costs (£)	129,596,877	129,423,625	-173,253
ICER (£ per QALY)			3,650
SICER (£ per QALY)			-4,628

Table 9.15: Scenario analysis results, summary. Scenario 6: SBP lowering occurs at 140 mmHg, not 160 mmHg. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	585	570	-15
Total MIs (N)	723	704	-19
Incident stroke (N)	542	534	-8
Total strokes (N)	697	687	-10
Deaths (N)	3,648	3,636	-12
Chronic healthcare costs (£)	72,750,268	72,078,916	-671,352
Acute event costs (£)	2,525,941	2,480,167	-45,774
Medication costs (£)	5,126,040	5,225,065	99,024
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,639	164,709	71
Total QALY	128,522	128,604	82
Total healthcare costs (£)	80,402,249	80,143,868	-258,381
Total indirect costs (£)	52,888,478	52,234,372	-654,106
Total costs (£)	133,290,728	132,378,240	-912,487
ICER (£ per QALY)			-3,157
SICER (£ per QALY)			-11,149

Table 9.16: Scenario analysis results, summary. Scenario 7: Moderate intensity statin. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	731	703	-28
Total MIs (N)	898	864	-34
Incident stroke (N)	554	550	-4
Total strokes (N)	707	706	-1
Deaths (N)	3,687	3,677	-10
Chronic healthcare costs (£)	75,147,175	74,292,955	-854,220
Acute event costs (£)	2,769,769	2,721,455	-48,314
Medication costs (£)	3,637,223	3,781,511	144,288
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	164,323	164,416	93
Total QALY	128,148	128,256	109
Total healthcare costs (£)	81,554,166	81,155,641	-398,526
Total indirect costs (£)	55,754,027	55,009,656	-744,371
Total costs (£)	137,308,194	136,165,297	-1,142,897
ICER (£ per QALY)			-3,672
SICER (£ per QALY)			-10,529

Table 9.17: Scenario analysis results, summary. Scenario 8: 0% discounting. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	8,993	8,993
Treatment modified in response to Lp(a) test	0	624	624
Incident MI (N)	659	632	-27
Total MIs (N)	817	782	-35
Incident stroke (N)	566	562	-4
Total strokes (N)	728	725	-3
Deaths (N)	3,670	3,659	-11
Chronic healthcare costs (£)	135,096,606	133,524,412	-1,572,195
Acute event costs (£)	4,776,538	4,698,293	-78,245
Medication costs (£)	6,955,149	7,178,084	222,936
Lp(a) test costs (£)	0	359,720	359,720
Total YLL	251,939	252,098	158
Total QALY	193,228	193,419	190
Total healthcare costs (£)	146,828,293	145,760,509	-1,067,784
Total indirect costs (£)	78,868,386	77,791,024	-1,077,362
Total costs (£)	225,696,679	223,551,533	-2,145,146
ICER (£ per QALY)			-5,608
SICER (£ per QALY)			-11,266

9.2 Study 2

We will conduct the following scenario analyses for study 2:

1. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL).
2. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL).
3. Vary the Lp(a) threshold that leads to treatment from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL).
4. 0% discounting.

For all of these, we will present both the primary results and results of the threshold analysis.

```
quietly {
forval s = 1/4 {
if `s´ == 1 {
local lpa = 50
}
else if `s´ == 2 {
local lpa = 70
}
else if `s´ == 3 {
local lpa = 110
}
else {
local lpa = 90
}
if `s´ <= 3 {
forval m = 0/1 {
use modstart2, clear
gen LPAT=.
gen LPT=.
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
```

```
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-0.517) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
>  ST == 0
bysort ind (age) : replace sbp = sbp-20 if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST ==
> 0
}
replace vhr = 1 if MI == 1 | ST == 1
if `m´ == 1 {
replace LPAT=1 if vhr==1 & cycle == `c´
replace LPT=1 if lpa >= `lpa´ & LPAT==1 & cycle == `c´ & vhr == 1
bysort ind (age) : replace lpa = lpa*(1-0.975) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
merge m:1 sex age using DMmod, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
```

```
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge errr
rename (rate) (rate_`i´)
}
replace rate_t2d = 0 if DM == 1
recast double rate_t2d-rate_nfICH
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
sort ind age
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*(2.083^(mcldl-aveldl))
replace rate_`i´MI=rate_`i´MI*(1.0054^(mclpa-avelpa))
replace rate_`i´MI=rate_`i´MI*(1.058^(mcsbp-avesbp))
replace rate_`i´MI=rate_`i´MI*(1.43^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(0.26*DMP)) if DM == 0
replace rate_`i´MI=1.26*rate_`i´MI/(1+(0.26*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*(1.08^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*(1.0035^(mclpa-avelpa))
replace rate_`i´IS=rate_`i´IS*(1.027^(mcsbp-avesbp))
replace rate_`i´IS=rate_`i´IS*(1.33^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(0.74*DMP)) if DM == 0
replace rate_`i´IS=1.74*rate_`i´IS/(1+(0.74*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*(1.039^(mcsbp-avesbp))
replace rate`i´MI=rate`i´MI*(1.0054^(mclpa-avelpa))
replace rate`i´S=rate`i´S*(1.0035^(mclpa-avelpa))
}
replace rate_blacdeath = rate_blacdeath*(2.52^(lsi-LSI))
replace rate_colcdeath = rate_colcdeath*(1.24^(lsi-LSI))
replace rate_oescdeath = rate_oescdeath*(3.67^(lsi-LSI))
replace rate_kidcdeath = rate_kidcdeath*(1.69^(lsi-LSI))
replace rate_luncdeath = rate_luncdeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*(37.6) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*(1.27^(lsi-LSI))
replace rate_pancdeath = rate_pancdeath*(2.13^(lsi-LSI))
replace rate_pneudeath = rate_pneudeath*(1.016^(mcsbp-avesbp))
replace rate_pneudeath = rate_pneudeath*(4.03^(lsi-LSI))
replace rate_copddeath = rate_copddeath*(13.64^(lsi-LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*(37.6) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*(1.09^(mcldl-aveldl))
replace rate_pddeath = rate_pddeath*(0.48^(lsi-LSI))
recode rate_ovacdeath .=0
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
```

```
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace LPT = LPT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT LPT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save modend2_`m´_`s´, replace
}
forval m = 0/1 {
use modstartsp, clear
if `m´ == 0 {
gen LPAT=.
gen LPT=.
}
if `m´ == 1 {
gen LPAT=1 if cycle == 0
gen LPT=1 if lpa >= `lpa´ & cycle == 0
}
forval c = 0/44 {
if `m´ == 1 {
bysort ind (age) : replace lpa = lpa*(1-0.975) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
replace mclpa = . if cycle!=`c´
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
merge m:1 sex age using LSImod, keep(1 3) nogen
merge m:1 age sex using INC/t2d
drop _merge errr
rename (rate) (rate_t2d)
replace rate_t2d = 0 if DM == 1
recast double rate_t2d
merge m:1 sex MI ST age using pevtp, keep(1 3) nogen
replace rate_t2d = rate_t2d*(1.21^(lsi-LSI))
foreach i in nf f {
replace rate`i´MI=rate`i´MI*(1.0054^(mclpa-avelpa))
replace rate`i´S=rate`i´S*(1.0035^(mclpa-avelpa))
}
```

```
sort ind age
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
foreach var of varlist lpa lsi {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save modendsp`m´_`s´, replace
}
}
forval c = 1/2 {
forval ii = 0/1 {
if `s´ <= 3 {
use modend2_`ii´_`s´, clear
}
else {
use modend2_`ii´, clear
}
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
```

```
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
```

```
if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
```

```
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
if `s´ == 4 {
gen DC = 1
}
else {
gen DC = 1/((1.05)^(cycle))
}
}

if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
if `s´ == 4 {
gen DC = 1
}
else {
gen DC = 1/((1.035)^(cycle))
}
}

gen N = 1 if cycle == 0
save threshtemp_`ii´_`s´_`c´, replace
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
```

```
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save tabres2_`ii´_`s´_`c´, replace
}
use tabres2_0_`s´_`c´, clear
merge 1:1 _n using tabres2_1_`s´_`c´, nogen
drop if inrange(_n,9,24)
replace val0 = val0[9]+val0[10]+val0[11]+val0[12]+val0[13]+val0[14]+val0[15] if _n == 15
replace val0 = val0[16]+val0[17] if _n == 17
replace val1 = val1[9]+val1[10]+val1[11]+val1[12]+val1[13]+val1[14]+val1[15] if _n == 15
replace val1 = val1[16]+val1[17] if _n == 17
drop if inrange(_n,9,14) | _n == 16
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 18
replace stat = "SICER" if _n == 19
replace val0 = . if _n>17
replace val1 = . if _n>17
replace diff = diff[15]/diff[14] if _n==18
replace diff = diff[17]/diff[14] if _n==19
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "Chronic healthcare costs (\textsterling)" if _n == 9
replace row = "Acute event costs (\textsterling)" if _n == 10
replace row = "Medication costs (\textsterling)" if _n == 11
replace row = "Lp(a) test costs (\textsterling)" if _n == 12
replace row = "Total YLL" if _n == 13
replace row = "Total QALY" if _n == 14
replace row = "Total healthcare costs (\textsterling)" if _n == 15
replace row = "Total indirect costs (\textsterling)" if _n == 16
replace row = "Total costs (\textsterling)" if _n == 17
replace row = "ICER (\textsterling \ per QALY)" if _n == 18
replace row = "SICER (\textsterling \ per QALY)" if _n == 19
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/SCErestabsum2_`s´_`c´.csv, delimiter(":") novarnames replace
forval i = 0/1 {
if `s´ <= 3 {
use modendsp`i´_`s´, clear
}
else {
use modendsp`i´, clear
}
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
drop xb errr UTlb UTub
}
if `c´ == 2 {
```

```
merge m:1 sex age using UTvals_UK, keep(3) nogen
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*6222
gen double STOHSHC = STOHS*4388
gen double DMOHSHC = DMOHS*3588
gen double MISHSHC =MISHS*6222
gen double MIDHSHC = MIDHS*8870
gen double STDHSHC = STDHS*8870
gen double MSDHSHC =MSDHS*8870
gen double ACMIC = 14434 if MIE == 1 & DTE == .
replace ACMIC = 3363 if MIE == 1 & DTE == 1
gen double ACSTC = 15659 if STE == 1 & DTE ==.
```

```
replace ACSTC = 13154 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}

if `c´ == 2 {
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
```

```
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
if `s´ == 4 {
gen DC = 1
}
else {
gen DC = 1/((1.05)^(cycle))
}
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
if `s´ == 4 {
gen DC = 1
}
else {
gen DC = 1/((1.035)^(cycle))
}
}
gen N = 1 if cycle == 0
save threshtempsp_`i´_`s´_`c´, replace
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
```

```
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 38
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabressp_`i´_`s´_`c´, replace
}
use tabressp_0_`s´_`c´, clear
merge 1:1 _n using tabressp_1_`s´_`c´, nogen
drop if inrange(_n,7,22)
replace val0 = val0[7]+val0[8]+val0[9]+val0[10]+val0[11]+val0[12]+val0[13] if _n == 13
replace val0 = val0[14]+val0[15] if _n == 15
replace val1 = val1[7]+val1[8]+val1[9]+val1[10]+val1[11]+val1[12]+val1[13] if _n == 13
replace val1 = val1[14]+val1[15] if _n == 15
drop if inrange(_n,7,12) | _n == 14
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 17
replace stat = "SICER" if _n == 18
replace val0 = . if _n>15
replace val1 = . if _n>15
replace diff = diff[13]/diff[12] if _n==16
replace diff = diff[15]/diff[12] if _n==17
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Total MIs (N)" if _n == 4
replace row = "Total strokes (N)" if _n == 5
replace row = "Deaths (N)" if _n == 6
replace row = "Chronic healthcare costs (\textsterling)" if _n == 7
replace row = "Acute event costs (\textsterling)" if _n == 8
replace row = "Medication costs (\textsterling)" if _n == 9
replace row = "Lp(a) test costs (\textsterling)" if _n == 10
replace row = "Total YLL" if _n == 11
replace row = "Total QALY" if _n == 12
replace row = "Total healthcare costs (\textsterling)" if _n == 13
replace row = "Total indirect costs (\textsterling)" if _n == 14
replace row = "Total costs (\textsterling)" if _n == 15
replace row = "ICER (\textsterling \ per QALY)" if _n == 16
replace row = "SICER (\textsterling \ per QALY)" if _n == 17
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
order row val0 val1 diff
drop stat
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
export delimited using CSV/SCErestabspsum_`s´_`c´.csv, delimiter(":") novarnames replace
forval t = 50(50)4000 {
forval ii = 0/1 {
use threshtemp_`ii´_`s´_`c´, clear
if `c´ == 1 {
replace DRUGSHC = DRUGSHC-4360 if LPT == 1
}
if `c´ == 2 {
replace DRUGSHC = DRUGSHC-3975 if LPT == 1
```

```
}
replace DRUGSHC = DRUGSHC+`t´ if LPT == 1
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
keep QLY HCC TSC
expand 3
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save thresh/tabres2_`ii´_c`t´_`s´_`c´, replace
}
use thresh/tabres2_0_c`t´_`s´_`c´, clear
merge 1:1 _n using thresh/tabres2_1_c`t´_`s´_`c´, nogen
gen double diff = val1-val0
gen ICER = diff[2]/diff[1]
gen SICER = diff[3]/diff[1]
keep ICER SICER
gen DC = `t´
gen pop = 1
keep if _n == 1
gen sce = `s´
save thresh/PP_`t´_`s´_`c´, replace
forval ii = 0/1 {
use threshtempsp_`ii´_`s´_`c´, clear
if `c´ == 1 {
replace DRUGSHC = DRUGSHC-4360 if LPT == 1
}
if `c´ == 2 {
replace DRUGSHC = DRUGSHC-3975 if LPT == 1
}
replace DRUGSHC = DRUGSHC+`t´ if LPT == 1
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
keep QLY HCC TSC
expand 3
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
```

```
save thresh/tabres2sp_`ii´_c`t´_`s´_`c´, replace
}
use thresh/tabres2sp_0_c`t´_`s´_`c´, clear
merge 1:1 _n using thresh/tabres2sp_1_c`t´_`s´_`c´, nogen
gen double diff = val1-val0
gen ICER = diff[2]/diff[1]
gen SICER = diff[3]/diff[1]
keep ICER SICER
gen DC = `t´
gen pop = 2
keep if _n == 1
gen sce = `s´
gen country = `c´
save thresh/SP_`t´_`s´_`c´, replace
}
}
}
}
```

9.2.1 Results

Table 9.18: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,888	1,888
Incident MI (N)	690	659	-31
Total MIs (N)	844	793	-51
Incident stroke (N)	617	590	-27
Total strokes (N)	797	741	-56
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs ($)	87,578,365	86,865,345	-713,021
Acute event costs ($)	10,911,055	10,290,702	-620,353
Medication costs ($)	43,333,417	100,135,563	56,802,146
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,293	36
Total QALY	117,856	117,907	51
Total healthcare costs ($)	141,822,837	197,439,869	55,617,032
Total indirect costs ($)	116,754,059	116,550,952	-203,107
Total costs ($)	258,576,897	313,990,822	55,413,925
ICER ($ per QALY)			1,086,931
SICER ($ per QALY)			1,082,962

```
forval c = 1/2 {
if `c´ == 1 {
local cc = "Australia"
local cur = "AUD"
local y = 28000
}
if `c´ == 2 {
local cc = "UK"
local cur = "GBP"
local y = "20000 30000"
}
clear
forval s = 1/4 {
forval t = 50(50)4000 {
append using thresh/PP_`t´_`s´_`c´
}
}
drop if DC > 1000
colorpalette viridis, n(5) nograph
twoway ///
(line ICER DC if sce == 1, color("`r(p1)´")) ///
(line ICER DC if sce == 2, color("`r(p2)´")) ///
(line ICER DC if sce == 3, color("`r(p3)´")) ///
(line ICER DC if sce == 4, color("`r(p4)´")) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
```

Table 9.19: Base case results, summary. Secondary prevention population. Study 2. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	2,610	2,610
Total MIs (N)	3,275	2,960	-315
Total strokes (N)	2,976	2,779	-197
Deaths (N)	6,725	6,629	-96
Chronic healthcare costs ($)	685,367,549	688,882,436	3,514,887
Acute event costs ($)	49,064,556	45,524,538	-3,540,018
Medication costs ($)	0	11,379,600	11,379,600
Lp(a) test costs ($)	0	250,000	250,000
Total YLL	108,441	108,983	542
Total QALY	67,737	68,090	353
Total healthcare costs ($)	734,432,105	746,036,574	11,604,468
Total indirect costs ($)	914,129,163	910,889,766	-3,239,398
Total costs ($)	1,648,561,269	1,656,926,339	8,365,071
ICER ($ per QALY)			32,860
SICER ($ per QALY)			23,687

```
order(1 "1" ///
2 "2" ///
3 "3" ///
4 "4") ///
cols(1) subtitle(Scenario)) yscale(nolog) xscale(nolog) ///
graphregion(color(white)) ///
ylabel(0(50000)200000, format(%9.0fc) angle(0)) ///
xlabel(0(100)1000, nogrid format(%9.0fc)) yline(`y´, lcol(black%30)) ///
ytitle("ICER") title("`cc´, primary prevention population", size(medium) placement(west) col(black))
>  ///
xtitle("Annual cost of Olpasiran (`cur´)")
graph save "Graph" GPH/treshPP_`c´_sce, replace
clear
forval s = 1/4 {
forval t = 50(50)4000 {
append using thresh/SP_`t´_`s´_`c´
}
}
colorpalette viridis, n(5) nograph
twoway ///
(line ICER DC if sce == 1, color("`r(p1)´")) ///
(line ICER DC if sce == 2, color("`r(p2)´")) ///
(line ICER DC if sce == 3, color("`r(p3)´")) ///
(line ICER DC if sce == 4, color("`r(p4)´")) ///
, legend(symxsize(0.13cm) position(11) ring(0) region(lcolor(white) color(none)) ///
order(1 "1" ///
2 "2" ///
3 "3" ///
4 "4") ///
cols(1) subtitle(Scenario)) yscale(nolog) xscale(nolog) ///
graphregion(color(white)) ///
```

Table 9.20: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,240	1,240
Incident MI (N)	690	665	-25
Total MIs (N)	844	799	-45
Incident stroke (N)	617	600	-17
Total strokes (N)	797	755	-42
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs ($)	87,578,365	86,990,370	-587,996
Acute event costs ($)	10,911,055	10,388,414	-522,641
Medication costs ($)	43,333,417	81,257,717	37,924,300
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,298	40
Total QALY	117,856	117,905	50
Total healthcare costs ($)	141,822,837	178,784,760	36,961,923
Total indirect costs ($)	116,754,059	116,514,675	-239,384
Total costs ($)	258,576,897	295,299,435	36,722,539
ICER ($ per QALY)			744,864
SICER ($ per QALY)			740,040

```
ylabel(0(5000)35000, format(%9.0fc) angle(0)) ///
xlabel(0(500)4000, nogrid format(%9.0fc)) yline(0, lcol(black) lpattern(dash)) yline(`y´, lcol(black
> %30)) ///
ytitle("ICER") title("`cc´, secondary prevention population", size(medium) placement(west) col(black
> )) ///
xtitle("Annual cost of Olpasiran (`cur´)")
graph save "Graph" GPH/treshSP_`c´_sce, replace
}


graph combine ///
GPH/treshPP_1_sce.gph ///
GPH/treshPP_2_sce.gph ///
GPH/treshSP_1_sce.gph ///
GPH/treshSP_2_sce.gph ///
, graphregion(color(white)) altshrink cols(2) xsize(5)
> erspective.)
```

Table 9.21: Base case results, summary. Secondary prevention population. Study 2. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	7,096	7,096	0
Lp(a) tests	0	7,096	7,096
Treatment modified in response to Lp(a) test	0	699	699
Total MIs (N)	2,591	2,439	-152
Total strokes (N)	1,724	1,696	-28
Deaths (N)	4,792	4,756	-36
Chronic healthcare costs ($)	499,135,193	500,386,814	1,251,621
Acute event costs ($)	33,435,269	32,245,588	-1,189,682
Medication costs ($)	0	3,047,640	3,047,640
Lp(a) test costs ($)	0	177,400	177,400
Total YLL	76,527	76,727	199
Total QALY	49,022	49,154	132
Total healthcare costs ($)	532,570,462	535,857,442	3,286,979
Total indirect costs ($)	583,966,793	581,772,508	-2,194,285
Total costs ($)	1,116,537,255	1,117,629,950	1,092,695
ICER ($ per QALY)			24,956
SICER ($ per QALY)			8,296

Table 9.22: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	251	251
Incident MI (N)	690	684	-6
Total MIs (N)	844	834	-10
Incident stroke (N)	617	600	-17
Total strokes (N)	797	765	-32
Deaths (N)	3,693	3,683	-10
Chronic healthcare costs ($)	87,578,365	87,350,375	-227,991
Acute event costs ($)	10,911,055	10,647,689	-263,366
Medication costs ($)	43,333,417	51,202,572	7,869,155
Lp(a) test costs ($)	0	148,259	148,259
Total YLL	141,258	141,277	19
Total QALY	117,856	117,882	27
Total healthcare costs ($)	141,822,837	149,348,895	7,526,058
Total indirect costs ($)	116,754,059	116,616,577	-137,482
Total costs ($)	258,576,897	265,965,473	7,388,576
ICER ($ per QALY)			280,503
SICER ($ per QALY)			275,378

Table 9.23: Base case results, summary. Secondary prevention population. Study 2. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). Australia.

Outcome	Control	Intervention	Difference
Population size	641	641	0
Lp(a) tests	0	641	641
Treatment modified in response to Lp(a) test	0	49	49
Total MIs (N)	265	251	-14
Total strokes (N)	270	272	2
Deaths (N)	487	486	-1
Chronic healthcare costs ($)	54,465,571	54,576,709	111,138
Acute event costs ($)	3,078,607	3,012,221	-66,386
Medication costs ($)	0	213,640	213,640
Lp(a) test costs ($)	0	16,025	16,025
Total YLL	9,134	9,152	18
Total QALY	5,749	5,761	11
Total healthcare costs ($)	57,544,178	57,818,595	274,417
Total indirect costs ($)	170,654,178	169,952,692	-701,485
Total costs ($)	228,198,356	227,771,287	-427,069
ICER ($ per QALY)			24,129
SICER ($ per QALY)			-37,552

Table 9.24: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 4: LDL-C lowering with statins causes diabetes. Australia.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs ($)	194,258,303	193,637,820	-620,483
Acute event costs ($)	22,267,643	21,467,868	-799,775
Medication costs ($)	91,737,770	137,004,338	45,266,568
Lp(a) test costs ($)	0	237,975	237,975
Total YLL	251,750	251,860	110
Total QALY	206,652	206,766	113
Total healthcare costs ($)	308,263,716	352,348,000	44,084,284
Total indirect costs ($)	191,005,719	190,560,137	-445,582
Total costs ($)	499,269,435	542,908,138	43,638,702
ICER ($ per QALY)			389,458
SICER ($ per QALY)			385,521

Table 9.25: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 5: 0% discounting. Australia.

Outcome	Control	Intervention	Difference
Population size	4,384	4,384	0
Lp(a) tests	0	4,238	4,238
Treatment modified in response to Lp(a) test	0	337	337
Incident MI (N)	263	260	-3
Total MIs (N)	300	292	-8
Incident stroke (N)	279	265	-14
Total strokes (N)	351	324	-27
Deaths (N)	1,564	1,555	-9
Chronic healthcare costs ($)	36,173,349	35,917,612	-255,737
Acute event costs ($)	5,414,657	5,138,097	-276,561
Medication costs ($)	23,362,673	35,725,383	12,362,710
Lp(a) test costs ($)	0	85,434	85,434
Total YLL	52,505	52,527	22
Total QALY	42,886	42,918	32
Total healthcare costs ($)	64,950,679	76,866,525	11,915,847
Total indirect costs ($)	14,739,219	14,689,809	-49,410
Total costs ($)	79,689,898	91,556,334	11,866,437
ICER ($ per QALY)			377,884
SICER ($ per QALY)			376,317

Table 9.26: Base case results, summary. Secondary prevention population. Study 2. Scenario 5: 0% discounting. Australia.

Outcome	Control	Intervention	Difference
Population size	6,913	6,913	0
Lp(a) tests	0	6,913	6,913
Treatment modified in response to Lp(a) test	0	778	778
Total MIs (N)	2,065	1,968	-97
Total strokes (N)	1,907	1,854	-53
Deaths (N)	4,451	4,414	-37
Chronic healthcare costs ($)	442,318,488	443,617,502	1,299,015
Acute event costs ($)	33,323,836	32,239,950	-1,083,886
Medication costs ($)	0	3,392,080	3,392,080
Lp(a) test costs ($)	0	172,825	172,825
Total YLL	69,254	69,455	201
Total QALY	43,186	43,316	130
Total healthcare costs ($)	475,642,323	479,422,357	3,780,034
Total indirect costs ($)	288,309,785	288,149,979	-159,806
Total costs ($)	763,952,108	767,572,336	3,620,228
ICER ($ per QALY)			29,115
SICER ($ per QALY)			27,884

Table 9.27: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,888	1,888
Incident MI (N)	690	659	-31
Total MIs (N)	844	793	-51
Incident stroke (N)	617	590	-27
Total strokes (N)	797	741	-56
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs (£)	76,542,513	76,006,043	-536,470
Acute event costs (£)	2,975,549	2,791,686	-183,863
Medication costs (£)	3,975,212	67,832,977	63,857,764
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,300	45
Total QALY	128,058	128,115	57
Total healthcare costs (£)	83,493,274	146,899,398	63,406,124
Total indirect costs (£)	57,084,779	56,975,521	-109,258
Total costs (£)	140,578,053	203,874,919	63,296,866
ICER (£ per QALY)			1,106,309
SICER (£ per QALY)			1,104,402

Table 9.28: Base case results, summary. Secondary prevention population. Study 2. Scenario 1: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 105 nmol/L (50 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	10,000	10,000
Treatment modified in response to Lp(a) test	0	2,610	2,610
Total MIs (N)	3,275	2,960	-315
Total strokes (N)	2,976	2,779	-197
Deaths (N)	6,725	6,629	-96
Chronic healthcare costs (£)	667,416,844	669,099,887	1,683,043
Acute event costs (£)	12,911,141	11,996,015	-915,126
Medication costs (£)	0	10,374,750	10,374,750
Lp(a) test costs (£)	0	400,000	400,000
Total YLL	121,593	122,239	647
Total QALY	70,620	71,005	385
Total healthcare costs (£)	680,327,985	691,870,652	11,542,667
Total indirect costs (£)	416,096,120	414,702,187	-1,393,933
Total costs (£)	1,096,424,105	1,106,572,839	10,148,734
ICER (£ per QALY)			29,958
SICER (£ per QALY)			26,340

Table 9.29: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	1,240	1,240
Incident MI (N)	690	665	-25
Total MIs (N)	844	799	-45
Incident stroke (N)	617	600	-17
Total strokes (N)	797	755	-42
Deaths (N)	3,693	3,675	-18
Chronic healthcare costs (£)	76,542,513	76,168,081	-374,432
Acute event costs (£)	2,975,549	2,825,543	-150,005
Medication costs (£)	3,975,212	46,488,760	42,513,548
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,305	51
Total QALY	128,058	128,113	56
Total healthcare costs (£)	83,493,274	125,751,078	42,257,804
Total indirect costs (£)	57,084,779	56,965,476	-119,303
Total costs (£)	140,578,053	182,716,554	42,138,501
ICER (£ per QALY)			756,844
SICER (£ per QALY)			754,707

Table 9.30: Base case results, summary. Secondary prevention population. Study 2. Scenario 2: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 149 nmol/L (70 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	7,096	7,096	0
Lp(a) tests	0	7,096	7,096
Treatment modified in response to Lp(a) test	0	699	699
Total MIs (N)	2,591	2,439	-152
Total strokes (N)	1,724	1,696	-28
Deaths (N)	4,792	4,756	-36
Chronic healthcare costs (£)	440,632,838	441,259,768	626,930
Acute event costs (£)	8,389,893	8,132,450	-257,443
Medication costs (£)	0	2,778,525	2,778,525
Lp(a) test costs (£)	0	283,840	283,840
Total YLL	85,735	85,976	240
Total QALY	50,933	51,078	145
Total healthcare costs (£)	449,022,731	452,454,583	3,431,852
Total indirect costs (£)	266,967,251	266,053,779	-913,473
Total costs (£)	715,989,982	718,508,361	2,518,379
ICER (£ per QALY)			23,727
SICER (£ per QALY)			17,411

Table 9.31: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	251	251
Incident MI (N)	690	684	-6
Total MIs (N)	844	834	-10
Incident stroke (N)	617	600	-17
Total strokes (N)	797	765	-32
Deaths (N)	3,693	3,683	-10
Chronic healthcare costs (£)	76,542,513	76,091,908	-450,605
Acute event costs (£)	2,975,549	2,887,510	-88,038
Medication costs (£)	3,975,212	12,714,409	8,739,197
Lp(a) test costs (£)	0	268,693	268,693
Total YLL	164,255	164,280	26
Total QALY	128,058	128,089	31
Total healthcare costs (£)	83,493,274	91,962,521	8,469,247
Total indirect costs (£)	57,084,779	57,022,721	-62,058
Total costs (£)	140,578,053	148,985,242	8,407,189
ICER (£ per QALY)			272,127
SICER (£ per QALY)			270,133

Table 9.32: Base case results, summary. Secondary prevention population. Study 2. Scenario 3: Lp(a) threshold for treatment varied from 192 nmol/L (90 mg/dL) to 236 nmol/L (110 mg/dL). UK.

Outcome	Control	Intervention	Difference
Population size	641	641	0
Lp(a) tests	0	641	641
Treatment modified in response to Lp(a) test	0	49	49
Total MIs (N)	265	251	-14
Total strokes (N)	270	272	2
Deaths (N)	487	486	-1
Chronic healthcare costs (£)	64,511,053	64,602,752	91,699
Acute event costs (£)	892,690	881,013	-11,677
Medication costs (£)	0	194,775	194,775
Lp(a) test costs (£)	0	25,640	25,640
Total YLL	10,794	10,813	20
Total QALY	6,444	6,456	12
Total healthcare costs (£)	65,403,743	65,704,180	300,437
Total indirect costs (£)	93,952,423	93,655,309	-297,114
Total costs (£)	159,356,166	159,359,489	3,323
ICER (£ per QALY)			25,441
SICER (£ per QALY)			281

Table 9.33: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 4: LDL-C lowering with statins causes diabetes. UK.

Outcome	Control	Intervention	Difference
Population size	10,000	10,000	0
Lp(a) tests	0	9,519	9,519
Treatment modified in response to Lp(a) test	0	678	678
Incident MI (N)	690	679	-11
Total MIs (N)	844	823	-21
Incident stroke (N)	617	598	-19
Total strokes (N)	797	757	-40
Deaths (N)	3,693	3,671	-22
Chronic healthcare costs (£)	137,933,493	137,273,026	-660,468
Acute event costs (£)	5,139,048	4,931,709	-207,339
Medication costs (£)	6,900,217	48,019,223	41,119,005
Lp(a) test costs (£)	0	380,760	380,760
Total YLL	251,750	251,860	110
Total QALY	192,977	193,079	103
Total healthcare costs (£)	149,972,758	190,604,717	40,631,959
Total indirect costs (£)	80,701,506	80,492,292	-209,214
Total costs (£)	230,674,265	271,097,010	40,422,745
ICER (£ per QALY)			396,326
SICER (£ per QALY)			394,285

Table 9.34: Base case results, summary. Secondary prevention population. Study 2. Scenario 4: LDL-C lowering with statins causes diabetes. UK.

Outcome	Control	Intervention	Difference
Population size	2,446	2,446	0
Lp(a) tests	0	2,446	2,446
Treatment modified in response to Lp(a) test	0	251	251
Total MIs (N)	945	880	-65
Total strokes (N)	799	784	-15
Deaths (N)	1,787	1,764	-23
Chronic healthcare costs (£)	196,072,227	197,146,791	1,074,564
Acute event costs (£)	3,346,401	3,250,287	-96,113
Medication costs (£)	0	997,725	997,725
Lp(a) test costs (£)	0	97,840	97,840
Total YLL	34,327	34,532	205
Total QALY	20,149	20,268	119
Total healthcare costs (£)	199,418,627	201,492,644	2,074,016
Total indirect costs (£)	217,094,832	215,682,209	-1,412,623
Total costs (£)	416,513,459	417,174,852	661,393
ICER (£ per QALY)			17,419
SICER (£ per QALY)			5,555

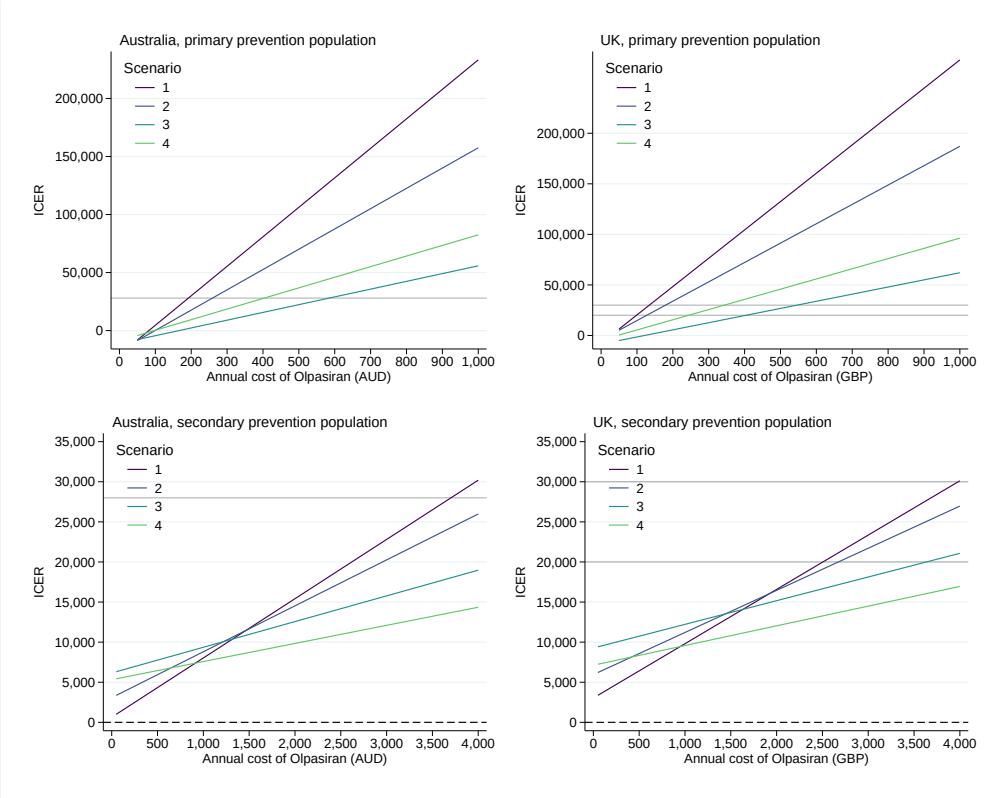
Table 9.35: Scenario analysis results, summary. Primary prevention population. Study 2. Scenario 5: 0% discounting. UK.

Outcome	Control	Intervention	Difference
Population size	4,384	4,384	0
Lp(a) tests	0	4,238	4,238
Treatment modified in response to Lp(a) test	0	337	337
Incident MI (N)	263	260	-3
Total MIs (N)	300	292	-8
Incident stroke (N)	279	265	-14
Total strokes (N)	351	324	-27
Deaths (N)	1,564	1,555	-9
Chronic healthcare costs (£)	30,681,818	30,240,345	-441,473
Acute event costs (£)	1,461,219	1,372,860	-88,359
Medication costs (£)	2,018,870	15,081,456	13,062,586
Lp(a) test costs (£)	0	145,259	145,259
Total YLL	58,841	58,868	27
Total QALY	44,375	44,409	35
Total healthcare costs (£)	34,161,906	46,839,919	12,678,013
Total indirect costs (£)	5,255,002	5,247,250	-7,751
Total costs (£)	39,416,908	52,087,170	12,670,262
ICER (£per QALY)			365,118
SICER (£per QALY)			364,895

Table 9.36: Base case results, summary. Secondary prevention population. Study 2. Scenario 5: 0% discounting. UK.

Outcome	Control	Intervention	Difference
Population size	6,913	6,913	0
Lp(a) tests	0	6,913	6,913
Treatment modified in response to Lp(a) test	0	778	778
Total MIs (N)	2,065	1,968	-97
Total strokes (N)	1,907	1,854	-53
Deaths (N)	4,451	4,414	-37
Chronic healthcare costs (£)	406,833,564	407,623,302	789,738
Acute event costs (£)	8,672,050	8,403,972	-268,078
Medication costs (£)	0	3,092,550	3,092,550
Lp(a) test costs (£)	0	276,520	276,520
Total YLL	76,472	76,711	239
Total QALY	44,026	44,167	141
Total healthcare costs (£)	415,505,614	419,396,344	3,890,730
Total indirect costs (£)	105,048,866	105,059,938	11,072
Total costs (£)	520,554,480	524,456,282	3,901,802
ICER (£per QALY)			27,621
SICER (£per QALY)			27,699

Figure 9.1: Threshold analysis results for scenario analyses in Study 2. ICERs are from the healthcare perspective.



10 One-way sensitivity analyses

10.1 Study 1

10.1.1 Code

```
*mkdir OSA
quietly {
foreach osa in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 ///
501 502 503 504 505 506 507 508 509 510 ///
601 602 603 604 605 606 607 608 609 ///
701 702 703 704 705 706 {
noisily di `osa´
forval rf = 1/2 {
foreach o in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 ///
133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 ///
501 502 503 504 505 506 507 508 ///
509 510 ///
601 602 603 604 605 606 607 608 ///
609 ///
701 702 703 704 705 706 {
local o`o´ = 0
}
if `rf´ == 1 {
local o`osa´ = -1.96
}
if `rf´ == 2 {
local o`osa´ = 1.96
}
local ll = 0.517
local sl = 20
if `osa´ == 401 & `rf´ == 1 {
local ll = 0.512
}
if `osa´ == 401 & `rf´ == 2 {
local ll = 0.522
}
if `osa´ == 402 & `rf´ == 1 {
local sl = 15
}
if `osa´ == 402 & `rf´ == 2 {
local sl = 25
}
local o502 = 0.785
local o503 = 0.79
local o504 = 0.65
local o505 = 0.65
local o506 = -0.055
```

```
local o507 = -0.164
local o508 = -0.164
local o509 = -0.01125
local o510 = -0.03
if `osa´ == 502 & `rf´ == 1 {
local o502 = 0.681
}
if `osa´ == 502 & `rf´ == 2 {
local o502 = 0.889
}
if `osa´ == 503 & `rf´ == 1 {
local o503 = 0.73
}
if `osa´ == 503 & `rf´ == 2 {
local o503 = 0.85
}
if `osa´ == 504 & `rf´ == 1 {
local o504 = 0.63
}
if `osa´ == 504 & `rf´ == 2 {
local o504 = 0.67
}
if `osa´ == 505 & `rf´ == 1 {
local o505 = 0.63
}
if `osa´ == 505 & `rf´ == 2 {
local o505 = 0.67
}
if `osa´ == 506 & `rf´ == 1 {
local o506 = -0.067
}
if `osa´ == 506 & `rf´ == 2 {
local o506 = -0.042
}
if `osa´ == 507 & `rf´ == 1 {
local o507 = -0.222
}
if `osa´ == 507 & `rf´ == 2 {
local o507 = -0.105
}
if `osa´ == 508 & `rf´ == 1 {
local o508 = -0.222
}
if `osa´ == 508 & `rf´ == 2 {
local o508 = -0.105
}
if `osa´ == 509 & `rf´ == 1 {
local o509 = -0.0135
}
if `osa´ == 509 & `rf´ == 2 {
local o509 = -0.009
}
if `osa´ == 510 & `rf´ == 1 {
local o510 = -0.036
}
if `osa´ == 510 & `rf´ == 2 {
local o510 = -0.024
}
local o601 = 3588
local o602 = 6222
local o603 = 4388
local o604 = 6222
local o605 = 8870
local o606 = 8870
local o607 = 8870
local o6081 = 14434
local o6091 = 15659
local o6010 = 2546
```

```
local o6011 = 2170
local o6020 = 3304
local o6021 = 2917
local o6030 = 7021
local o6031 = 7351
local o6040 = 14442
local o6041 = 12616
local o6050 = 4511
local o6051 = 3917
local o6060 = 10014
local o6061 = 10494
local o6070 = 14442
local o6071 = 12616
local o6082 = 2212
local o6092 = 4626
if `osa´ == 601 & `rf´ == 1 {
local o601 = 2816
local o6010 = 2462
local o6011 = 2090
}
if `osa´ == 601 & `rf´ == 2 {
local o601 = 4539
local o6010 = 2633
local o6011 = 2254
}
if `osa´ == 602 & `rf´ == 1 {
local o602 = 6222*0.9
local o6020 = 3026
local o6021 = 2701
}
if `osa´ == 602 & `rf´ == 2 {
local o602 = 6222*1.1
local o6020 = 3607
local o6021 = 3149
}
if `osa´ == 603 & `rf´ == 1 {
local o603 = 4388*0.9
local o6030 = 5852
local o6031 = 5923
}
if `osa´ == 603 & `rf´ == 2 {
local o603 = 4388*1.1
local o6030 = 8421
local o6031 = 8062
}
if `osa´ == 604 & `rf´ == 1 {
local o604 = 6222*0.9
local o6040 = 9929
local o6041 = 8484
}
if `osa´ == 604 & `rf´ == 2 {
local o604 = 6222*1.1
local o6040 = 21004
local o6041 = 18756
}
if `osa´ == 605 & `rf´ == 1 {
local o605 = 6804
local o6050 = 3947
local o6051 = 3480
}
if `osa´ == 605 & `rf´ == 2 {
local o605 = 10937
local o6050 = 5155
local o6051 = 4409
}
if `osa´ == 606 & `rf´ == 1 {
local o606 = 6804
local o6060 = 7615
```

```
local o6061 = 7701
}
if `osa´ == 606 & `rf´ == 2 {
local o606 = 10937
local o6060 = 13167
local o6061 = 14300
}
if `osa´ == 607 & `rf´ == 1 {
local o607 = 6804
local o6070 = 9929
local o6071 = 8484
}
if `osa´ == 607 & `rf´ == 2 {
local o607 = 10937
local o6070 = 21004
local o6071 = 18756
}
if `osa´ == 608 & `rf´ == 1 {
local o6081 = 14434*0.9
local o6082 = 2212*0.9
}
if `osa´ == 608 & `rf´ == 2 {
local o6081 = 14434*1.1
local o6082 = 2212*1.1
}
if `osa´ == 609 & `rf´ == 1 {
local o6091 = 15659*0.9
local o6092 = 4626*0.9
}
if `osa´ == 609 & `rf´ == 2 {
local o6091 = 15659*1.1
local o6092 = 4626*1.1
}
local o701 = 55
local o702 = 90
local o703 = 21.5
local o704 = 6
local o705 = 1.46
local o706 = 1.92
if `osa´ == 701 & `rf´ == 1 {
local o701 = 44
}
if `osa´ == 701 & `rf´ == 2 {
local o701 = 66
}
if `osa´ == 702 & `rf´ == 1 {
local o702 = 72
}
if `osa´ == 702 & `rf´ == 2 {
local o702 = 108
}
if `osa´ == 703 & `rf´ == 1 {
local o703 = 17.2
}
if `osa´ == 703 & `rf´ == 2 {
local o703 = 25.8
}
if `osa´ == 704 & `rf´ == 1 {
local o704 = 2
}
if `osa´ == 704 & `rf´ == 2 {
local o704 = 10
}
if `osa´ == 705 & `rf´ == 1 {
local o705 = 1.36
}
if `osa´ == 705 & `rf´ == 2 {
local o705 = 1.55
```

```
}
if `osa´ == 706 & `rf´ == 1 {
local o706 = 1.80
}
if `osa´ == 706 & `rf´ == 2 {
local o706 = 2.06
}
if `osa´ <= 402 {
forval m = 0/1 {
use modstart, clear
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
if `m´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
if `c´ == 0 {
gen LPAT=.
}
}
if `m´ == 1 {
if `c´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70)
gen LPAT=1 if vhr!=1 & cycle == `c´
replace vhr = 1 if lpa >= 90 & cycle == `c´
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70 | lpa >= 90) & MI == 0 & ST == 0
}
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
```

```
> = 0
}
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace aveldl = aveldl+(`o201´*errr)
drop errr
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
gen errr = (avelpaub-avelpalb)/3.92
replace avelpa = avelpa+(`o202´*errr)
drop errr
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
gen errr = (avesbpub-avesbplb)/3.92
replace avesbp = avesbp+(`o203´*errr)
drop errr
merge m:1 sex age using DMmod, keep(1 3) nogen
replace DMP = exp(ln(DMP)+(`o204´*errr))
drop errr
merge m:1 sex age using LSImod, keep(1 3) nogen
replace LSI = exp(ln(LSI)+(`o205´*errr))
drop errr
```

```
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge
rename (rate errr) (rate_`i´ errr_`i´)
}
replace rate_nfMI = exp(ln(rate_nfMI)+(`o101´*errr_nfMI))
replace rate_fMI = exp(ln(rate_fMI)+(`o102´*errr_fMI))
replace rate_nfIS = exp(ln(rate_nfIS)+(`o103´*errr_nfIS))
replace rate_fIS = exp(ln(rate_fIS)+(`o104´*errr_fIS))
replace rate_nfICH = exp(ln(rate_nfICH)+(`o105´*errr_nfICH))
replace rate_fICH = exp(ln(rate_fICH)+(`o106´*errr_fICH))
replace rate_blacdeath = exp(ln(rate_blacdeath)+(`o107´*errr_blacdeath))
replace rate_colcdeath = exp(ln(rate_colcdeath)+(`o108´*errr_colcdeath))
replace rate_oescdeath = exp(ln(rate_oescdeath)+(`o109´*errr_oescdeath))
replace rate_kidcdeath = exp(ln(rate_kidcdeath)+(`o110´*errr_kidcdeath))
replace rate_luncdeath = exp(ln(rate_luncdeath)+(`o111´*errr_luncdeath))
replace rate_ovacdeath = exp(ln(rate_ovacdeath)+(`o112´*errr_ovacdeath))
replace rate_pancdeath = exp(ln(rate_pancdeath)+(`o113´*errr_pancdeath))
replace rate_pneudeath = exp(ln(rate_pneudeath)+(`o114´*errr_pneudeath))
replace rate_copddeath = exp(ln(rate_copddeath)+(`o115´*errr_copddeath))
replace rate_alsdeath = exp(ln(rate_alsdeath)+(`o116´*errr_alsdeath))
replace rate_pddeath = exp(ln(rate_pddeath)+(`o117´*errr_pddeath))
replace rate_otherdeath = exp(ln(rate_otherdeath)+(`o118´*errr_otherdeath))
replace rate_t2d = exp(ln(rate_t2d)+(`o134´*errr_t2d))
replace rate_t2d = 0 if DM == 1
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
drop  errr_`i´
}
recast double rate_t2d-rate_nfICH
replace rate_t2d = rate_t2d*((exp(ln(1.21)+(`o312´*((ln(1.03)-ln(1.41))/3.92))))^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*((exp(ln(2.083)+(`o301´*((ln(2.222)-ln(2.000))/3.92))))^(mcldl-aveldl)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.0054)+(`o302´*((ln(1.0045)-ln(1.0062))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.058)+(`o303´*((ln(1.051)-ln(1.064))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.43)+(`o304´*((ln(1.22)-ln(1.62))/3.92))))^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(((exp(ln(1.26)+(`o305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´MI=(exp(ln(1.26)+(`o305´*((ln(1.40)-ln(1.08))/3.92))))*rate_`i´MI/(1+(((exp(ln(1.26)
> +(`o305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.08)+(`o306´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.0035)+(`o307´*((ln(1.0023)-ln(1.0045))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.027)+(`o308´*((ln(1.018)-ln(1.037))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.33)+(`o309´*((ln(1.22)-ln(1.46))/3.92))))^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(((exp(ln(1.74)+(`o310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´IS=(exp(ln(1.74)+(`o310´*((ln(2.47)-ln(1.19))/3.92))))*rate_`i´IS/(1+(((exp(ln(1.74)
> +(`o310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*((exp(ln(1.039)+(`o311´*((ln(1.010)-ln(1.069))/3.92))))^(mcsbp-avesb
> p))
}
replace rate_blacdeath = rate_blacdeath*((exp(ln(2.52)+(`o313´*((ln(1.66)-ln(3.81))/3.92))))^(lsi-LS
> I))
replace rate_colcdeath = rate_colcdeath*((exp(ln(1.24)+(`o314´*((ln(1.06)-ln(1.44))/3.92))))^(lsi-LS
> I))
replace rate_oescdeath = rate_oescdeath*((exp(ln(3.67)+(`o315´*((ln(1.67)-ln(8.02))/3.92))))^(lsi-LS
> I))
replace rate_kidcdeath = rate_kidcdeath*((exp(ln(1.69)+(`o316´*((ln(1.04)-ln(3.05))/3.92))))^(lsi-LS
> I))
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`o317´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
```

```
> LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`o317´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*((exp(ln(1.27)+(`o318´*((ln(1.04)-ln(1.57))/3.92))))^(lsi-LS
> I))
replace rate_pancdeath = rate_pancdeath*((exp(ln(2.13)+(`o319´*((ln(1.15)-ln(3.90))/3.92))))^(lsi-LS
> I))
replace rate_pneudeath = rate_pneudeath*((exp(ln(1.016)+(`o320´*((ln(1.008)-ln(1.025))/3.92))))^(mcs
> bp-avesbp))
replace rate_pneudeath = rate_pneudeath*((exp(ln(4.03)+(`o321´*((ln(3.16)-ln(5.11))/3.92))))^(lsi-LS
> I))
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`o322´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
> LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`o322´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*((exp(ln(1.09)+(`o323´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-av
> eldl))
replace rate_pddeath = rate_pddeath*((exp(ln(0.48)+(`o324´*((ln(0.27)-ln(1.01))/3.92))))^(lsi-LSI))
recode rate_ovacdeath .=0
merge m:1 sex MI ST age using pevtpe, keep(1 3) nogen
replace ratenfMI = exp(ln(ratenfMI)+(`o119´*errrnfMI)) if MI == 1 & ST == 0
replace ratefMI = exp(ln(ratefMI)+(`o120´*errrfMI)) if MI == 1 & ST == 0
replace ratenfS = exp(ln(ratenfS)+(`o121´*errrnfS)) if MI == 1 & ST == 0
replace ratefS = exp(ln(ratefS)+(`o122´*errrfS)) if MI == 1 & ST == 0
replace rateothd = exp(ln(rateothd)+(`o123´*errrothd)) if MI == 1 & ST == 0
replace ratenfMI = exp(ln(ratenfMI)+(`o124´*errrnfMI)) if MI == 0 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o125´*errrfMI)) if MI == 0 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o126´*errrnfS)) if MI == 0 & ST == 1
replace ratefS = exp(ln(ratefS)+(`o127´*errrfS)) if MI == 0 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o128´*errrothd)) if MI == 0 & ST == 1
replace ratenfMI = exp(ln(ratenfMI)+(`o129´*errrnfMI)) if MI == 1 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o130´*errrfMI)) if MI == 1 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o131´*errrnfS)) if MI == 1 & ST == 1
replace ratefS = exp(ln(ratefS)+(`o132´*errrfS)) if MI == 1 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o133´*errrothd)) if MI == 1 & ST == 1
drop errrfMI-errrothd
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
```

```
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT
}
save OSA/modend`m´_`osa´_`rf´, replace
}
}

forval c = 1/2 {
forval i = 0/1 {
if `osa´ <= 402 {
use OSA/modend`i´_`osa´_`rf´, clear
}
else {
use OSA/modend`i´_0_1, clear
}
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= 90
keep if cycle!=.

if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
replace UT = invlogit(xb+`o501´*errr)
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
replace UT = UT+(UT*0.0255*`o501´)
replace UT = 0 if UT < 0
replace UT = 1 if UT > 1
}

sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
```

```
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*`o503´
gen double STOHSQL = STOHS*UT*`o504´
gen double DMOHSQL = DMOHS*UT*`o502´
gen double MISHSQL = MISHS*UT*`o505´
gen double MIDHSQL = MIDHS*UT*(`o502´+`o506´)
gen double STDHSQL = STDHS*UT*(`o502´+`o507´)
gen double MSDHSQL = MSDHS*UT*(`o502´+`o508´)
replace MIOHSQL = MIOHSQL+`o509´ if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o509´ if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL+`o509´ if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o509´ if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL+`o510´ if STE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o510´ if STE == 1 & DTE==.
replace STDHSQL = STDHSQL+`o510´ if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o510´ if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*`o602´
gen double STOHSHC = STOHS*`o603´
gen double DMOHSHC = DMOHS*`o601´
gen double MISHSHC = MISHS*`o604´
gen double MIDHSHC = MIDHS*`o605´
gen double STDHSHC = STDHS*`o606´
gen double MSDHSHC = MSDHS*`o607´
gen double ACMIC = `o6081´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6081´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6091´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6091´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*`o6020´ if sex == 0
gen double STOHSHC = STOHS*`o6030´ if sex == 0
gen double DMOHSHC = DMOHS*`o6010´ if sex == 0
gen double MISHSHC = MISHS*`o6040´ if sex == 0
gen double MIDHSHC = MIDHS*`o6050´ if sex == 0
gen double STDHSHC = STDHS*`o6060´ if sex == 0
```

```
gen double MSDHSHC = MSDHS*`o6070´ if sex == 0
replace MIOHSHC = MIOHS*`o6021´ if sex == 1
replace STOHSHC = STOHS*`o6031´ if sex == 1
replace DMOHSHC = DMOHS*`o6011´ if sex == 1
replace MISHSHC = MISHS*`o6041´ if sex == 1
replace MIDHSHC = MIDHS*`o6051´ if sex == 1
replace STDHSHC = STDHS*`o6061´ if sex == 1
replace MSDHSHC = MSDHS*`o6071´ if sex == 1
gen double ACMIC = `o6082´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6082´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6092´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6092´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(`o705´*(1-WFP_GP))
gen STOHS_WFP = 1-(`o706´*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(`o706´*(1-WFP_GP))
gen MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-`o703´)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-`o703´)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-`o704´)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-`o703´)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace STOHS_WFP = STOHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MIDHS_WFP = MIDHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace STDHS_WFP = STDHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace MSDHS_WFP = MSDHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MIOHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 &
```

```
>  DTE==.
replace STOHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace STDHS_WFP = STDHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save OSA/tabres_`i´_`osa´_`rf´_`c´, replace
}
}
/*
if `osa´ <= 402 {
forval m = 0/1 {
erase OSA/modend`i´_`osa´_`rf´.dta
}
}
*/
}
}
```

```
}
clear
foreach osa in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 ///
501 502 503 504 505 506 507 508 509 510 ///
601 602 603 604 605 606 607 608 609 ///
701 702 703 704 705 706 {
forval rf = 1/2 {
forval c = 1/2 {
use OSA/tabres_0_`osa´_`rf´_`c´, clear
merge 1:1 _n using OSA/tabres_1_`osa´_`rf´_`c´, nogen
gen double diff = val1-val0
expand 2 if stat=="TSC"
replace stat = "ICER" if _n == 41
replace diff = diff[38]/diff[37] if _n==41
keep if _n == 41
keep diff
gen osa = `osa´
gen rf = `rf´
gen country = `c´
save OSA/icer_`osa´_`rf´_`c´, replace
}
}
}
clear
foreach osa in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 ///
501 502 503 504 505 506 507 508 509 510 ///
601 602 603 604 605 606 607 608 609  {
forval rf = 1/2 {
forval c = 1/2 {
append using OSA/icer_`osa´_`rf´_`c´
}
}
}
bysort country osa (rf) : gen double iceru = diff[_n+1]
keep if rf == 1
rename diff icerl
label variable icerl "Lower limit"
label variable iceru "Upper limit"
gen AA = ""
replace AA = "TP: Incident non-fatal MI" if osa == 101
replace AA = "TP: Incident fatal MI" if osa == 102
replace AA = "TP: Incident non-fatal IS" if osa == 103
replace AA = "TP: Incident fatal IS" if osa == 104
replace AA = "TP: Incident non-fatal ICH" if osa == 105
replace AA = "TP: Incident fatal ICH" if osa == 106
replace AA = "TP: Bladder cancer death " if osa == 107
replace AA = "TP: Colorectal cancer death " if osa == 108
replace AA = "TP: Oesophageal cancer death " if osa == 109
replace AA = "TP: Kidney cancer death " if osa == 110
replace AA = "TP: Lung cancer death " if osa == 111
```

```
replace AA = "TP: Ovarian cancer death " if osa == 112
replace AA = "TP: Pancreatic cancer death " if osa == 113
replace AA = "TP: Pneumonia death " if osa == 114
replace AA = "TP: COPD death " if osa == 115
replace AA = "TP: ALS death " if osa == 116
replace AA = "TP: Parksinson´s disease death " if osa == 117
replace AA = "TP: Other death " if osa == 118
replace AA = "TP: Non-fatal MI following MI" if osa == 119
replace AA = "TP: Fatal MI following MI" if osa == 120
replace AA = "TP: Non-fatal stroke following MI" if osa == 121
replace AA = "TP: Fatal stroke following MI" if osa == 122
replace AA = "TP: Other death following MI" if osa == 123
replace AA = "TP: Non-fatal MI following stroke" if osa == 124
replace AA = "TP: Fatal MI following stroke" if osa == 125
replace AA = "TP: Non-fatal stroke following stroke" if osa == 126
replace AA = "TP: Fatal stroke following stroke" if osa == 127
replace AA = "TP: Other death following stroke" if osa == 128
replace AA = "TP: Non-fatal MI following MI and stroke" if osa == 129
replace AA = "TP: Fatal MI following MI and stroke" if osa == 130
replace AA = "TP: Non-fatal stroke following MI and stroke" if osa == 131
replace AA = "TP: Fatal stroke following MI and stroke" if osa == 132
replace AA = "TP: Other death following MI and stroke" if osa == 133
replace AA = "TP: Incidence of type 2 diabetes" if osa == 134
replace AA = "RF UKB: Mean cumulative LDL-C" if osa == 201
replace AA = "RF UKB: Mean cumulative Lp(a)" if osa == 202
replace AA = "RF UKB: Mean cumulative SBP" if osa == 203
replace AA = "RF UKB: Prevalence of type 2 diabetes" if osa == 204
replace AA = "RF UKB: Mean LSI" if osa == 205
replace AA = "RF OC: LDL-C on incident MI" if osa == 301
replace AA = "RF OC: Lp(a) on incident MI" if osa == 302
replace AA = "RF OC: SBP on incident MI" if osa == 303
replace AA = "RF OC: Smoking on incident MI" if osa == 304
replace AA = "RF OC: Type 2 diabetes on incident MI" if osa == 305
replace AA = "RF OC: LDL-C on incident IS" if osa == 306
replace AA = "RF OC: Lp(a) on incident IS" if osa == 307
replace AA = "RF OC: SBP on incident IS" if osa == 308
replace AA = "RF OC: Smoking on incident IS" if osa == 309
replace AA = "RF OC: Type 2 diabetes on incident IS" if osa == 310
replace AA = "RF OC: SBP on incident ICH" if osa == 311
replace AA = "RF OC: Smoking on incident type 2 diabetes" if osa == 312
replace AA = "RF OC: Smoking on death from bladder cancer" if osa == 313
replace AA = "RF OC: Smoking on death from colorectal cancer" if osa == 314
replace AA = "RF OC: Smoking on death from oesophageal cancer" if osa == 315
replace AA = "RF OC: Smoking on death from kidney cancer" if osa == 316
replace AA = "RF OC: Smoking on death from lung cancer" if osa == 317
replace AA = "RF OC: Smoking on ovarian cancer" if osa == 318
replace AA = "RF OC: Smoking on pancreatic cancer" if osa == 319
replace AA = "RF OC: SBP on death from Pneumonia" if osa == 320
replace AA = "RF OC: Smoking on death from Pneumonia" if osa == 321
replace AA = "RF OC: Smoking on death from COPD" if osa == 322
replace AA = "RF OC: LDL-C on death from ALS" if osa == 323
replace AA = "RF OC: Smoking on death from Parkinson´s disease" if osa == 324
replace AA = "Effect of LDL-C lowering intervention" if osa == 401
replace AA = "Effect of SBP lowering intervention" if osa == 402
replace AA = "Chronic utility for no CVD or diabetes" if osa == 501
replace AA = "Chronic utility for diabetes" if osa == 502
replace AA = "Chronic utility for MI" if osa == 503
replace AA = "Chronic utility for stroke" if osa == 504
replace AA = "Chronic utility for MI and stroke" if osa == 505
replace AA = "Chronic disutility for MI in diabetes" if osa == 506
replace AA = "Chronic disutility for stroke in diabetes" if osa == 507
replace AA = "Chronic disutility for MI and stroke in diabetes" if osa == 508
replace AA = "Acute disutility for MI" if osa == 509
replace AA = "Acute disutility for stroke" if osa == 510
replace AA = "Chronic cost of diabetes" if osa == 601
replace AA = "Chronic cost of MI" if osa == 602
replace AA = "Chronic cost of stroke" if osa == 603
replace AA = "Chronic cost of MI and stroke" if osa == 604
```

```
replace AA = "Chronic cost of MI in diabetes" if osa == 605
replace AA = "Chronic cost of stroke in diabetes" if osa == 606
replace AA = "Chronic cost of MI and stroke in diabetes" if osa == 607
replace AA = "Acute cost of MI" if osa == 608
replace AA = "Acute cost of stroke" if osa == 609
save OSA/osares, replace
forval c = 1/2 {
if "`c´" == "1" {
local cc = "Australia"
local xlab = "8000(2000)16000"
}
else {
local cc = "UK"
local xlab = "-8000(2000)0"
}
use OSA/osares, clear
keep if country == `c´
sort osa
gen ref = icerl[1]
local ref = ref[1]
drop if osa == 0
gen sz = ((iceru-ref)^2)+((icerl-ref)^2)
sort sz
gen njm = _n
forval ii = 1/84 {
local `ii´ = AA[`ii´]
}
twoway ///
(bar icerl njm, horizontal base(`ref´) color(magenta)) ///
(bar iceru njm, horizontal base(`ref´) color(dknavy)) ///
, ylabel( ///
1 "`1´" ///
2 "`2´" ///
3 "`3´" ///
4 "`4´" ///
5 "`5´" ///
6 "`6´" ///
7 "`7´" ///
8 "`8´" ///
9 "`9´" ///
10 "`10´" ///
11 "`11´" ///
12 "`12´" ///
13 "`13´" ///
14 "`14´" ///
15 "`15´" ///
16 "`16´" ///
17 "`17´" ///
18 "`18´" ///
19 "`19´" ///
20 "`20´" ///
21 "`21´" ///
22 "`22´" ///
23 "`23´" ///
24 "`24´" ///
25 "`25´" ///
26 "`26´" ///
27 "`27´" ///
28 "`28´" ///
29 "`29´" ///
30 "`30´" ///
31 "`31´" ///
32 "`32´" ///
33 "`33´" ///
34 "`34´" ///
35 "`35´" ///
36 "`36´" ///
37 "`37´" ///
```

```
38 "`38´" ///
39 "`39´" ///
40 "`40´" ///
41 "`41´" ///
42 "`42´" ///
43 "`43´" ///
44 "`44´" ///
45 "`45´" ///
46 "`46´" ///
47 "`47´" ///
48 "`48´" ///
49 "`49´" ///
50 "`50´" ///
51 "`51´" ///
52 "`52´" ///
53 "`53´" ///
54 "`54´" ///
55 "`55´" ///
56 "`56´" ///
57 "`57´" ///
58 "`58´" ///
59 "`59´" ///
60 "`60´" ///
61 "`61´" ///
62 "`62´" ///
63 "`63´" ///
64 "`64´" ///
65 "`65´" ///
66 "`66´" ///
67 "`67´" ///
68 "`68´" ///
69 "`69´" ///
70 "`70´" ///
71 "`71´" ///
72 "`72´" ///
73 "`73´" ///
74 "`74´" ///
75 "`75´" ///
76 "`76´" ///
77 "`77´" ///
78 "`78´" ///
79 "`79´" ///
80 "`80´" ///
81 "`81´" ///
82 "`82´" ///
83 "`83´" ///
84 "`84´" ///
, angle(0) nogrid) ysize(2) xsize(1.4) ///
legend(position(12) region(lcolor(white)) cols(2)) ///
ytitle("") graphregion(color(white) margin(30 10 2 2)) xlabel(`xlab´, format(%15.0fc)) ///
xtitle("ICER", margin(medium)) ///
title("`cc´", placement(west) size(medium) col(black))
graph save GPH/OSAT_`c´, replace
}
```

10.1.2 Results

```
. use OSA/osares, clear
. list icerl iceru AA if country == 1 & osa !=0, separator(0)
```

	icerl	iceru	AA
2.	12133.85	9212.4602	TP: Incident non-fatal MI
3.	11270.699	10921.937	TP: Incident fatal MI
4.	10423.047	11306.798	TP: Incident non-fatal IS
5.	11268.071	11341.305	TP: Incident fatal IS
6.	11076.828	11439.57	TP: Incident non-fatal ICH
7.	11360.935	11653.667	TP: Incident fatal ICH
8.	11038.924	11375.529	TP: Bladder cancer death
9.	11179.742	11474.063	TP: Colorectal cancer death
10.	10835.45	11226.639	TP: Oesophageal cancer death
11.	11204.34	11528.996	TP: Kidney cancer death
12.	9534.7562	9774.7911	TP: Lung cancer death
13.	11379.029	11821.536	TP: Ovarian cancer death
14.	11267.119	11421.443	TP: Pancreatic cancer death
15.	11207.299	11088.48	TP: Pneumonia death
16.	11353.589	11544.426	TP: COPD death
17.	11238.388	11251.983	TP: ALS death
18.	11272.137	11321.276	TP: Parksinson´s disease death
19.	10570.306	10493.186	TP: Other death
20.	11710.381	11165.39	TP: Non-fatal MI following MI
21.	11692.968	11227.677	TP: Fatal MI following MI
22.	11603.052	11046.047	TP: Non-fatal stroke following MI
23.	11468.181	11173.754	TP: Fatal stroke following MI
24.	11447.107	11335.806	TP: Other death following MI
25.	11434.074	11433.187	TP: Non-fatal MI following stroke
26.	11434.074	11434.074	TP: Fatal MI following stroke
27.	11464.132	11288.424	TP: Non-fatal stroke following stroke
28.	11442.667	11498.484	TP: Fatal stroke following stroke
29.	11441.567	11434.074	TP: Other death following stroke
30.	11444.73	11357.599	TP: Non-fatal MI following MI and stroke
31.	11422.741	11433.466	TP: Fatal MI following MI and stroke
32.	11442.435	11341.657	TP: Non-fatal stroke following MI and stroke
33.	11421.212	11431.485	TP: Fatal stroke following MI and stroke
34.	11442.435	11433.823	TP: Other death following MI and stroke
35.	10902.453	11432.958	TP: Incidence of type 2 diabetes
36.	10813.418	11223.491	RF UKB: Mean cumulative LDL-C
37.	11354.249	11365.072	RF UKB: Mean cumulative Lp(a)
38.	11026.968	10967.903	RF UKB: Mean cumulative SBP
39.	11354.249	11291.407	RF UKB: Prevalence of type 2 diabetes
40.	11742.931	9826.8822	RF UKB: Mean LSI
41.	11624.084	11017.356	RF OC: LDL-C on incident MI
42.	10506.185	12725.748	RF OC: Lp(a) on incident MI
43.	10351.412	12096.827	RF OC: SBP on incident MI
44.	8058.9148	9708.177	RF OC: Smoking on incident MI
45.	10963.91	11236.772	RF OC: Type 2 diabetes on incident MI
46.	11150.178	11668.414	RF OC: LDL-C on incident IS
47.	11202.21	11081.504	RF OC: Lp(a) on incident IS
48.	11302.587	11021.169	RF OC: SBP on incident IS
49.	11024.237	11228.749	RF OC: Smoking on incident IS
50.	11290.447	10856.154	RF OC: Type 2 diabetes on incident IS
51.	11315.984	11147.262	RF OC: SBP on incident ICH
52.	14518.808	8984.7581	RF OC: Smoking on incident type 2 diabetes
53.	11016.9	11372.083	RF OC: Smoking on death from bladder cancer
54.	11302.595	11426.893	RF OC: Smoking on death from colorectal cancer
55.	11263.588	10206.787	RF OC: Smoking on death from oesophageal cancer
56.	11039.976	11384.915	RF OC: Smoking on death from kidney cancer
57.	13141.79	10477.022	RF OC: Smoking on death from lung cancer
58.	11383.523	11434.074	RF OC: Smoking on ovarian cancer
59.	10445.233	10333.481	RF OC: Smoking on pancreatic cancer
60.	11304.657	11401.907	RF OC: SBP on death from Pneumonia
61.	11050.177	11187.807	RF OC: Smoking on death from Pneumonia

62.	10718.361	11507.629	RF OC: Smoking on death from COPD
63.	11434.074	11383.523	RF OC: LDL-C on death from ALS
64.	11209.171	11157.172	RF OC: Smoking on death from Parkinson´s disease
65.	11435.169	11326.219	Effect of LDL-C lowering intervention
66.	8873.1163	12011.581	Effect of SBP lowering intervention
67.	11679.605	11221.792	Chronic utility for no CVD or diabetes
68.	11592.003	11280.391	Chronic utility for diabetes
69.	10438.407	12639.713	Chronic utility for MI
70.	11391.472	11476.996	Chronic utility for stroke
71.	11414.686	11453.528	Chronic utility for MI and stroke
72.	11429.69	11438.827	Chronic disutility for MI in diabetes
73.	11462.131	11405.674	Chronic disutility for stroke in diabetes
74.	11431.518	11436.676	Chronic disutility for MI and stroke in diabetes
75.	11428.999	11439.154	Acute disutility for MI
76.	11432.648	11435.501	Acute disutility for stroke
77.	11320.06	11574.524	Chronic cost of diabetes
78.	12582.897	10285.251	Chronic cost of MI
79.	11531.158	11336.99	Chronic cost of stroke
80.	11496.292	11371.856	Chronic cost of MI and stroke
81.	11511.932	11356.179	Chronic cost of MI in diabetes
82.	11327.34	11540.86	Chronic cost of stroke in diabetes
83.	11443.828	11424.316	Chronic cost of MI and stroke in diabetes
84.	11736.991	11131.158	Acute cost of MI
85.	11448.882	11419.267	Acute cost of stroke

```
. list icerl iceru AA if country == 2 & osa !=0, separator(0)
```

	icerl	iceru	AA
87.	-4476.2587	-3873.1584	TP: Incident non-fatal MI
88.	-4235.5097	-3395.3478	TP: Incident fatal MI
89.	-2351.959	-3368.2541	TP: Incident non-fatal IS
90.	-3782.8809	-4900.8506	TP: Incident fatal IS
91.	-3737.6709	-3324.2904	TP: Incident non-fatal ICH
92.	-3461.8685	-3548.5925	TP: Incident fatal ICH
93.	-3732.5151	-2787.3567	TP: Bladder cancer death
94.	-3150.579	-3695.3049	TP: Colorectal cancer death
95.	-4521.8784	-4061.6143	TP: Oesophageal cancer death
96.	-3086.2943	-4294.3658	TP: Kidney cancer death
97.	-5425.7973	-3514.6832	TP: Lung cancer death
98.	-3419.5831	-2728.1813	TP: Ovarian cancer death
99.	-3045.1243	-3323.2525	TP: Pancreatic cancer death
100.	-3630.3939	-3200.9326	TP: Pneumonia death
101.	-3648.1477	-3157.5848	TP: COPD death
102.	-3469.5142	-3374.859	TP: ALS death
103.	-3181.5947	-3453.2175	TP: Parksinson´s disease death
104.	-3649.2159	-4471.7603	TP: Other death
105.	-2808.1835	-2503.3817	TP: Non-fatal MI following MI
106.	-2900.9464	-2605.2618	TP: Fatal MI following MI
107.	-3120.3933	-3624.2117	TP: Non-fatal stroke following MI
108.	-3327.7838	-2889.9489	TP: Fatal stroke following MI
109.	-3480.0719	-2604.8533	TP: Other death following MI
110.	-3411.0417	-2806.6361	TP: Non-fatal MI following stroke
111.	-3411.0417	-3411.0417	TP: Fatal MI following stroke
112.	-3372.3597	-4135.486	TP: Non-fatal stroke following stroke
113.	-3431.0695	-3354.5629	TP: Fatal stroke following stroke
114.	-3470.7037	-3411.0417	TP: Other death following stroke
115.	-3169.3916	-3809.962	TP: Non-fatal MI following MI and stroke
116.	-3233.3946	-3573.5043	TP: Fatal MI following MI and stroke
117.	-3387.1282	-3876.7325	TP: Non-fatal stroke following MI and stroke
118.	-3454.2022	-3786.3667	TP: Fatal stroke following MI and stroke
119.	-3387.1282	-3256.1421	TP: Other death following MI and stroke
120.	-1511.1987	-4251.6696	TP: Incidence of type 2 diabetes
121.	-2589.1598	-4455.7332	RF UKB: Mean cumulative LDL-C
122.	-3548.4835	-3634.6155	RF UKB: Mean cumulative Lp(a)
123.	-2437.6628	-3997.1604	RF UKB: Mean cumulative SBP
124.	-3548.4835	-3685.5286	RF UKB: Prevalence of type 2 diabetes
125.	-2613.1103	-3727.3778	RF UKB: Mean LSI

```
126. | -2649.1971   -3398.3562                      RF OC: LDL-C on incident MI |
127. | -2380.6896    -6996.505                      RF OC: Lp(a) on incident MI |
128. | -1423.2179   -1833.3581                        RF OC: SBP on incident MI |
129. | -4669.7052    -7057.566                    RF OC: Smoking on incident MI |
130. | -4616.9115   -4424.1301           RF OC: Type 2 diabetes on incident MI |
131. | -4290.5105   -3258.4416                      RF OC: LDL-C on incident IS |
132. | -3508.9263   -3717.2763                      RF OC: Lp(a) on incident IS |
133. | -3553.3874   -4180.8061                        RF OC: SBP on incident IS |
134. | -4143.7331   -3043.2109                    RF OC: Smoking on incident IS |
135. | -3684.6421   -3529.6073           RF OC: Type 2 diabetes on incident IS |
136. | -3667.0596   -3777.3285                       RF OC: SBP on incident ICH |
137. | -444.07722   -6774.8953      RF OC: Smoking on incident type 2 diabetes |
138. | -4048.9914   -2644.2001     RF OC: Smoking on death from bladder cancer |
139. | -3563.0345   -2961.9399  RF OC: Smoking on death from colorectal cancer |
140. |   -3559.85   -5623.1191  RF OC: Smoking on death from oesophageal cancer |
141. | -4458.9566   -2670.0187      RF OC: Smoking on death from kidney cancer |
142. | -3844.3698   -4058.6456       RF OC: Smoking on death from lung cancer |
143. |  -3797.665   -3411.0417              RF OC: Smoking on ovarian cancer |
144. | -3275.6408   -5033.4368           RF OC: Smoking on pancreatic cancer |
145. | -3755.3086   -2961.5621            RF OC: SBP on death from Pneumonia |
146. | -4488.1274   -3661.3364         RF OC: Smoking on death from Pneumonia |
147. | -2962.6322   -2188.4586              RF OC: Smoking on death from COPD |
148. | -3411.0417    -3797.665                 RF OC: LDL-C on death from ALS |
149. | -3810.1153   -2967.7216  RF OC: Smoking on death from Parkinson´s disease |
150. |  -4913.717   -4498.9815          Effect of LDL-C lowering intervention |
151. | -1754.6727     -1772.12            Effect of SBP lowering intervention |
152. | -3589.9664   -3249.1056         Chronic utility for no CVD or diabetes |
153. | -3456.1804   -3367.0669                  Chronic utility for diabetes |
154. | -3117.3916   -3765.7672                        Chronic utility for MI |
155. | -3399.7318   -3422.4272                   Chronic utility for stroke |
156. | -3404.6062   -3417.5016             Chronic utility for MI and stroke |
157. | -3409.3487   -3412.8777           Chronic disutility for MI in diabetes |
158. | -3419.5546   -3402.4255      Chronic disutility for stroke in diabetes |
159. | -3409.7695   -3412.3368 Chronic disutility for MI and stroke in diabetes |
160. | -3409.5594   -3412.5254                       Acute disutility for MI |
161. | -3410.6167   -3411.4668                  Acute disutility for stroke |
162. | -3424.8846   -3396.5814                     Chronic cost of diabetes |
163. | -2923.5313   -3939.3867                           Chronic cost of MI |
164. |  -3174.288   -3724.4551                     Chronic cost of stroke |
165. | -2892.8061   -4176.9643                Chronic cost of MI and stroke |
166. | -3385.9669   -3439.3738               Chronic cost of MI in diabetes |
167. | -3558.0402   -3215.6924            Chronic cost of stroke in diabetes |
168. | -3371.7087   -3468.2328      Chronic cost of MI and stroke in diabetes |
169. | -3365.4207   -3456.6628                           Acute cost of MI |
170. | -3407.0959   -3414.9876                     Acute cost of stroke |
```

```
graph combine ///
GPH/OSAT_1.gph ///
GPH/OSAT_2.gph ///
, graphregion(color(white)) iscale(0.3) xsize(1.5) ysize(2.2) cols(2)
```

Figure 10.1: Tornado diagrams for one-way sensitivity analyses. These show the ICER at the lower and upper bounds of each model input.

10.2 Study 2

```
quietly {
foreach osa in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 403 ///
501 502 503 504 505 506 507 508 509 510 ///
601 602 603 604 605 606 607 608 609 ///
701 702 703 704 705 706 {
noisily di `osa´
forval rf = 1/2 {
foreach o in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 ///
133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 403 ///
501 502 503 504 505 506 507 508 ///
509 510 ///
601 602 603 604 605 606 607 608 ///
609 ///
701 702 703 704 705 706 {
local o`o´ = 0
}
if `rf´ == 1 {
local o`osa´ = -1.96
}
if `rf´ == 2 {
local o`osa´ = 1.96
}
local ll = 0.517
local sl = 20
local lp = 0.975
if `osa´ == 401 & `rf´ == 1 {
local ll = 0.512
}
if `osa´ == 401 & `rf´ == 2 {
local ll = 0.522
}
if `osa´ == 402 & `rf´ == 1 {
local sl = 15
}
if `osa´ == 402 & `rf´ == 2 {
local sl = 25
}
if `osa´ == 403 & `rf´ == 1 {
local lp = 0.94
}
if `osa´ == 403 & `rf´ == 2 {
local lp = 1
}
local o502 = 0.785
local o503 = 0.79
local o504 = 0.65
local o505 = 0.65
```

```
local o506 = -0.055
local o507 = -0.164
local o508 = -0.164
local o509 = -0.01125
local o510 = -0.03
if `osa´ == 502 & `rf´ == 1 {
local o502 = 0.681
}
if `osa´ == 502 & `rf´ == 2 {
local o502 = 0.889
}
if `osa´ == 503 & `rf´ == 1 {
local o503 = 0.73
}
if `osa´ == 503 & `rf´ == 2 {
local o503 = 0.85
}
if `osa´ == 504 & `rf´ == 1 {
local o504 = 0.63
}
if `osa´ == 504 & `rf´ == 2 {
local o504 = 0.67
}
if `osa´ == 505 & `rf´ == 1 {
local o505 = 0.63
}
if `osa´ == 505 & `rf´ == 2 {
local o505 = 0.67
}
if `osa´ == 506 & `rf´ == 1 {
local o506 = -0.067
}
if `osa´ == 506 & `rf´ == 2 {
local o506 = -0.042
}
if `osa´ == 507 & `rf´ == 1 {
local o507 = -0.222
}
if `osa´ == 507 & `rf´ == 2 {
local o507 = -0.105
}
if `osa´ == 508 & `rf´ == 1 {
local o508 = -0.222
}
if `osa´ == 508 & `rf´ == 2 {
local o508 = -0.105
}
if `osa´ == 509 & `rf´ == 1 {
local o509 = -0.0135
}
if `osa´ == 509 & `rf´ == 2 {
local o509 = -0.009
}
if `osa´ == 510 & `rf´ == 1 {
local o510 = -0.036
}
if `osa´ == 510 & `rf´ == 2 {
local o510 = -0.024
}
local o601 = 3588
local o602 = 6222
local o603 = 4388
local o604 = 6222
local o605 = 8870
local o606 = 8870
local o607 = 8870
local o6081 = 14434
local o6091 = 15659
```

```
local o6010 = 2546
local o6011 = 2170
local o6020 = 3304
local o6021 = 2917
local o6030 = 7021
local o6031 = 7351
local o6040 = 14442
local o6041 = 12616
local o6050 = 4511
local o6051 = 3917
local o6060 = 10014
local o6061 = 10494
local o6070 = 14442
local o6071 = 12616
local o6082 = 2212
local o6092 = 4626
if `osa´ == 601 & `rf´ == 1 {
local o601 = 2816
local o6010 = 2462
local o6011 = 2090
}
if `osa´ == 601 & `rf´ == 2 {
local o601 = 4539
local o6010 = 2633
local o6011 = 2254
}
if `osa´ == 602 & `rf´ == 1 {
local o602 = 6222*0.9
local o6020 = 3026
local o6021 = 2701
}
if `osa´ == 602 & `rf´ == 2 {
local o602 = 6222*1.1
local o6020 = 3607
local o6021 = 3149
}
if `osa´ == 603 & `rf´ == 1 {
local o603 = 4388*0.9
local o6030 = 5852
local o6031 = 5923
}
if `osa´ == 603 & `rf´ == 2 {
local o603 = 4388*1.1
local o6030 = 8421
local o6031 = 8062
}
if `osa´ == 604 & `rf´ == 1 {
local o604 = 6222*0.9
local o6040 = 9929
local o6041 = 8484
}
if `osa´ == 604 & `rf´ == 2 {
local o604 = 6222*1.1
local o6040 = 21004
local o6041 = 18756
}
if `osa´ == 605 & `rf´ == 1 {
local o605 = 6804
local o6050 = 3947
local o6051 = 3480
}
if `osa´ == 605 & `rf´ == 2 {
local o605 = 10937
local o6050 = 5155
local o6051 = 4409
}
if `osa´ == 606 & `rf´ == 1 {
local o606 = 6804
```

```
local o6060 = 7615
local o6061 = 7701
}
if `osa´ == 606 & `rf´ == 2 {
local o606 = 10937
local o6060 = 13167
local o6061 = 14300
}
if `osa´ == 607 & `rf´ == 1 {
local o607 = 6804
local o6070 = 9929
local o6071 = 8484
}
if `osa´ == 607 & `rf´ == 2 {
local o607 = 10937
local o6070 = 21004
local o6071 = 18756
}
if `osa´ == 608 & `rf´ == 1 {
local o6081 = 14434*0.9
local o6082 = 2212*0.9
}
if `osa´ == 608 & `rf´ == 2 {
local o6081 = 14434*1.1
local o6082 = 2212*1.1
}
if `osa´ == 609 & `rf´ == 1 {
local o6091 = 15659*0.9
local o6092 = 4626*0.9
}
if `osa´ == 609 & `rf´ == 2 {
local o6091 = 15659*1.1
local o6092 = 4626*1.1
}
local o701 = 55
local o702 = 90
local o703 = 21.5
local o704 = 6
local o705 = 1.46
local o706 = 1.92
if `osa´ == 701 & `rf´ == 1 {
local o701 = 44
}
if `osa´ == 701 & `rf´ == 2 {
local o701 = 66
}
if `osa´ == 702 & `rf´ == 1 {
local o702 = 72
}
if `osa´ == 702 & `rf´ == 2 {
local o702 = 108
}
if `osa´ == 703 & `rf´ == 1 {
local o703 = 17.2
}
if `osa´ == 703 & `rf´ == 2 {
local o703 = 25.8
}
if `osa´ == 704 & `rf´ == 1 {
local o704 = 2
}
if `osa´ == 704 & `rf´ == 2 {
local o704 = 10
}
if `osa´ == 705 & `rf´ == 1 {
local o705 = 1.36
}
if `osa´ == 705 & `rf´ == 2 {
```

```
local o705 = 1.55
}
if `osa´ == 706 & `rf´ == 1 {
local o706 = 1.80
}
if `osa´ == 706 & `rf´ == 2 {
local o706 = 2.06
}
if `osa´ <= 403 {
forval m = 0/1 {
use modstart2, clear
gen LPAT=.
gen LPT=.
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
}
replace vhr = 1 if MI == 1 | ST == 1
if `m´ == 1 {
replace LPAT=1 if vhr==1 & cycle == `c´
replace LPT=1 if lpa >= 90 & LPAT==1 & cycle == `c´ & vhr == 1
bysort ind (age) : replace lpa = lpa*(1-`lp´) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumldl=.
gen mcldl=.
```

```
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace aveldl = aveldl+(`o201´*errr)
drop errr
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace avelpa = avelpa+(`o202´*errr)
drop errr
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace avesbp = avesbp+(`o203´*errr)
drop errr
merge m:1 sex age using DMmod, keep(1 3) nogen
replace DMP = exp(ln(DMP)+(`o204´*errr))
drop errr
merge m:1 sex age using LSImod, keep(1 3) nogen
replace LSI = exp(ln(LSI)+(`o205´*errr))
drop errr
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
merge m:1 age sex using INC/`i´
drop _merge
rename (rate errr) (rate_`i´ errr_`i´)
}
replace rate_nfMI = exp(ln(rate_nfMI)+(`o101´*errr_nfMI))
replace rate_fMI = exp(ln(rate_fMI)+(`o102´*errr_fMI))
replace rate_nfIS = exp(ln(rate_nfIS)+(`o103´*errr_nfIS))
replace rate_fIS = exp(ln(rate_fIS)+(`o104´*errr_fIS))
replace rate_nfICH = exp(ln(rate_nfICH)+(`o105´*errr_nfICH))
replace rate_fICH = exp(ln(rate_fICH)+(`o106´*errr_fICH))
replace rate_blacdeath = exp(ln(rate_blacdeath)+(`o107´*errr_blacdeath))
replace rate_colcdeath = exp(ln(rate_colcdeath)+(`o108´*errr_colcdeath))
replace rate_oescdeath = exp(ln(rate_oescdeath)+(`o109´*errr_oescdeath))
```

```
replace rate_kidcdeath = exp(ln(rate_kidcdeath)+(`o110´*errr_kidcdeath))
replace rate_luncdeath = exp(ln(rate_luncdeath)+(`o111´*errr_luncdeath))
replace rate_ovacdeath = exp(ln(rate_ovacdeath)+(`o112´*errr_ovacdeath))
replace rate_pancdeath = exp(ln(rate_pancdeath)+(`o113´*errr_pancdeath))
replace rate_pneudeath = exp(ln(rate_pneudeath)+(`o114´*errr_pneudeath))
replace rate_copddeath = exp(ln(rate_copddeath)+(`o115´*errr_copddeath))
replace rate_alsdeath = exp(ln(rate_alsdeath)+(`o116´*errr_alsdeath))
replace rate_pddeath = exp(ln(rate_pddeath)+(`o117´*errr_pddeath))
replace rate_otherdeath = exp(ln(rate_otherdeath)+(`o118´*errr_otherdeath))
replace rate_t2d = exp(ln(rate_t2d)+(`o134´*errr_t2d))
replace rate_t2d = 0 if DM == 1
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
drop  errr_`i´
}
recast double rate_t2d-rate_nfICH
merge m:1 sex MI ST age using pevtpe, keep(1 3) nogen
replace ratenfMI = exp(ln(ratenfMI)+(`o119´*errrnfMI)) if MI == 1 & ST == 0
replace ratefMI = exp(ln(ratefMI)+(`o120´*errrfMI)) if MI == 1 & ST == 0
replace ratenfS = exp(ln(ratenfS)+(`o121´*errrnfS)) if MI == 1 & ST == 0
replace ratefS = exp(ln(ratefS)+(`o122´*errrfS)) if MI == 1 & ST == 0
replace rateothd = exp(ln(rateothd)+(`o123´*errrothd)) if MI == 1 & ST == 0
replace ratenfMI = exp(ln(ratenfMI)+(`o124´*errrnfMI)) if MI == 0 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o125´*errrfMI)) if MI == 0 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o126´*errrnfS)) if MI == 0 & ST == 1
replace ratefS = exp(ln(ratefS)+(`o127´*errrfS)) if MI == 0 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o128´*errrothd)) if MI == 0 & ST == 1
replace ratenfMI = exp(ln(ratenfMI)+(`o129´*errrnfMI)) if MI == 1 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o130´*errrfMI)) if MI == 1 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o131´*errrnfS)) if MI == 1 & ST == 1
replace ratefS = exp(ln(ratefS)+(`o132´*errrfS)) if MI == 1 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o133´*errrothd)) if MI == 1 & ST == 1
replace rate_t2d = rate_t2d*((exp(ln(1.21)+(`o312´*((ln(1.03)-ln(1.41))/3.92))))^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*((exp(ln(2.083)+(`o301´*((ln(2.222)-ln(2.000))/3.92))))^(mcldl-aveldl)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.0054)+(`o302´*((ln(1.0045)-ln(1.0062))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.058)+(`o303´*((ln(1.051)-ln(1.064))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.43)+(`o304´*((ln(1.22)-ln(1.62))/3.92))))^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(((exp(ln(1.26)+(`o305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´MI=(exp(ln(1.26)+(`o305´*((ln(1.40)-ln(1.08))/3.92))))*rate_`i´MI/(1+(((exp(ln(1.26)
> +(`o305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.08)+(`o306´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.0035)+(`o307´*((ln(1.0023)-ln(1.0045))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.027)+(`o308´*((ln(1.018)-ln(1.037))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.33)+(`o309´*((ln(1.22)-ln(1.46))/3.92))))^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(((exp(ln(1.74)+(`o310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´IS=(exp(ln(1.74)+(`o310´*((ln(2.47)-ln(1.19))/3.92))))*rate_`i´IS/(1+(((exp(ln(1.74)
> +(`o310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*((exp(ln(1.039)+(`o311´*((ln(1.010)-ln(1.069))/3.92))))^(mcsbp-avesb
> p))
replace rate`i´MI=rate`i´MI*((exp(ln(1.0054)+(`o302´*((ln(1.0045)-ln(1.0062))/3.92))))^(mclpa-avelpa
> ))
replace rate`i´S=rate`i´S*((exp(ln(1.0035)+(`o307´*((ln(1.0023)-ln(1.0045))/3.92))))^(mclpa-avelpa))
}
replace rate_blacdeath = rate_blacdeath*((exp(ln(2.52)+(`o313´*((ln(1.66)-ln(3.81))/3.92))))^(lsi-LS
> I))
replace rate_colcdeath = rate_colcdeath*((exp(ln(1.24)+(`o314´*((ln(1.06)-ln(1.44))/3.92))))^(lsi-LS
> I))
replace rate_oescdeath = rate_oescdeath*((exp(ln(3.67)+(`o315´*((ln(1.67)-ln(8.02))/3.92))))^(lsi-LS
> I))
```

```
replace rate_kidcdeath = rate_kidcdeath*((exp(ln(1.69)+(`o316´*((ln(1.04)-ln(3.05))/3.92))))^(lsi-LS
> I))
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`o317´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
> LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`o317´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
replace rate_ovacdeath = rate_ovacdeath*((exp(ln(1.27)+(`o318´*((ln(1.04)-ln(1.57))/3.92))))^(lsi-LS
> I))
replace rate_pancdeath = rate_pancdeath*((exp(ln(2.13)+(`o319´*((ln(1.15)-ln(3.90))/3.92))))^(lsi-LS
> I))
replace rate_pneudeath = rate_pneudeath*((exp(ln(1.016)+(`o320´*((ln(1.008)-ln(1.025))/3.92))))^(mcs
> bp-avesbp))
replace rate_pneudeath = rate_pneudeath*((exp(ln(4.03)+(`o321´*((ln(3.16)-ln(5.11))/3.92))))^(lsi-LS
> I))
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`o322´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
> LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`o322´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*((exp(ln(1.09)+(`o323´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-av
> eldl))
replace rate_pddeath = rate_pddeath*((exp(ln(0.48)+(`o324´*((ln(0.27)-ln(1.01))/3.92))))^(lsi-LSI))
recode rate_ovacdeath .=0
drop errrfMI-errrothd
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
```

```
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace LPT = LPT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT LPT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save OSA/modend2_`m´_`osa´_`rf´, replace
}
}
if `osa´ == 0 | inrange(`osa´,119,134) | `osa´ == 202 | `osa´ == 205 | `osa´ == 302 | `osa´ == 307 |
>  `osa´ == 312  | `osa´ == 403 {
forval m = 0/1 {
use modstartsp, clear
if `m´ == 0 {
gen LPAT=.
gen LPT=.
}
if `m´ == 1 {
gen LPAT=1 if cycle == 0
gen LPT=1 if lpa >= 90 & cycle == 0
}
forval c = 0/44 {
if `m´ == 1 {
bysort ind (age) : replace lpa = lpa*(1-`lp´) if cycle == `c´ & LPT == 1 & LPT[_n-1]!=1
}
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
replace mclpa = . if cycle!=`c´
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
gen errr = (avelpaub-avelpalb)/3.92
replace avelpa = avelpa+(`o202´*errr)
drop errr
merge m:1 sex age using LSImod, keep(1 3) nogen
replace LSI = exp(ln(LSI)+(`o205´*errr))
drop errr
merge m:1 age sex using INC/t2d
drop _merge
rename (rate errr) (rate_t2d errr_t2d)
replace rate_t2d = exp(ln(rate_t2d)+(`o134´*errr_t2d))
drop errr_t2d
replace rate_t2d = 0 if DM == 1
recast double rate_t2d
merge m:1 sex MI ST age using pevtpe, keep(1 3) nogen
replace ratenfMI = exp(ln(ratenfMI)+(`o119´*errrnfMI)) if MI == 1 & ST == 0
replace ratefMI = exp(ln(ratefMI)+(`o120´*errrfMI)) if MI == 1 & ST == 0
replace ratenfS = exp(ln(ratenfS)+(`o121´*errrnfS)) if MI == 1 & ST == 0
replace ratefS = exp(ln(ratefS)+(`o122´*errrfS)) if MI == 1 & ST == 0
replace rateothd = exp(ln(rateothd)+(`o123´*errrothd)) if MI == 1 & ST == 0
replace ratenfMI = exp(ln(ratenfMI)+(`o124´*errrnfMI)) if MI == 0 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o125´*errrfMI)) if MI == 0 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o126´*errrnfS)) if MI == 0 & ST == 1
replace ratefS = exp(ln(ratefS)+(`o127´*errrfS)) if MI == 0 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o128´*errrothd)) if MI == 0 & ST == 1
replace ratenfMI = exp(ln(ratenfMI)+(`o129´*errrnfMI)) if MI == 1 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`o130´*errrfMI)) if MI == 1 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`o131´*errrnfS)) if MI == 1 & ST == 1
```

```
replace ratefS = exp(ln(ratefS)+(`o132´*errrfS)) if MI == 1 & ST == 1
replace rateothd = exp(ln(rateothd)+(`o133´*errrothd)) if MI == 1 & ST == 1
replace rate_t2d = rate_t2d*((exp(ln(1.21)+(`o312´*((ln(1.03)-ln(1.41))/3.92))))^(lsi-LSI))
foreach i in nf f {
replace rate`i´MI=rate`i´MI*((exp(ln(1.0054)+(`o302´*((ln(1.0045)-ln(1.0062))/3.92))))^(mclpa-avelpa
> ))
replace rate`i´S=rate`i´S*((exp(ln(1.0035)+(`o307´*((ln(1.0023)-ln(1.0045))/3.92))))^(mclpa-avelpa))
}
drop errrfMI-errrothd
sort ind age
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
foreach var of varlist lpa lsi {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT LPT
}
save OSA/modendsp`m´_`osa´_`rf´, replace
}
}

forval c = 1/2 {
forval ii = 0/1 {
if `osa´ <= 403 {
use OSA/modend2_`ii´_`osa´_`rf´, clear
}
else {
use OSA/modend2_`ii´_0_1, clear
}
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.

if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
replace UT = invlogit(xb+`o501´*errr)
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
replace UT = UT+(UT*0.0255*`o501´)
replace UT = 0 if UT < 0
replace UT = 1 if UT > 1
}
```

```
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*`o503´
gen double STOHSQL = STOHS*UT*`o504´
gen double DMOHSQL = DMOHS*UT*`o502´
gen double MISHSQL = MISHS*UT*`o505´
gen double MIDHSQL = MIDHS*UT*(`o502´+`o506´)
gen double STDHSQL = STDHS*UT*(`o502´+`o507´)
gen double MSDHSQL = MSDHS*UT*(`o502´+`o508´)
replace MIOHSQL = MIOHSQL+`o509´ if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o509´ if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL+`o509´ if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o509´ if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL+`o510´ if STE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o510´ if STE == 1 & DTE==.
replace STDHSQL = STDHSQL+`o510´ if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o510´ if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*`o602´
gen double STOHSHC = STOHS*`o603´
gen double DMOHSHC = DMOHS*`o601´
gen double MISHSHC = MISHS*`o604´
gen double MIDHSHC = MIDHS*`o605´
gen double STDHSHC = STDHS*`o606´
gen double MSDHSHC = MSDHS*`o607´
gen double ACMIC = `o6081´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6081´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6091´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6091´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
```

```
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*`o6020´ if sex == 0
gen double STOHSHC = STOHS*`o6030´ if sex == 0
gen double DMOHSHC = DMOHS*`o6010´ if sex == 0
gen double MISHSHC = MISHS*`o6040´ if sex == 0
gen double MIDHSHC = MIDHS*`o6050´ if sex == 0
gen double STDHSHC = STDHS*`o6060´ if sex == 0
gen double MSDHSHC = MSDHS*`o6070´ if sex == 0
replace MIOHSHC = MIOHS*`o6021´ if sex == 1
replace STOHSHC = STOHS*`o6031´ if sex == 1
replace DMOHSHC = DMOHS*`o6011´ if sex == 1
replace MISHSHC = MISHS*`o6041´ if sex == 1
replace MIDHSHC = MIDHS*`o6051´ if sex == 1
replace STDHSHC = STDHS*`o6061´ if sex == 1
replace MSDHSHC = MSDHS*`o6071´ if sex == 1
gen double ACMIC = `o6082´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6082´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6092´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6092´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}


gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(`o705´*(1-WFP_GP))
gen STOHS_WFP = 1-(`o706´*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(`o706´*(1-WFP_GP))
gen MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-`o703´)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-`o703´)/365.25)
```

```
replace DMOHS_WFP= DMOHS_WFP*((365.25-`o704´)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-`o703´)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace STOHS_WFP = STOHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MIDHS_WFP = MIDHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace STDHS_WFP = STDHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace MSDHS_WFP = MSDHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MIOHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace STOHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace STDHS_WFP = STDHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
```

```
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save OSA/tabres2_`ii´_`osa´_`rf´_`c´, replace
}
forval i = 0/1 {
if inrange(`osa´,119,134) | `osa´ == 202 | `osa´ == 205 | `osa´ == 302 | `osa´ == 307 | `osa´ == 312
>   | `osa´ == 403 {
use OSA/modendsp`i´_`osa´_`rf´, clear
}
else {
use OSA/modendsp`i´_0_1, clear
}
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
replace UT = invlogit(xb+`o501´*errr)
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
replace UT = UT+(UT*0.0255*`o501´)
replace UT = 0 if UT < 0
replace UT = 1 if UT > 1
}

sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
```

```
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*`o503´
gen double STOHSQL = STOHS*UT*`o504´
gen double DMOHSQL = DMOHS*UT*`o502´
gen double MISHSQL = MISHS*UT*`o505´
gen double MIDHSQL = MIDHS*UT*(`o502´+`o506´)
gen double STDHSQL = STDHS*UT*(`o502´+`o507´)
gen double MSDHSQL = MSDHS*UT*(`o502´+`o508´)
replace MIOHSQL = MIOHSQL+`o509´ if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o509´ if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL+`o509´ if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o509´ if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL+`o510´ if STE == 1 & DTE==.
replace MISHSQL = MISHSQL+`o510´ if STE == 1 & DTE==.
replace STDHSQL = STDHSQL+`o510´ if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`o510´ if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*`o602´
gen double STOHSHC = STOHS*`o603´
gen double DMOHSHC = DMOHS*`o601´
gen double MISHSHC = MISHS*`o604´
gen double MIDHSHC = MIDHS*`o605´
gen double STDHSHC = STDHS*`o606´
gen double MSDHSHC = MSDHS*`o607´
gen double ACMIC = `o6081´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6081´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6091´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6091´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+4360 if LPT == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}

if `c´ == 2 {
gen double MIOHSHC = MIOHS*`o6020´ if sex == 0
gen double STOHSHC = STOHS*`o6030´ if sex == 0
gen double DMOHSHC = DMOHS*`o6010´ if sex == 0
gen double MISHSHC = MISHS*`o6040´ if sex == 0
gen double MIDHSHC = MIDHS*`o6050´ if sex == 0
gen double STDHSHC = STDHS*`o6060´ if sex == 0
gen double MSDHSHC = MSDHS*`o6070´ if sex == 0
replace MIOHSHC = MIOHS*`o6021´ if sex == 1
replace STOHSHC = STOHS*`o6031´ if sex == 1
replace DMOHSHC = DMOHS*`o6011´ if sex == 1
replace MISHSHC = MISHS*`o6041´ if sex == 1
replace MIDHSHC = MIDHS*`o6051´ if sex == 1
replace STDHSHC = STDHS*`o6061´ if sex == 1
replace MSDHSHC = MSDHS*`o6071´ if sex == 1
gen double ACMIC = `o6082´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`o6082´ if MIE == 1 & DTE == 1
gen double ACSTC = `o6092´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`o6092´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
```

```
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(`o705´*(1-WFP_GP))
gen STOHS_WFP = 1-(`o706´*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(`o706´*(1-WFP_GP))
gen MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`o705´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`o706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-`o703´)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-`o703´)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-`o704´)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-`o703´)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-`o703´-`o704´)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace STOHS_WFP = STOHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MIDHS_WFP = MIDHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace STDHS_WFP = STDHS_WFP*((365.25-`o702´)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`o701´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace MSDHS_WFP = MSDHS_WFP*((365.25-`o702´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MIOHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace STOHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace STDHS_WFP = STDHS_WFP*((182.625-`o702´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o701´)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`o702´)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
```

```
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}
gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MIE STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 38
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save OSA/tabressp_`i´_`osa´_`rf´_`c´, replace
}
}
}
}
}
```

11 Probabilistic sensitivity analyses

Disribution parameters are calculated as follows. For beta distributions,

$$\alpha = \mu^2(\frac{1-\mu}{\sigma^2} - \frac{1}{\mu})$$

and

$$\beta = \alpha(\frac{1}{\mu} - 1)$$

So for the utility values with a beta distribution, the chronic utilities for diabetes, MI, and stroke, the alpha and beta values are calculated, respectively, as:

$$\alpha_{DM} = 0.785^2(\frac{1-0.785}{(\frac{0.889-0.681}{3.92})^2} - \frac{1}{0.785}) = 46.3$$

$$\alpha_{MI} = 0.79^2(\frac{1-0.79}{(\frac{0.85-0.73}{3.92})^2} - \frac{1}{0.79}) = 139.1$$

$$\alpha_{ST} = 0.65^2(\frac{1-0.65}{(\frac{0.67-0.63}{3.92})^2} - \frac{1}{0.65}) = 1419.5$$

$$\beta_{DM} = 46.27(\frac{1}{0.785} - 1) = 12.7$$

$$\beta_{MI} = 139.07(\frac{1}{0.79} - 1) = 37.0$$

$$\beta_{ST} = 1419.54(\frac{1}{0.65} - 1) = 764.4$$

For gamma distributions,

$$k = \frac{\mu^2}{\sigma^2}$$

$$\theta = \frac{\sigma^2}{\mu}$$

For the costs with a gamma distribution, the chronic costs of diabetes, MI, and/or stroke, and acute costs of MI and stroke, the k and θ values are calculated, for Australia then the UK, respectively, as:

Australia:

$$k_{CDMA} = \frac{3588^2}{(\frac{4539-2816}{3.92})^2} = 66.6$$

$$k_{CMIA} = \frac{6222^2}{(\frac{1.1\times6222-0.9\times6222}{3.92})^2} = 384.2$$

$$k_{CSTA} = \frac{4388^2}{(\frac{1.1\times4388-0.9\times4388}{3.92})^2} = 384.2$$

$$k_{CDCA} = \frac{8870^2}{(\frac{10937-6804}{3.92})^2} = 70.8$$

$$k_{AMIA} = \frac{14434^2}{(\frac{1.1\times14434-0.9\times14434}{3.92})^2} = 384.2$$

$$k_{ASTA} = \frac{15659^2}{(\frac{1.1\times15659-0.9\times15659}{3.92})^2} = 384.2$$

$$\theta_{CDMA} = \frac{(\frac{4539-2816}{3.92})^2}{3588} = 53.8$$

$$\theta_{CMIA} = \frac{(\frac{1.1\times6222-0.9\times6222}{3.92})^2}{6222} = 16.2$$

$$\theta_{CSTA} = \frac{(\frac{1.1\times4388-0.9\times4388}{3.92})^2}{4388} = 11.4$$

$$\theta_{CDCA} = \frac{(\frac{10937-6804}{3.92})^2}{9112} = 125.3$$

$$\theta_{AMIA} = \frac{(\frac{1.1\times14434-0.9\times14434}{3.92})^2}{14434} = 37.6$$

$$\theta_{ASTA} = \frac{(\frac{1.1\times15659-0.9\times15659}{3.92})^2}{15659} = 40.8$$

UK:

$$k_{CDMUF} = \frac{2546^2}{(\frac{2633-2462}{3.92})^2} = 3406$$

$$k_{CDMUM} = \frac{2170^2}{(\frac{2254-2090}{3.92})^2} = 2690$$

$$k_{CMIUF} = \frac{3304^2}{(\frac{3607-3026}{3.92})^2} = 496.9$$

$$k_{CMIUM} = \frac{2917^2}{(\frac{3149-2701}{3.92})^2} = 651.5$$

$$k_{CSTUF} = \frac{7021^2}{(\frac{8421-5852}{3.92})^2} = 114.8$$

$$k_{CSTUM} = \frac{7351^2}{(\frac{8062-5923}{3.92})^2} = 181.5$$

$$k_{CMSUF} = \frac{14442^2}{(\frac{21004-9929}{3.92})^2} = 26.1$$

$$k_{CMSUM} = \frac{12616^2}{(\frac{18756-8484}{3.92})^2} = 23.2$$

$$k_{CMDUF} = \frac{4511^2}{(\frac{5155-3947}{3.92})^2} = 214.3$$

$$k_{CMDUM} = \frac{3917^2}{(\frac{4409-3480}{3.92})^2} = 273.2$$

$$k_{CSDUF} = \frac{10014^2}{(\frac{13167-7615}{3.92})^2} = 50.0$$

$$k_{CSDUM} = \frac{10494^2}{(\frac{14300-7701}{3.92})^2} = 38.9$$

$$k_{AMIU} = \frac{2212^2}{(\frac{1.1\times2212-0.9\times2212}{3.92})^2} = 384.2$$

$$k_{ASTU} = \frac{4626^2}{(\frac{1.1\times4626-0.9\times4626}{3.92})^2} = 384.2$$

$$\theta_{CDMUF} = \frac{(\frac{2633-2462}{3.92})^2}{2546} = 0.75$$

$$\theta_{CDMUM} = \frac{(\frac{2254-2090}{3.92})^2}{2170} = 0.81$$

$$\theta_{CMIUF} = \frac{(\frac{3607-3026}{3.92})^2}{3304} = 6.65$$

$$\theta_{CMIUM} = \frac{(\frac{3149-2701}{3.92})^2}{2917} = 4.48$$

$$\theta_{CSTUF} = \frac{(\frac{8421-5852}{3.92})^2}{7021} = 61.2$$

$$\theta_{CSTUM} = \frac{(\frac{8062-5923}{3.92})^2}{7351} = 40.5$$

$$\theta_{CMSUF} = \frac{(\frac{21004-9929}{3.92})^2}{14442} = 552.7$$

$$\theta_{CMSUM} = \frac{(\frac{18756-8484}{3.92})^2}{12616} = 544.3$$

$$\theta_{CMDUF} = \frac{(\frac{5155-3947}{3.92})^2}{4511} = 21.1$$

$$\theta_{CMDUM} = \frac{(\frac{4409-3480}{3.92})^2}{3917} = 14.3$$

$$\theta_{CSDUF} = \frac{(\frac{13167-7615}{3.92})^2}{10014} = 200.3$$

$$\theta_{CSDUM} = \frac{(\frac{14300-7701}{3.92})^2}{10494} = 270.0$$

$$\theta_{AMIU} = \frac{(\frac{1.1\times2212-0.9\times2212}{3.92})^2}{2212} = 5.8$$

$$\theta_{ASTU} = \frac{(\frac{1.1\times4626-0.9\times4626}{3.92})^2}{4626} = 12.0$$

11.1 Study 1

```
*mkdir PSA
quietly {
forval psa = 1/500 {
if `psa´ == 1 {
set seed 174697
}
if `psa´ == 51 {
set seed 42384
}
if `psa´ == 101 {
set seed 923887
}
if `psa´ == 151 {
set seed 7660735
}
if `psa´ == 201 {
set seed 525303
}
if `psa´ == 251 {
set seed 25007
}
if `psa´ == 301 {
set seed 15824
}
```

```
if `psa´ == 351 {
set seed 1606670
}
if `psa´ == 401 {
set seed 174697
}
if `psa´ == 451 {
set seed 3284454
}
noisily di `psa´
foreach p in 0 ///
101 102 103 104 105 106 107 108 ///
109 110 111 112 113 114 115 116 ///
117 118 119 120 121 122 123 124 ///
125 126 127 128 129 130 131 132 ///
133 134 ///
201 202 203 204 205 ///
301 302 303 304 305 306 307 308 ///
309 310 311 312 313 314 315 316 ///
317 318 319 320 321 322 323 324 ///
401 402 ///
501 502 503 504 505 506 507 508 ///
509 510 ///
601 602 603 604 605 606 607 608 ///
609 ///
701 702 703 704 705 706 {
local p`p´ = rnormal()
}
local ll = 0.517+(`p401´*((0.522-0.512)/3.92))
local sl = 20+(`p402´*((25-15)/3.92))
local p502 = rbeta(46.3,12.7)
local p503 = rbeta(139.1,37.0)
local p504 = rbeta(1419.5,764.4)
local p505 = rbeta(1419.5,764.4)
local p506 = -0.055 + `p506´*((0.067-0.042)/3.92)
local p507 = -0.164 + `p507´*((0.222-0.105)/3.92)
local p508 = -0.164 + `p508´*((0.222-0.105)/3.92)
local p509 = -0.01125 + `p509´*(((0.01125*1.1)-(0.01125*0.9))/3.92)
local p510 = -0.03 + `p510´*(((0.03*1.1)-(0.03*0.9))/3.92)
local p601 = rgamma(66.6,53.8)
local p602 = rgamma(384.2,16.2)
local p603 = rgamma(384.2,11.4)
local p604 = rgamma(384.2,16.2)
local p605 = rgamma(70.8,125.3)
local p606 = rgamma(70.8,125.3)
local p607 = rgamma(70.8,125.3)
local p6081 = rgamma(384.2,37.6)
local p6091 = rgamma(384.2,40.8)
local p6010 = rgamma(3406,0.75)
local p6011 = rgamma(2690,0.81)
local p6020 = rgamma(496.9,6.65)
local p6021 = rgamma(651.5,4.48)
local p6030 = rgamma(114.8,61.2)
local p6031 = rgamma(181.5,40.5)
local p6040 = rgamma(26.1,552.7)
local p6041 = rgamma(23.2,544.3)
local p6050 = rgamma(214.3,21.1)
local p6051 = rgamma(273.2,14.3)
local p6060 = rgamma(50.0,200.3)
local p6061 = rgamma(38.9,270.0)
local p6070 = rgamma(26.1,552.7)
local p6071 = rgamma(23.2,544.3)
local p6082 = rgamma(384.2,5.8)
local p6092 = rgamma(384.2,12.0)
local p701 = 55+(`p701´*(((55*1.2)-(55*0.8))/3.92))
local p702 = 90+(`p702´*(((90*1.2)-(90*0.8))/3.92))
local p703 = 21.5+(`p703´*(((21.5*1.2)-(21.5*0.8))/3.92))
local p704 = 6+(`p704´*((10-2)/3.92))
```

```
local p705 = exp(ln(1.46)+(`p705´*((ln(1.55)-ln(1.36))/3.92)))
local p706 = exp(ln(1.92)+(`p706´*((ln(2.06)-ln(1.80))/3.92)))
use modstart, clear
replace rand = runiform()
save PSA/modstart_psa_`psa´, replace
forval m = 0/1 {
use PSA/modstart_psa_`psa´, clear
forval c = 0/44 {
if `c´ == 0 | `c´ == 5 | `c´ == 10 | `c´ == 15 | `c´ == 20 ///
| `c´ == 25 |  `c´ == 30 | `c´ == 35 | `c´ == 40 {
gen tyr = 100*(1-0.9776^(exp( ///
(0.4648*((age-60)/5)) + ///
(0.7744*cs) + ///
(0.3131*((sbp-120)/20)) + ///
(0.1002*(ldl+hdl+0.5-6)) + ///
(-0.2606*((hdl-1.3)/0.5)) + ///
(-0.1088*(cs*(age-60)/5)) + ///
(-0.0277*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0226*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0613*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 0
replace tyr = 100*(1-0.9605^(exp( ///
(0.3742*((age-60)/5)) + ///
(0.6012*cs) + ///
(0.2777*((sbp-120)/20)) + ///
(0.1458*(ldl+hdl+0.5-6)) + ///
(-0.2698*((hdl-1.3)/0.5)) + ///
(-0.0755*(cs*(age-60)/5)) + ///
(-0.0255*(((sbp-120)/20)*((age-60)/5))) + ///
(-0.0281*((ldl+hdl+0.5-6)*((age-60)/5))) + ///
(0.0426*(((hdl-1.3)/0.5)*((age-60)/5))) ///
))) if cycle == `c´ & sex == 1
if `m´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
if `c´ == 0 {
gen LPAT=.
}
}
if `m´ == 1 {
if `c´ == 0 {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70)
gen LPAT=1 if vhr!=1 & cycle == `c´
replace vhr = 1 if lpa >= 90 & cycle == `c´
}
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | (tyr >=7.5 & age <50) | (tyr >= 10 & inrange(age,50,69)) |
> age >= 70 | lpa >= 90) & MI == 0 & ST == 0
}
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
}
}
```

```
else {
gen vhr = 1 if cycle == `c´ & (DM == 1 | age >= 70) & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & vhr == 1 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & vhr == 1 & sbp >= 140 & DT!=1 & MI == 0 & ST == 0
replace LLT = 1 if cycle == `c´ & ldl >= 5 & DT!=1 & MI == 0 & ST == 0
replace AHT = 1 if cycle == `c´ & sbp >= 160 & DT!=1 & MI == 0 & ST == 0
bysort ind (age) : replace ldl = ldl*(1-`ll´) if cycle == `c´ & LLT == 1 & LLT[_n-1]!=1 & MI == 0 &
> ST == 0
bysort ind (age) : replace sbp = sbp-`sl´ if cycle == `c´ & AHT == 1 & AHT[_n-1]!=1 & MI == 0 & ST =
> = 0
}
gen cumldl=.
gen mcldl=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen ldllog_`ii´ = ldl*logf_`ii´
bysort ind (age) : gen cumldllog_`ii´ = sum(ldllog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumldl = cumldllog_`ii´ if age == `ii´
replace mcldl = cumldllog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumldl
gen cumlpa=.
gen mclpa=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-3)/3)^(-2) if age <= `ii´
gen lpalog_`ii´ = lpa*logf_`ii´
bysort ind (age) : gen cumlpalog_`ii´ = sum(lpalog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumlpa = cumlpalog_`ii´ if age == `ii´
replace mclpa = cumlpalog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumlpa
gen cumsbp=.
gen mcsbp=.
forval ii = 0/84 {
gen logf_`ii´ = ((age-`ii´-2.1)/2.1)^(-2) if age <= `ii´
gen sbplog_`ii´ = sbp*logf_`ii´
bysort ind (age) : gen cumsbplog_`ii´ = sum(sbplog_`ii´)
bysort ind (age) : gen cumlog_`ii´ = sum(logf_`ii´)
replace cumsbp = cumsbplog_`ii´ if age == `ii´
replace mcsbp = cumsbplog_`ii´/cumlog_`ii´ if age == `ii´
}
drop logf_0-cumlog_84 cumsbp
replace mcldl = . if cycle!=`c´
replace mclpa = . if cycle!=`c´
replace mcsbp = . if cycle!=`c´
merge m:1 sex age using aveldl_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace aveldl = aveldl+(`p201´*errr)
drop errr
merge m:1 sex age using avelpa_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace avelpa = avelpa+(`p202´*errr)
drop errr
merge m:1 sex age using avesbp_cal, keep(1 3) nogen
gen errr = (aveldlub-aveldllb)/3.92
replace avesbp = avesbp+(`p203´*errr)
drop errr
merge m:1 sex age using DMmod, keep(1 3) nogen
replace DMP = exp(ln(DMP)+(`p204´*errr))
drop errr
merge m:1 sex age using LSImod, keep(1 3) nogen
replace LSI = exp(ln(LSI)+(`p205´*errr))
drop errr
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
```

```
merge m:1 age sex using INC/`i´
drop _merge
rename (rate errr) (rate_`i´ errr_`i´)
}
replace rate_nfMI = exp(ln(rate_nfMI)+(`p101´*errr_nfMI))
replace rate_fMI = exp(ln(rate_fMI)+(`p102´*errr_fMI))
replace rate_nfIS = exp(ln(rate_nfIS)+(`p103´*errr_nfIS))
replace rate_fIS = exp(ln(rate_fIS)+(`p104´*errr_fIS))
replace rate_nfICH = exp(ln(rate_nfICH)+(`p105´*errr_nfICH))
replace rate_fICH = exp(ln(rate_fICH)+(`p106´*errr_fICH))
replace rate_blacdeath = exp(ln(rate_blacdeath)+(`p107´*errr_blacdeath))
replace rate_colcdeath = exp(ln(rate_colcdeath)+(`p108´*errr_colcdeath))
replace rate_oescdeath = exp(ln(rate_oescdeath)+(`p109´*errr_oescdeath))
replace rate_kidcdeath = exp(ln(rate_kidcdeath)+(`p110´*errr_kidcdeath))
replace rate_luncdeath = exp(ln(rate_luncdeath)+(`p111´*errr_luncdeath))
replace rate_ovacdeath = exp(ln(rate_ovacdeath)+(`p112´*errr_ovacdeath))
replace rate_pancdeath = exp(ln(rate_pancdeath)+(`p113´*errr_pancdeath))
replace rate_pneudeath = exp(ln(rate_pneudeath)+(`p114´*errr_pneudeath))
replace rate_copddeath = exp(ln(rate_copddeath)+(`p115´*errr_copddeath))
replace rate_alsdeath = exp(ln(rate_alsdeath)+(`p116´*errr_alsdeath))
replace rate_pddeath = exp(ln(rate_pddeath)+(`p117´*errr_pddeath))
replace rate_otherdeath = exp(ln(rate_otherdeath)+(`p118´*errr_otherdeath))
replace rate_t2d = exp(ln(rate_t2d)+(`p134´*errr_t2d))
replace rate_t2d = 0 if DM == 1
foreach i in t2d oescdeath colcdeath pancdeath luncdeath ovacdeath kidcdeath ///
blacdeath pneudeath copddeath alsdeath pddeath otherdeath ///
fMI nfMI fIS nfIS fICH nfICH {
drop  errr_`i´
}
recast double rate_t2d-rate_nfICH
replace rate_t2d = rate_t2d*((exp(ln(1.21)+(`p312´*((ln(1.03)-ln(1.41))/3.92))))^(lsi-LSI))
foreach i in nf f {
replace rate_`i´MI=rate_`i´MI*((exp(ln(2.083)+(`p301´*((ln(2.222)-ln(2.000))/3.92))))^(mcldl-aveldl)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.0054)+(`p302´*((ln(1.0045)-ln(1.0062))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.058)+(`p303´*((ln(1.051)-ln(1.064))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´MI=rate_`i´MI*((exp(ln(1.43)+(`p304´*((ln(1.22)-ln(1.62))/3.92))))^(lsi-LSI))
replace rate_`i´MI=rate_`i´MI/(1+(((exp(ln(1.26)+(`p305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´MI=(exp(ln(1.26)+(`p305´*((ln(1.40)-ln(1.08))/3.92))))*rate_`i´MI/(1+(((exp(ln(1.26)
> +(`p305´*((ln(1.40)-ln(1.08))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.08)+(`p306´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-aveldl))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.0035)+(`p307´*((ln(1.0023)-ln(1.0045))/3.92))))^(mclpa-avel
> pa))
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.027)+(`p308´*((ln(1.018)-ln(1.037))/3.92))))^(mcsbp-avesbp)
> )
replace rate_`i´IS=rate_`i´IS*((exp(ln(1.33)+(`p309´*((ln(1.22)-ln(1.46))/3.92))))^(lsi-LSI))
replace rate_`i´IS=rate_`i´IS/(1+(((exp(ln(1.74)+(`p310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if D
> M == 0
replace rate_`i´IS=(exp(ln(1.74)+(`p310´*((ln(2.47)-ln(1.19))/3.92))))*rate_`i´IS/(1+(((exp(ln(1.74)
> +(`p310´*((ln(2.47)-ln(1.19))/3.92))))-1)*DMP)) if DM == 1
replace rate_`i´ICH=rate_`i´ICH*((exp(ln(1.039)+(`p311´*((ln(1.010)-ln(1.069))/3.92))))^(mcsbp-avesb
> p))
}
replace rate_blacdeath = rate_blacdeath*((exp(ln(2.52)+(`p313´*((ln(1.66)-ln(3.81))/3.92))))^(lsi-LS
> I))
replace rate_colcdeath = rate_colcdeath*((exp(ln(1.24)+(`p314´*((ln(1.06)-ln(1.44))/3.92))))^(lsi-LS
> I))
replace rate_oescdeath = rate_oescdeath*((exp(ln(3.67)+(`p315´*((ln(1.67)-ln(8.02))/3.92))))^(lsi-LS
> I))
replace rate_kidcdeath = rate_kidcdeath*((exp(ln(1.69)+(`p316´*((ln(1.04)-ln(3.05))/3.92))))^(lsi-LS
> I))
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`p317´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
> LSI)) if lsi-LSI <= 0.694*2
replace rate_luncdeath = rate_luncdeath*((exp(ln(13.64)+(`p317´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
```

```
replace rate_ovacdeath = rate_ovacdeath*((exp(ln(1.27)+(`p318´*((ln(1.04)-ln(1.57))/3.92))))^(lsi-LS
> I))
replace rate_pancdeath = rate_pancdeath*((exp(ln(2.13)+(`p319´*((ln(1.15)-ln(3.90))/3.92))))^(lsi-LS
> I))
replace rate_pneudeath = rate_pneudeath*((exp(ln(1.016)+(`p320´*((ln(1.008)-ln(1.025))/3.92))))^(mcs
> bp-avesbp))
replace rate_pneudeath = rate_pneudeath*((exp(ln(4.03)+(`p321´*((ln(3.16)-ln(5.11))/3.92))))^(lsi-LS
> I))
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`p322´*((ln(8.85)-ln(21.03))/3.92))))^(lsi-
> LSI)) if lsi-LSI <= 0.694*2
replace rate_copddeath = rate_copddeath*((exp(ln(13.64)+(`p322´*((ln(8.85)-ln(21.03))/3.92))))^(0.69
> 2*2)) if lsi-LSI > 0.694*2
replace rate_alsdeath = rate_alsdeath*((exp(ln(1.09)+(`p323´*((ln(1.03)-ln(1.14))/3.92))))^(mcldl-av
> eldl))
replace rate_pddeath = rate_pddeath*((exp(ln(0.48)+(`p324´*((ln(0.27)-ln(1.01))/3.92))))^(lsi-LSI))
recode rate_ovacdeath .=0
merge m:1 sex MI ST age using pevtpe, keep(1 3) nogen
replace ratenfMI = exp(ln(ratenfMI)+(`p119´*errrnfMI)) if MI == 1 & ST == 0
replace ratefMI = exp(ln(ratefMI)+(`p120´*errrfMI)) if MI == 1 & ST == 0
replace ratenfS = exp(ln(ratenfS)+(`p121´*errrnfS)) if MI == 1 & ST == 0
replace ratefS = exp(ln(ratefS)+(`p122´*errrfS)) if MI == 1 & ST == 0
replace rateothd = exp(ln(rateothd)+(`p123´*errrothd)) if MI == 1 & ST == 0
replace ratenfMI = exp(ln(ratenfMI)+(`p124´*errrnfMI)) if MI == 0 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`p125´*errrfMI)) if MI == 0 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`p126´*errrnfS)) if MI == 0 & ST == 1
replace ratefS = exp(ln(ratefS)+(`p127´*errrfS)) if MI == 0 & ST == 1
replace rateothd = exp(ln(rateothd)+(`p128´*errrothd)) if MI == 0 & ST == 1
replace ratenfMI = exp(ln(ratenfMI)+(`p129´*errrnfMI)) if MI == 1 & ST == 1
replace ratefMI = exp(ln(ratefMI)+(`p130´*errrfMI)) if MI == 1 & ST == 1
replace ratenfS = exp(ln(ratenfS)+(`p131´*errrnfS)) if MI == 1 & ST == 1
replace ratefS = exp(ln(ratefS)+(`p132´*errrfS)) if MI == 1 & ST == 1
replace rateothd = exp(ln(rateothd)+(`p133´*errrothd)) if MI == 1 & ST == 1
drop errrfMI-errrothd
sort ind age
gen ratesum0 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace ratesum0 = ratesum0+`var´ if MI == 0 & ST == 0
}
gen tpsum0 = 1-exp(-ratesum0)
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = tpsum0*`var´/ratesum0 if MI == 0 & ST == 0
}
gen ratesum1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace ratesum1 = ratesum1+`var´ if MI == 1 | ST == 1
}
gen tpsum1 = 1-exp(-ratesum1)
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = tpsum1*`var´/ratesum1 if MI == 1 | ST == 1
}
local var1 = 0
foreach var of varlist rate_t2d-rate_nfICH {
replace `var´ = `var´+`var1´ if MI == 0 & ST == 0
local var1 = "`var´"
}
local var1 = 0
foreach var of varlist rate_t2d ratefMI-rateothd {
replace `var´ = `var´+`var1´ if MI == 1 | ST == 1
local var1 = "`var´"
}
replace rand = . if DT == 1
replace DME=1 if inrange(rand,0,rate_t2d) & DM == 0 & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,ratefMI) | inrange(rand,ratenfMI,ratefS) | inrange(rand,rate
> nfS,rateothd)) & (MI == 1 | ST == 1) & cycle == `c´
replace MIE=1 if inrange(rand,rate_t2d,ratenfMI) & (MI == 1 | ST == 1) & cycle == `c´
replace STE=1 if inrange(rand,ratenfMI,ratenfS) & (MI == 1 | ST == 1) & cycle == `c´
replace DTE=1 if (inrange(rand,rate_t2d,rate_fMI) | inrange(rand,rate_nfMI,rate_fIS) | inrange(rand,
> rate_nfIS,rate_fICH)) & MI==0 & ST == 0 & cycle == `c´
```

```
replace MIE=1 if inrange(rand,rate_otherdeath,rate_nfMI) & MI== 0 & ST == 0 & cycle == `c´
replace STE=1 if inrange(rand,rate_nfMI,rate_nfICH) & MI== 0 & ST == 0 & cycle == `c´
bysort ind (age) : replace DT = max(DT[_n-1],DTE[_n-1]) if cycle[_n-1]==`c´
foreach var of varlist DM MI ST {
bysort ind (age) : replace `var´ = max(`var´[_n-1],`var´E[_n-1]) if cycle[_n-1]==`c´
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
bysort ind (age) : replace ldl = ldl[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace lpa = lpa[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace sbp = sbp[_n-1]+0.91 if cycle[_n-1]==`c´ & sex == 0
bysort ind (age) : replace sbp = sbp[_n-1]+0.56 if cycle[_n-1]==`c´ & sex == 1
bysort ind (age) : replace LLT = LLT[_n-1] if cycle[_n-1]==`c´
bysort ind (age) : replace AHT = AHT[_n-1] if cycle[_n-1]==`c´
foreach var of varlist hdl-lsi LLT AHT {
bysort ind (age) : replace `var´ = . if cycle[_n-1]==`c´ & (DTE[_n-1]==1 | DT[_n-1]==1)
}
keep ind-rand LPAT
}
save PSA/modend`m´_`psa´, replace
}
forval c = 1/2 {
forval i = 0/1 {
use PSA/modend`i´_`psa´, clear
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= 90
keep if cycle!=.
if `c´ == 1 {
merge m:1 sex age using UTvals_AU, keep(3) nogen
replace UT = invlogit(xb+`p501´*errr)
drop xb errr UTlb UTub
}
if `c´ == 2 {
merge m:1 sex age using UTvals_UK, keep(3) nogen
replace UT = UT+(UT*0.0255*`p501´)
replace UT = 0 if UT < 0
replace UT = 1 if UT > 1
}
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
```

```
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*`p503´
gen double STOHSQL = STOHS*UT*`p504´
gen double DMOHSQL = DMOHS*UT*`p502´
gen double MISHSQL = MISHS*UT*`p505´
gen double MIDHSQL = MIDHS*UT*(`p502´+`p506´)
gen double STDHSQL = STDHS*UT*(`p502´+`p507´)
gen double MSDHSQL = MSDHS*UT*(`p502´+`p508´)
replace MIOHSQL = MIOHSQL+`p509´ if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL+`p509´ if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL+`p509´ if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`p509´ if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL+`p510´ if STE == 1 & DTE==.
replace MISHSQL = MISHSQL+`p510´ if STE == 1 & DTE==.
replace STDHSQL = STDHSQL+`p510´ if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL+`p510´ if STE == 1 & DTE==.

if `c´ == 1 {
gen double MIOHSHC = MIOHS*`p602´
gen double STOHSHC = STOHS*`p603´
gen double DMOHSHC = DMOHS*`p601´
gen double MISHSHC = MISHS*`p604´
gen double MIDHSHC = MIDHS*`p605´
gen double STDHSHC = STDHS*`p606´
gen double MSDHSHC = MSDHS*`p607´
gen double ACMIC = `p6081´ if MIE == 1 & DTE == .
replace ACMIC = 0.233*`p6081´ if MIE == 1 & DTE == 1
gen double ACSTC = `p6091´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`p6091´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+200 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+200+143+212 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 25 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.801 if sex == 0 & inrange(age,40,44)
replace WFP_GP = 0.801 if sex == 0 & inrange(age,45,54)
replace WFP_GP = 0.622 if sex == 0 & inrange(age,55,66)
replace WFP_GP = 0.890 if sex == 1 & inrange(age,40,44)
replace WFP_GP = 0.865 if sex == 1 & inrange(age,45,54)
replace WFP_GP = 0.725 if sex == 1 & inrange(age,55,66)
}
if `c´ == 2 {
gen double MIOHSHC = MIOHS*`p6020´ if sex == 0
gen double STOHSHC = STOHS*`p6030´ if sex == 0
gen double DMOHSHC = DMOHS*`p6010´ if sex == 0
gen double MISHSHC = MISHS*`p6040´ if sex == 0
gen double MIDHSHC = MIDHS*`p6050´ if sex == 0
gen double STDHSHC = STDHS*`p6060´ if sex == 0
gen double MSDHSHC = MSDHS*`p6070´ if sex == 0
replace MIOHSHC = MIOHS*`p6021´ if sex == 1
replace STOHSHC = STOHS*`p6031´ if sex == 1
replace DMOHSHC = DMOHS*`p6011´ if sex == 1
replace MISHSHC = MISHS*`p6041´ if sex == 1
replace MIDHSHC = MIDHS*`p6051´ if sex == 1
replace STDHSHC = STDHS*`p6061´ if sex == 1
replace MSDHSHC = MSDHS*`p6071´ if sex == 1
gen double ACMIC = `p6082´ if MIE == 1 & DTE == .
```

```
replace ACMIC = 0.233*`p6082´ if MIE == 1 & DTE == 1
gen double ACSTC = `p6092´ if STE == 1 & DTE ==.
replace ACSTC = 0.841*`p6092´ if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
}

gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(`p705´*(1-WFP_GP))
gen STOHS_WFP = 1-(`p706´*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(`p706´*(1-WFP_GP))
gen MIDHS_WFP = 1-(`p705´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`p705´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(`p705´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(`p705´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(`p706´*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-`p703´)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-`p703´)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-`p704´)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-`p703´)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-`p703´-`p704´)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-`p703´-`p704´)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-`p703´-`p704´)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-`p701´)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace STOHS_WFP = STOHS_WFP*((365.25-`p702´)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`p701´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`p702´)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & D
> TE==.
replace MIDHS_WFP = MIDHS_WFP*((365.25-`p701´)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace STDHS_WFP = STDHS_WFP*((365.25-`p702´)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MISHS_WFP = MISHS_WFP*((365.25-`p701´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & D
> TE==.
replace MSDHS_WFP = MSDHS_WFP*((365.25-`p702´)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & D
> TE==.
replace MIOHS_WFP = MIOHS_WFP*((182.625-`p701´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace STOHS_WFP = MIOHS_WFP*((182.625-`p702´)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`p701´)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 &
>  DTE==.
replace MISHS_WFP = MIOHS_WFP*((182.625-`p702´)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 &
>  DTE==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-`p701´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
```

```
replace STDHS_WFP = STDHS_WFP*((182.625-`p702´)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`p701´)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 &
>  DTE==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-`p702´)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 &
>  DTE==.

if `c´ == 1 {
gen WFP = 0 if age < 67
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*73003
gen DC = 1/((1.05)^(cycle))
}
if `c´ == 2 {
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
}

gen N = 1 if cycle == 0
foreach var of varlist HEAHS-LPATHC INDC {
gen double `var´_DC = `var´*DC
}
collapse (sum) N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC INDC_DC
gen double YLL = HEAHS_DC+MIOHS_DC+STOHS_DC+DMOHS_DC+MISHS_DC+MIDHS_DC+STDHS_DC+MSDHS_DC
gen double QLY = HEAHSQL_DC+MIOHSQL_DC+STOHSQL_DC+DMOHSQL_DC+MISHSQL_DC+MIDHSQL_DC+STDHSQL_DC+MSDHSQ
> L_DC
gen double HCC = MIOHSHC_DC+STOHSHC_DC+DMOHSHC_DC+MISHSHC_DC+MIDHSHC_DC+STDHSHC_DC+MSDHSHC_DC+ACMIC_
> DC+ACSTC_DC+DRUGSHC_DC+LPATHC_DC
gen double TSC = HCC+INDC_DC
order N LPAT LPATT MI1 MIE ST1 STE DTE HEAHS_DC-LPATHC_DC YLL QLY HCC INDC TSC
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist N-TSC {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
gen i = `i´
gen psa = `psa´
gen c = `c´
save PSA/tabres_`i´_`psa´_`c´, replace
}
}
erase PSA/modstart_psa_`psa´.dta
forval m = 0/1 {
erase PSA/modend`m´_`psa´.dta
}
}
}
clear
forval c = 1/2 {
forval psa = 1/500 {
use PSA/tabres_0_`psa´_`c´, clear
merge 1:1 _n using PSA/tabres_1_`psa´_`c´, nogen
gen double diff = val1-val0
expand 3 if stat=="TSC"
replace stat = "ICER" if _n == 41
replace stat = "SICER" if _n == 42
```

```
replace val0 = . if _n>40
replace val1 = . if _n>40
replace diff = diff[38]/diff[37] if _n==41
replace diff = diff[40]/diff[37] if _n==42
drop i
gen njm = _n
save PSA/tabres_`psa´_`c´, replace
}
}
clear
forval c = 1/2 {
forval psa = 1/500 {
append using PSA/tabres_`psa´_`c´
}
}
save PSA/Res, replace
forval c = 1/2 {
use PSA/Res, clear
keep if c == `c´
mat A = (.,.,.,.)
mat B = (.,.,.,.)
mat C = (.,.,.,.)
forval i = 1/42 {
centile val0 if njm == `i´, centile(50 2.5 97.5)
mat A = (A\0`i´,r(c_1),r(c_2),r(c_3))
centile val1 if njm == `i´, centile(50 2.5 97.5)
mat B = (B\0`i´,r(c_1),r(c_2),r(c_3))
centile diff if njm == `i´, centile(50 2.5 97.5)
mat C = (C\0`i´,r(c_1),r(c_2),r(c_3))
}
mat A = A[2..43,1..4]
mat B = B[2..43,1..4]
mat C = C[2..43,1..4]
keep if _n <= 42
svmat double A
svmat double B
svmat double C
drop val0-diff
tostring A1-C4, force format(%15.0fc) replace
gen A = A2 + " (" + A3 + ", " + A4 + ")"
gen B = B2 + " (" + B3 + ", " + B4 + ")"
gen C = C2 + " (" + C3 + ", " + C4 + ")"
gen row = ""
replace row = "Population size" if _n == 1
replace row = "Lp(a) tests" if _n == 2
replace row = "Treatment modified in response to Lp(a) test" if _n == 3
replace row = "Incident MI (N)" if _n == 4
replace row = "Total MIs (N)" if _n == 5
replace row = "Incident stroke (N)" if _n == 6
replace row = "Total strokes (N)" if _n == 7
replace row = "Deaths (N)" if _n == 8
replace row = "YLL no CVD or diabetes" if _n == 9
replace row = "YLL with MI" if _n == 10
replace row = "YLL with stroke" if _n == 11
replace row = "YLL with diabetes" if _n == 12
replace row = "YLL with MI and stroke" if _n == 13
replace row = "YLL with MI and diabetes" if _n == 14
replace row = "YLL with stroke and diabetes" if _n == 15
replace row = "YLL with MI, stroke, and diabetes" if _n == 16
replace row = "QALY with no CVD or diabetes" if _n == 17
replace row = "QALY with MI" if _n == 18
replace row = "QALY with stroke" if _n == 19
replace row = "QALY with diabetes" if _n == 20
replace row = "QALY with MI and stroke" if _n == 21
replace row = "QALY with MI and diabetes" if _n == 22
replace row = "QALY with stroke and diabetes" if _n == 23
replace row = "QALY with MI, stroke, and diabetes" if _n == 24
replace row = "Chronic MI healthcare costs (\textsterling)" if _n == 25
```

```
replace row = "Chronic stroke healthcare costs (\textsterling)" if _n == 26
replace row = "Chronic diabetes healthcare costs (\textsterling)" if _n == 27
replace row = "Chronic MI and stroke healthcare costs (\textsterling)" if _n == 28
replace row = "Chronic MI and diabetes healthcare costs (\textsterling)" if _n == 29
replace row = "Chronic stroke and diabetes healthcare costs (\textsterling)" if _n == 30
replace row = "Chronic MI, diabetes, and stroke healthcare costs (\textsterling)" if _n == 31
replace row = "Acute MI costs (\textsterling)" if _n == 32
replace row = "Acute stroke costs (\textsterling)" if _n == 33
replace row = "Medication costs (\textsterling)" if _n == 34
replace row = "Lp(a) test costs (\textsterling)" if _n == 35
replace row = "Total YLL" if _n == 36
replace row = "Total QALY" if _n == 37
replace row = "Total healthcare costs (\textsterling)" if _n == 38
replace row = "Total indirect costs (\textsterling)" if _n == 39
replace row = "Total costs (\textsterling)" if _n == 40
replace row = "ICER (\textsterling \ per QALY)" if _n == 41
replace row = "SICER (\textsterling \ per QALY)" if _n == 42
if `c´ == 1 {
replace row = subinstr(row,"\textsterling","\\$",.)
}
keep row A B C
order row A B C
replace A = "" if A == ". (., .)"
replace B = "" if B == ". (., .)"
export delimited using CSV/PSArestab_`c´.csv, delimiter(":") novarnames replace
drop if inrange(_n,9,31)
save tempsumtabpsa_`c´, replace
}
clear
append using tempsumtabpsa_1 tempsumtabpsa_2
drop if inrange(_n,20,27)
gen CC = "Overall" if _n == 1
replace CC = "Australia" if _n == 9
replace CC = "UK" if _n == 20
order CC
export delimited using CSV/PSArestabsumpaper.csv, delimiter(":") novarnames replace
use PSA/Res, clear
keep if njm ==38 | njm == 37
bysort c psa (njm) : gen double diff1 = diff[_n+1]
rename (diff diff1) (qly hcc)
drop if njm == 38
keep psa c qly hcc
replace hcc = hcc/1000000
forval c = 1/2 {
if `c´ == 1 {
local cc = "Australia"
local wtp = 28000
local cur = "AUD"
local rr = 54
}
else {
local cc = "UK"
local wtp = 20000
local cur = "GBP"
local rr = 75
}
twoway ///
(scatter hcc qly if c == `c´, msize(vsmall) col(magenta)) ///
(function y = x*`wtp´/1000000, ra(0 `rr´) col(dknavy)) ///
, graphregion(color(white)) ///
legend(off) ///
ytitle("Incremental costs (`cur´, millions)") xtitle("Incremental QALYs") ///
xline(0, lcol(gs12)) yline(0, lcol(gs12)) xscale(range(0 200)) yscale(range(-1 1.5)) ///
ylabel(-2(0.5)1.5, angle(0) format(%9.1fc) nogrid) xlabel(0(50)200, format(%9.0f)) ///
title("`cc´, healthcare perspective", col(black) size(medium) placement(west))
graph save GPH/PSA_`c´, replace
}
use PSA/Res, clear
```

```
keep if njm ==40 | njm == 37
bysort c psa (njm) : gen double diff1 = diff[_n+1]
rename (diff diff1) (qly hcc)
drop if njm == 40
keep psa c qly hcc
replace hcc = hcc/1000000
forval c = 1/2 {
if `c´ == 1 {
local cc = "Australia"
local wtp = 28000
local cur = "AUD"
local rr = 54
}
else {
local cc = "UK"
local wtp = 20000
local cur = "GBP"
local rr = 75
}
twoway ///
(scatter hcc qly if c == `c´, msize(vsmall) col(magenta)) ///
(function y = x*`wtp´/1000000, ra(0 `rr´) col(dknavy)) ///
, graphregion(color(white)) ///
legend(off) ///
ytitle("Incremental costs (`cur´, millions)") xtitle("Incremental QALYs") ///
xline(0, lcol(gs12)) yline(0, lcol(gs12)) xscale(range(0 200)) yscale(range(-1 1.5)) ///
ylabel(-3(0.5)1.5, angle(0) format(%9.1fc) nogrid) xlabel(0(50)200, format(%9.0f)) ///
title("`cc´, societal perspective", col(black) size(medium) placement(west))
graph save GPH/PSAS_`c´, replace
}
```

11.1.1 Results

Table 11.1: Probabilistic sensitivity analysis results for Lp(a) testing (intervention) compared to standard of care (control). Australia.

Outcome	Control	Intervention	Difference
Population size	10,000 (10,000, 10,000)	10,000 (10,000, 10,000)	0 (0, 0)
Lp(a) tests	0 (0, 0)	8,993 (8,993, 8,993)	8,993 (8,993, 8,993)
Treatment modified in response to Lp(a) test	0 (0, 0)	624 (624, 624)	624 (624, 624)
Incident MI (N)	706 (589, 852)	678 (563, 824)	-27 (-37, -18)
Total MIs (N)	853 (703, 1,023)	820 (674, 983)	-34 (-49, -22)
Incident stroke (N)	592 (465, 751)	590 (460, 750)	-3 (-12, 6)
Total strokes (N)	753 (595, 946)	746 (588, 941)	-4 (-20, 12)
Deaths (N)	3,862 (3,378, 5,153)	3,854 (3,371, 5,151)	-7 (-17, 2)
YLL no CVD or diabetes	119,615 (118,167, 120,842)	119,838 (118,398, 121,011)	204 (135, 295)
YLL with MI	2,499 (2,103, 2,950)	2,341 (1,975, 2,766)	-151 (-223, -69)
YLL with stroke	980 (782, 1,216)	968 (777, 1,193)	-8 (-61, 47)
YLL with diabetes	17,001 (16,319, 17,642)	17,025 (16,349, 17,678)	21 (3, 45)
YLL with MI and stroke	94 (55, 148)	91 (54, 145)	-3 (-16, 7)
YLL with MI and diabetes	432 (317, 569)	418 (302, 561)	-14 (-36, 4)
YLL with stroke and diabetes	321 (212, 448)	320 (207, 446)	-0 (-16, 14)
YLL with MI, stroke, and diabetes	25 (9, 46)	24 (9, 45)	0 (-6, 3)
QALY with no CVD or diabetes	103,495 (101,335, 105,740)	103,680 (101,499, 105,935)	175 (115, 253)
QALY with MI	1,697 (1,398, 2,032)	1,601 (1,301, 1,916)	-103 (-153, -48)
QALY with stroke	536 (425, 664)	529 (424, 650)	-5 (-33, 26)
QALY with diabetes	11,471 (9,344, 12,949)	11,482 (9,354, 12,970)	14 (2, 31)
QALY with MI and stroke	52 (30, 82)	50 (29, 81)	-2 (-9, 4)
QALY with MI and diabetes	267 (184, 378)	259 (179, 367)	-9 (-24, 3)
QALY with stroke and diabetes	164 (109, 247)	165 (108, 246)	-0 (-9, 8)
QALY with MI, stroke, and diabetes	13 (4, 25)	12 (4, 23)	0 (-3, 2)
Chronic MI healthcare costs ($)	15,623,331 (12,728,290, 19,143,684)	14,693,482 (12,016,433, 17,994,807)	-946,612 (-1,457,609, -440,285)
Chronic stroke healthcare costs ($)	4,273,766 (3,329,714, 5,405,303)	4,255,822 (3,323,210, 5,415,251)	-36,617 (-263,478, 201,660)
Chronic diabetes healthcare costs ($)	61,323,109 (47,647,245, 75,874,422)	61,367,495 (47,722,902, 75,942,652)	74,593 (13,392, 166,441)
Chronic MI and stroke healthcare costs ($)	588,527 (343,707, 939,950)	565,349 (326,742, 928,908)	-17,446 (-96,047, 45,372)
Chronic MI and diabetes healthcare costs ($)	3,873,013 (2,514,738, 5,396,069)	3,734,017 (2,457,552, 5,265,739)	-123,969 (-344,923, 33,503)
Chronic stroke and diabetes healthcare costs ($)	2,739,776 (1,806,825, 4,343,886)	2,737,609 (1,800,126, 4,330,291)	-3,089 (-143,487, 133,073)
Chronic MI, diabetes, and stroke healthcare costs ($)	220,625 (75,015, 406,502)	216,694 (77,094, 403,213)	0 (-55,802, 25,898)
Acute MI costs ($)	5,780,574 (4,692,003, 7,026,919)	5,508,051 (4,471,742, 6,673,717)	-274,279 (-402,577, -154,279)
Acute stroke costs ($)	4,975,699 (4,033,393, 6,262,238)	4,936,471 (3,972,196, 6,244,173)	-28,267 (-154,927, 91,463)
Medication costs ($)	43,232,701 (42,348,745, 44,023,181)	45,133,906 (44,180,354, 45,917,632)	1,900,115 (1,833,956, 1,962,155)
Lp(a) test costs ($)	0 (0, 0)	224,825 (224,825, 224,825)	224,825 (224,825, 224,825)
Total YLL	141,027 (139,466, 142,365)	141,080 (139,507, 142,430)	49 (2, 105)
Total QALY	117,624 (114,349, 120,369)	117,703 (114,436, 120,442)	72 (32, 117)
Total healthcare costs ($)	142,940,821 (128,000,763, 159,985,264)	143,580,902 (128,757,820, 160,727,473)	793,032 (133,542, 1,292,857)
Total indirect costs ($)	116,591,701 (104,250,395, 133,868,239)	115,345,588 (102,960,312, 131,774,929)	-1,340,495 (-2,579,312, -387,370)
Total costs ($)	259,501,674 (238,376,930, 281,142,075)	258,831,495 (238,204,158, 280,149,444)	-538,942 (-2,077,236, 590,738)
ICER ($ per QALY)			10,531 (1,555, 28,537)
SICER ($ per QALY)			-8,008 (-23,450, 12,499)

Abbreviations: MI – Myocardial infarction; YLL – Years of life lived; QALY – Quality-adjusted life years; ICER – Incremental cost-effectiveness ratio; SICER – Incremental cost-effectiveness ratio (societal perspective).

Table 11.2: Probabilistic sensitivity analysis results for Lp(a) testing (intervention) compared to standard of care (control). The UK.

Outcome	Control	Intervention	Difference
Population size	10,000 (10,000, 10,000)	10,000 (10,000, 10,000)	0 (0, 0)
Lp(a) tests	0 (0, 0)	8,993 (8,993, 8,993)	8,993 (8,993, 8,993)
Treatment modified in response to Lp(a) test	0 (0, 0)	624 (624, 624)	624 (624, 624)
Incident MI (N)	706 (589, 852)	678 (563, 824)	-27 (-37, -18)
Total MIs (N)	853 (703, 1,023)	820 (674, 983)	-34 (-49, -22)
Incident stroke (N)	592 (465, 751)	590 (460, 750)	-3 (-12, 6)
Total strokes (N)	753 (595, 946)	746 (588, 941)	-4 (-20, 12)
Deaths (N)	3,862 (3,378, 5,153)	3,854 (3,371, 5,151)	-7 (-17, 2)
YLL no CVD or diabetes	137,535 (135,721, 139,087)	137,799 (135,981, 139,292)	254 (164, 364)
YLL with MI	3,084 (2,586, 3,632)	2,894 (2,425, 3,414)	-187 (-275, -83)
YLL with stroke	1,235 (984, 1,521)	1,221 (977, 1,499)	-9 (-75, 56)
YLL with diabetes	20,868 (20,015, 21,677)	20,894 (20,061, 21,710)	29 (5, 60)
YLL with MI and stroke	123 (74, 192)	118 (72, 185)	-4 (-20, 9)
YLL with MI and diabetes	563 (410, 742)	542 (393, 729)	-19 (-47, 5)
YLL with stroke and diabetes	422 (279, 587)	420 (271, 581)	-1 (-21, 17)
YLL with MI, stroke, and diabetes	34 (12, 60)	33 (12, 59)	0 (-9, 4)
QALY with no CVD or diabetes	111,884 (106,100, 118,364)	112,071 (106,261, 118,525)	203 (132, 294)
QALY with MI	1,945 (1,577, 2,340)	1,833 (1,487, 2,207)	-119 (-178, -55)
QALY with stroke	621 (490, 779)	615 (489, 761)	-5 (-39, 29)
QALY with diabetes	13,008 (10,424, 14,816)	13,023 (10,436, 14,846)	18 (3, 38)
QALY with MI and stroke	62 (38, 99)	60 (36, 96)	-2 (-10, 5)
QALY with MI and diabetes	319 (216, 447)	308 (210, 430)	-10 (-29, 3)
QALY with stroke and diabetes	197 (131, 293)	199 (129, 294)	-0 (-10, 9)
QALY with MI, stroke, and diabetes	15 (5, 30)	15 (6, 28)	0 (-4, 2)
Chronic MI healthcare costs (£)	9,418,297 (7,766,481, 11,192,141)	8,808,704 (7,206,313, 10,459,240)	-578,421 (-859,398, -264,583)
Chronic stroke healthcare costs (£)	8,818,233 (6,896,446, 11,404,862)	8,760,939 (6,859,890, 11,261,068)	-60,489 (-549,027, 393,945)
Chronic diabetes healthcare costs (£)	49,305,728 (46,965,829, 51,603,488)	49,375,522 (47,029,821, 51,695,392)	67,049 (12,057, 138,546)
Chronic MI and stroke healthcare costs (£)	1,633,159 (833,077, 2,771,589)	1,564,393 (801,515, 2,689,112)	-48,116 (-263,396, 122,779)
Chronic MI and diabetes healthcare costs (£)	2,288,034 (1,686,468, 3,073,735)	2,195,982 (1,631,290, 2,992,604)	-77,136 (-214,719, 19,341)
Chronic stroke and diabetes healthcare costs (£)	4,377,424 (2,725,940, 6,411,441)	4,326,765 (2,697,491, 6,422,046)	-4,986 (-213,181, 194,888)
Chronic MI, diabetes, and stroke healthcare costs (£)	435,831 (155,442, 879,406)	421,044 (150,130, 873,435)	0 (-103,880, 51,860)
Acute MI costs (£)	1,032,395 (835,144, 1,256,238)	984,424 (793,896, 1,194,508)	-47,601 (-70,693, -27,067)
Acute stroke costs (£)	1,830,631 (1,452,837, 2,318,541)	1,813,449 (1,439,630, 2,307,700)	-10,323 (-52,209, 30,158)
Medication costs (£)	3,965,725 (3,880,567, 4,041,337)	4,132,035 (4,042,474, 4,208,367)	166,637 (160,410, 172,319)
Lp(a) test costs (£)	0 (0, 0)	359,720 (359,720, 359,720)	359,720 (359,720, 359,720)
Total YLL	163,941 (161,984, 165,659)	164,015 (162,005, 165,743)	65 (3, 138)
Total QALY	128,183 (121,231, 135,455)	128,274 (121,313, 135,530)	85 (37, 138)
Total healthcare costs (£)	83,275,201 (78,367,082, 88,299,668)	83,105,527 (77,948,611, 87,967,064)	-254,903 (-849,164, 272,835)
Total indirect costs (£)	56,924,510 (50,761,619, 65,110,517)	56,281,840 (50,196,962, 64,166,541)	-631,006 (-1,262,144, -170,045)
Total costs (£)	140,571,242 (132,842,238, 149,168,998)	139,529,012 (132,201,594, 148,212,495)	-888,709 (-1,862,705, -28,594)
ICER (£ per QALY)			-2,957 (-10,660, 4,325)
SICER (£ per QALY)			-10,426 (-20,654, -734)

Abbreviations: MI – Myocardial infarction; YLL – Years of life lived; QALY – Quality-adjusted life years; ICER – Incremental cost-effectiveness ratio; SICER – Incremental cost-effectiveness ratio (societal perspective).

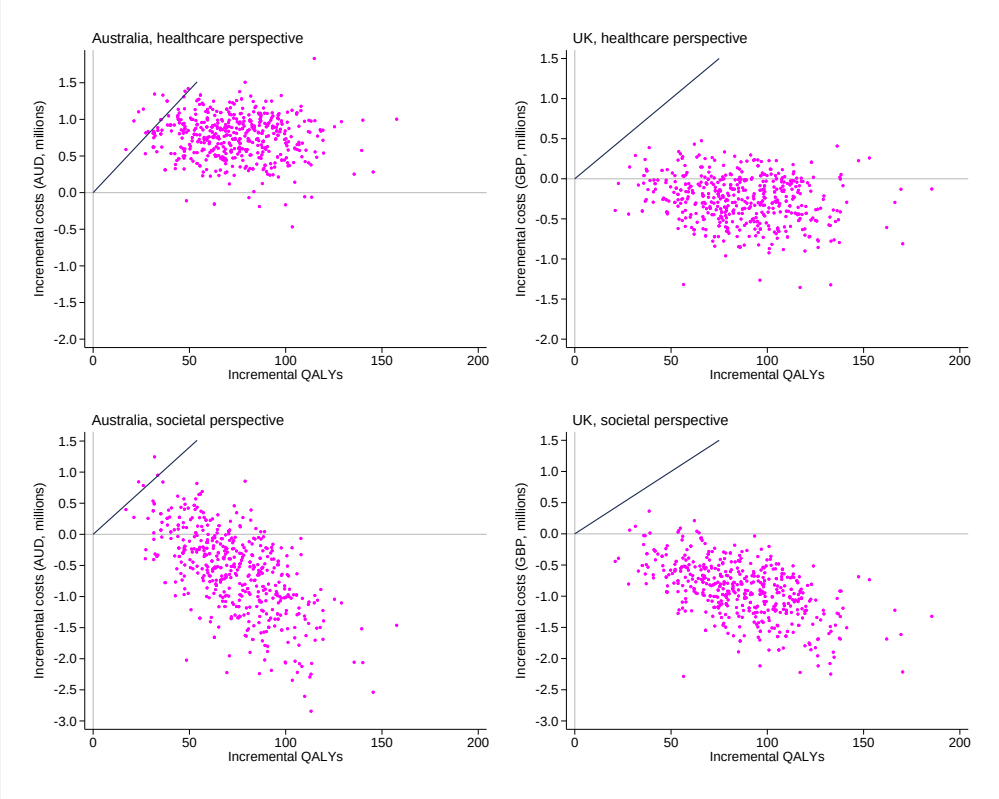


Figure 11.1: Results of the probabilistic sensitivity analysis for Lp(a) testing (intervention) compared to standard of care (control) in a common cost-effectiveness plane for each country.

```
graph combine ///
GPH/PSA_1.gph ///
GPH/PSA_2.gph ///
GPH/PSAS_1.gph ///
GPH/PSAS_2.gph ///
, graphregion(color(white)) cols(2) xsize(5) altshrink
> to standard of care (control) ///
in a common cost-effectiveness plane for each country.)

. use PSA/Res, clear
. keep if njm == 41
(41,000 observations deleted)
. count if c == 1 & diff < 28000
  486
. count if c == 1 & diff < 0
  9
. count if c == 2 & diff < 20000
  500
. count if c == 2 & diff < 0
  407
. use PSA/Res, clear
. keep if njm == 42
(41,000 observations deleted)
. count if c == 1 & diff < 28000
  496
```

```
. count if c == 1 & diff < 0
  410
. count if c == 2 & diff < 20000
  500
. count if c == 2 & diff < 0
  490
```

12 Cost adaptation

We will perform cost adaptation based on the approach of Ademi et al. [160] to present results for the following countries (in addition to Australia and the UK):

- Austria
- France
- Germany
- Italy
- The Netherlands
- Spain
- Poland
- the United States of America (USA).

To do this, I will need the per-capita expenditure on healthcare for each country, the purchasing power parity for each country, and the mean annual income for each country. These data will be sourced from the Organization for Economic Cooperation and Development (OECD) [161, 162, 163]. I will use these data to adjust costs with the UK serving as the reference. The willingness-to-pay threshold for comparison will be drawn from a study estimating cost-effectiveness thresholds for 174 countries by Pichon-Riviere et al. [164], which gave the following estimated cost-effectiveness thresholds per QALY in 2019 USD:

- Austria – $46,380
- Canada – $44,638
- France – $40,006
- Germany – $47,461
- Italy – $46,357
- The Netherlands – $47,122
- Spain – $24,733
- Poland – $8,389
- the United States of America (USA) – $95,958.

Table 12.1: Health spending, purchasing power parity (PPP) ratio, and average annual wages for selected OECD countries (all data in 2022 USD).

Country	Health spending	PPP	Average annual wages
Austria	7,275	0.70	71,340
Canada	6,319	1.16	66,066
France	6,630	0.67	60,086
Germany	8,011	0.69	65,822
Italy	4,291	0.60	50,263
The Netherlands	6,729	0.73	71,776
Spain	4,432	0.58	50,859
Poland	2,973	1.79	40,023
UK	5,493	0.65	57,323
USA	12,555	1.00	79,882

12.1 Study 1

```
quietly {
forval i = 0/1 {
use modend`i´, clear
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
sort ind age
gen LPATT=1 if LPAT==1 & lpa >= 90
keep if cycle!=.
merge m:1 sex age using UTvals_UK, keep(3) nogen
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
```

```
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
```

```
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
gen N = 1 if cycle == 0
forval c = 1/10 {
if `c´ == 1 {
local CC = "UK"
local HS = 5493
local PP = 0.65
local AW = 57323
}
if `c´ == 2 {
local CC = "Austria"
local HS = 7275
local PP = 0.70
local AW = 71340
}
if `c´ == 3 {
local CC = "Canada"
local HS = 6319
local PP = 1.16
local AW = 66066
}
if `c´ == 4 {
local CC = "France"
local HS = 6630
local PP = 0.67
local AW = 60086
}
if `c´ == 5 {
local CC = "Germany"
local HS = 8011
local PP = 0.69
local AW = 65822
}
if `c´ == 6 {
local CC = "Italy"
local HS = 4291
local PP = 0.60
local AW = 50263
}
if `c´ == 7 {
local CC = "The Netherlands"
local HS = 6729
local PP = 0.73
local AW = 71776
}
if `c´ == 8 {
```

```
local CC = "Spain"
local HS = 4432
local PP = 0.58
local AW = 50859
}
if `c´ == 9 {
local CC = "Poland"
local HS = 2973
local PP = 1.79
local AW = 40023
}
if `c´ == 10 {
local CC = "USA"
local HS = 12555
local PP = 1
local AW = 79882
}
foreach var of varlist HEAHS-MSDHSQL {
gen double `var´_DC_`c´ = `var´*DC
}
foreach var of varlist MIOHSHC-LPATHC {
gen double `var´_DC_`c´ = `var´*DC*(`HS´/5493)*(`PP´/0.65)
}
gen double INDC_DC_`c´ = INDC*DC*(`HS´/5493)*(`PP´/0.65)*(`AW´/57323)
}
collapse (sum) N HEAHS_DC_1-INDC_DC_10
forval c = 1/10 {
gen double QLY_`c´ = HEAHSQL_DC_`c´+MIOHSQL_DC_`c´+STOHSQL_DC_`c´+DMOHSQL_DC_`c´+MISHSQL_DC_`c´+MIDH
> SQL_DC_`c´+STDHSQL_DC_`c´+MSDHSQL_DC_`c´
gen double HCC_`c´ = MIOHSHC_DC_`c´+STOHSHC_DC_`c´+DMOHSHC_DC_`c´+MISHSHC_DC_`c´+MIDHSHC_DC_`c´+STDH
> SHC_DC_`c´+MSDHSHC_DC_`c´+ACMIC_DC_`c´+ACSTC_DC_`c´+DRUGSHC_DC_`c´+LPATHC_DC_`c´
gen double IND_`c´ = INDC_DC_`c´
gen double TSC_`c´ = HCC_`c´+INDC_DC_`c´
}
keep QLY_1-TSC_10
expand 40
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY_1-TSC_10 {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabres_`i´_CA, replace
}
}
use tabres_0_CA, clear
merge 1:1 _n using tabres_1_CA, nogen
gen double diff = val1-val0
gen njm = _n
gen country = substr(stat,-1,1)
replace country = "10" if country == "0"
destring country, replace
expand 3 if substr(stat,1,3)=="TSC"
gen njmm = _n
bysort country (njm njmm) : replace val0 = . if _n>=5
bysort country (njm njmm) : replace val1 = . if _n>=5
bysort country (njm njmm) : replace diff = diff[2]/diff[1] if _n==5
bysort country (njm njmm) : replace diff = diff[4]/diff[1] if _n==6
bysort country (njm njmm) : drop if _n == 1
gen row = ""
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Euro)" if _n == 1
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Euro)" if _n == 2
bysort country (njm njmm) : replace row = "Total costs (2023 Euro)" if _n == 3
bysort country (njm njmm) : replace row = "ICER (2023 Euro per QALY)" if _n == 4
```

```
bysort country (njm njmm) : replace row = "SICER (2023 Euro per QALY)" if _n == 5
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 GBP)" if _n == 1 & country =
> = 1
bysort country (njm njmm) : replace row = "Total indirect costs (2023 GBP)" if _n == 2 & country ==
> 1
bysort country (njm njmm) : replace row = "Total costs (2023 GBP)" if _n == 3 & country == 1
bysort country (njm njmm) : replace row = "ICER (2023 GBP per QALY)" if _n == 4 & country == 1
bysort country (njm njmm) : replace row = "SICER (2023 GBP per QALY)" if _n == 5 & country == 1
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 CAD)" if _n == 1 & country =
> = 3
bysort country (njm njmm) : replace row = "Total indirect costs (2023 CAD)" if _n == 2 & country ==
> 3
bysort country (njm njmm) : replace row = "Total costs (2023 CAD)" if _n == 3 & country == 3
bysort country (njm njmm) : replace row = "ICER (2023 CAD per QALY)" if _n == 4 & country == 3
bysort country (njm njmm) : replace row = "SICER (2023 CAD per QALY)" if _n == 5 & country == 3
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Zloty)" if _n == 1 & country
>  == 9
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Zloty)" if _n == 2 & country =
> = 9
bysort country (njm njmm) : replace row = "Total costs (2023 Zloty)" if _n == 3 & country == 9
bysort country (njm njmm) : replace row = "ICER (2023 Zloty per QALY)" if _n == 4 & country == 9
bysort country (njm njmm) : replace row = "SICER (2023 Zloty per QALY)" if _n == 5 & country == 9
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 USD)" if _n == 1 & country =
> = 10
bysort country (njm njmm) : replace row = "Total indirect costs (2023 USD)" if _n == 2 & country ==
> 10
bysort country (njm njmm) : replace row = "Total costs (2023 USD)" if _n == 3 & country == 10
bysort country (njm njmm) : replace row = "ICER (2023 USD per QALY)" if _n == 4 & country == 10
bysort country (njm njmm) : replace row = "SICER (2023 USD per QALY)" if _n == 5 & country == 10
gen C = ""
bysort country (njm njmm) : replace C = "UK" if _n == 1 & country == 1
bysort country (njm njmm) : replace C = "Austria" if _n == 1 & country == 2
bysort country (njm njmm) : replace C = "Canada" if _n == 1 & country == 3
bysort country (njm njmm) : replace C = "France" if _n == 1 & country == 4
bysort country (njm njmm) : replace C = "Germany" if _n == 1 & country == 5
bysort country (njm njmm) : replace C = "Italy" if _n == 1 & country == 6
bysort country (njm njmm) : replace C = "The Netherlands" if _n == 1 & country == 7
bysort country (njm njmm) : replace C = "Spain" if _n == 1 & country == 8
bysort country (njm njmm) : replace C = "Poland" if _n == 1 & country == 9
bysort country (njm njmm) : replace C = "USA" if _n == 1 & country == 10
order C row val0 val1 diff
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
drop stat-njmm
drop if _n <= 5
export delimited using CSV/CAres.csv, delimiter(":") novarnames replace
```

Table 12.2: Cost adaptation results. Study 1.

Country	Outcome	Control	Intervention	Difference
Austria	Total healthcare costs (2023 Euro)	116,440,213	115,971,237	-468,976
	Total indirect costs (2023 Euro)	98,841,798	97,532,239	-1,309,559
	Total costs (2023 Euro)	215,282,011	213,503,476	-1,778,535
	ICER (2023 Euro per QALY)			-4,787
	SICER (2023 Euro per QALY)			-18,153
Canada	Total healthcare costs (2023 CAD)	167,601,653	166,926,618	-675,034
	Total indirect costs (2023 CAD)	131,753,103	130,007,501	-1,745,602
	Total costs (2023 CAD)	299,354,756	296,934,119	-2,420,636
	ICER (2023 CAD per QALY)			-6,890
	SICER (2023 CAD per QALY)			-24,707
France	Total healthcare costs (2023 Euro)	101,568,791	101,159,711	-409,080
	Total indirect costs (2023 Euro)	72,616,970	71,654,864	-962,105
	Total costs (2023 Euro)	174,185,761	172,814,576	-1,371,185
	ICER (2023 Euro per QALY)			-4,175
	SICER (2023 Euro per QALY)			-13,995
Germany	Total healthcare costs (2023 Euro)	126,388,562	125,879,519	-509,044
	Total indirect costs (2023 Euro)	98,988,192	97,676,694	-1,311,499
	Total costs (2023 Euro)	225,376,755	223,556,212	-1,820,543
	ICER (2023 Euro per QALY)			-5,196
	SICER (2023 Euro per QALY)			-18,582
Italy	Total healthcare costs (2023 Euro)	58,868,330	58,631,231	-237,099
	Total indirect costs (2023 Euro)	35,207,458	34,740,993	-466,465
	Total costs (2023 Euro)	94,075,788	93,372,224	-703,564
	ICER (2023 Euro per QALY)			-2,420
	SICER (2023 Euro per QALY)			-7,181
The Netherlands	Total healthcare costs (2023 Euro)	112,316,960	111,864,591	-452,369
	Total indirect costs (2023 Euro)	95,924,410	94,653,503	-1,270,907
	Total costs (2023 Euro)	208,241,370	206,518,095	-1,723,275
	ICER (2023 Euro per QALY)			-4,617
	SICER (2023 Euro per QALY)			-17,589
Spain	Total healthcare costs (2023 Euro)	58,775,955	58,539,229	-236,727
	Total indirect costs (2023 Euro)	35,569,033	35,097,777	-471,256
	Total costs (2023 Euro)	94,344,988	93,637,006	-707,982
	ICER (2023 Euro per QALY)			-2,416
	SICER (2023 Euro per QALY)			-7,226
Poland	Total healthcare costs (2023 Zloty)	121,680,194	121,190,113	-490,080
	Total indirect costs (2023 Zloty)	57,947,416	57,179,668	-767,748
	Total costs (2023 Zloty)	179,627,610	178,369,782	-1,257,828
	ICER (2023 Zloty per QALY)			-5,002
	SICER (2023 Zloty per QALY)			-12,838
USA	Total healthcare costs (2023 USD)	287,070,568	285,914,360	-1,156,208
	Total indirect costs (2023 USD)	272,861,425	269,246,273	-3,615,152
	Total costs (2023 USD)	559,931,993	555,160,633	-4,771,361
	ICER (2023 USD per QALY)			-11,801
	SICER (2023 USD per QALY)			-48,701

Abbreviations: QALY – Quality-adjusted life years; ICER – Incremental cost-effectiveness ratio; SICER – Incremental cost-effectiveness ratio (societal perspective).

12.2 Study 2

```
quietly {
forval ii = 0/1 {
use modend2_`ii´, clear
bysort ind MIE (age) : gen MI1 = 1 if _n == 1 & MIE == 1
bysort ind STE (age) : gen ST1 = 1 if _n == 1 & STE == 1
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
merge m:1 sex age using UTvals_UK, keep(3) nogen
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.
gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
```

```
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+18.00 if LLT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC+15.91+12.42+9.91 if AHT == 1 & MI == 0 & ST == 0
replace DRUGSHC = DRUGSHC/2 if MIE == 1 | STE == 1 | DTE == 1
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)
gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
```

```
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
gen N = 1 if cycle == 0

forval c = 1/9 {
if `c´ == 1 {
local CC = "UK"
local HS = 5493
local PP = 0.65
local AW = 57323
}
if `c´ == 2 {
local CC = "Austria"
local HS = 7275
local PP = 0.70
local AW = 71340
}
if `c´ == 3 {
local CC = "France"
local HS = 6630
local PP = 0.67
local AW = 60086
}
if `c´ == 4 {
local CC = "Germany"
local HS = 8011
local PP = 0.69
local AW = 65822
}
if `c´ == 5 {
local CC = "Italy"
local HS = 4291
local PP = 0.60
local AW = 50263
}
if `c´ == 6 {
local CC = "The Netherlands"
local HS = 6729
local PP = 0.73
local AW = 71776
}
if `c´ == 7 {
local CC = "Spain"
local HS = 4432
local PP = 0.58
local AW = 50859
}
```

```
if `c´ == 8 {
local CC = "Poland"
local HS = 2973
local PP = 1.79
local AW = 40023
}
if `c´ == 9 {
local CC = "USA"
local HS = 12555
local PP = 1
local AW = 79882
}
foreach var of varlist HEAHS-MSDHSQL {
gen double `var´_DC_`c´ = `var´*DC
}
foreach var of varlist MIOHSHC-LPATHC {
gen double `var´_DC_`c´ = `var´*DC*(`HS´/5493)*(`PP´/0.65)
}
gen double INDC_DC_`c´ = INDC*DC*(`HS´/5493)*(`PP´/0.65)*(`AW´/57323)
}
collapse (sum) N HEAHS_DC_1-INDC_DC_9
forval c = 1/9 {
gen double QLY_`c´ = HEAHSQL_DC_`c´+MIOHSQL_DC_`c´+STOHSQL_DC_`c´+DMOHSQL_DC_`c´+MISHSQL_DC_`c´+MIDH
> SQL_DC_`c´+STDHSQL_DC_`c´+MSDHSQL_DC_`c´
gen double HCC_`c´ = MIOHSHC_DC_`c´+STOHSHC_DC_`c´+DMOHSHC_DC_`c´+MISHSHC_DC_`c´+MIDHSHC_DC_`c´+STDH
> SHC_DC_`c´+MSDHSHC_DC_`c´+ACMIC_DC_`c´+ACSTC_DC_`c´+DRUGSHC_DC_`c´+LPATHC_DC_`c´
gen double IND_`c´ = INDC_DC_`c´
gen double TSC_`c´ = HCC_`c´+INDC_DC_`c´
}
keep QLY_1-TSC_9
expand 32
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY_1-TSC_9 {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`ii´
save tabres_`ii´_CA_2, replace
}
}
use tabres_0_CA_2, clear
merge 1:1 _n using tabres_1_CA_2, nogen
gen double diff = val1-val0
gen njm = _n
gen country = substr(stat,-1,1)
expand 3 if substr(stat,1,3)=="TSC"
gen njmm = _n
bysort country (njm njmm) : replace val0 = . if _n>=5
bysort country (njm njmm) : replace val1 = . if _n>=5
bysort country (njm njmm) : replace diff = diff[2]/diff[1] if _n==5
bysort country (njm njmm) : replace diff = diff[4]/diff[1] if _n==6
bysort country (njm njmm) : drop if _n == 1
gen row = ""
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Euro)" if _n == 1
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Euro)" if _n == 2
bysort country (njm njmm) : replace row = "Total costs (2023 Euro)" if _n == 3
bysort country (njm njmm) : replace row = "ICER (2023 Euro per QALY)" if _n == 4
bysort country (njm njmm) : replace row = "SICER (2023 Euro per QALY)" if _n == 5
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 GBP)" if _n == 1 & country =
> = "1"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 GBP)" if _n == 2 & country ==
> "1"
bysort country (njm njmm) : replace row = "ICER (2023 GBP per QALY)" if _n == 4 & country == "1"
bysort country (njm njmm) : replace row = "SICER (2023 GBP per QALY)" if _n == 5 & country == "1"
```

```
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Zloty)" if _n == 1 & country
>  == "8"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Zloty)" if _n == 2 & country =
> = "8"
bysort country (njm njmm) : replace row = "Total costs (2023 Zloty)" if _n == 3 & country == "8"
bysort country (njm njmm) : replace row = "ICER (2023 Zloty per QALY)" if _n == 4 & country == "8"
bysort country (njm njmm) : replace row = "SICER (2023 Zloty per QALY)" if _n == 5 & country == "8"
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 USD)" if _n == 1 & country =
> = "9"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 USD)" if _n == 2 & country ==
> "9"
bysort country (njm njmm) : replace row = "Total costs (2023 USD)" if _n == 3 & country == "9"
bysort country (njm njmm) : replace row = "ICER (2023 USD per QALY)" if _n == 4 & country == "9"
bysort country (njm njmm) : replace row = "SICER (2023 USD per QALY)" if _n == 5 & country == "9"
gen C = ""
bysort country (njm njmm) : replace C = "UK" if _n == 1 & country == "1"
bysort country (njm njmm) : replace C = "Austria" if _n == 1 & country == "2"
bysort country (njm njmm) : replace C = "France" if _n == 1 & country == "3"
bysort country (njm njmm) : replace C = "Germany" if _n == 1 & country == "4"
bysort country (njm njmm) : replace C = "Italy" if _n == 1 & country == "5"
bysort country (njm njmm) : replace C = "The Netherlands" if _n == 1 & country == "6"
bysort country (njm njmm) : replace C = "Spain" if _n == 1 & country == "7"
bysort country (njm njmm) : replace C = "Poland" if _n == 1 & country == "8"
bysort country (njm njmm) : replace C = "USA" if _n == 1 & country == "9"
order C row val0 val1 diff
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
drop stat-njmm
export delimited using CSV/CAres2.csv, delimiter(":") novarnames replace

quietly {
forval i = 0/1 {
use modendsp`i´, clear
bysort ind LPAT (age) : replace LPAT=. if _n!=1
gen LPATT=1 if LPAT==1 & LPT == 1
keep if cycle!=.
merge m:1 sex age using UTvals_UK, keep(3) nogen
sort ind age
gen double HEAHS = .
gen double MIOHS = .
gen double STOHS = .
gen double DMOHS = .
gen double MISHS = .
gen double MIDHS = .
gen double STDHS = .
gen double MSDHS = .
replace HEAHS = 1 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==. & STE==. & DME==. & DTE==.
replace HEAHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & (MIE==1 | STE==1 | DME==1 | DTE==1)
replace MIOHS = 1 if MI==1 & ST==0 & DM==0 & DT==0 & STE==. & DME==. & DTE==.
replace MIOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==.
replace MIOHS = 0.5 if MI==1 & ST==0 & DM==0 & DT==0 & (STE==1 | DME==1 | DTE==1)
replace STOHS = 1 if MI==0 & ST==1 & DM==0 & DT==0 & MIE==. & DME==. & DTE==.
replace STOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.
replace STOHS = 0.5 if MI==0 & ST==1 & DM==0 & DT==0 & (MIE==1 | DME==1 | DTE==1)
replace DMOHS = 1 if MI==0 & ST==0 & DM==1 & DT==0 & MIE==. & STE==. & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.
replace DMOHS = 0.5 if MI==0 & ST==0 & DM==1 & DT==0 & (MIE==1 | STE==1 | DTE==1)
replace MISHS = 1 if MI==1 & ST==1 & DM==0 & DT==0 & DME==. & DTE==.
replace MISHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE==.) | (MI==0 & ST==1 & DM==0 &
> DT==0 & MIE==1 & DTE==.)
replace MISHS = 0.5 if MI==1 & ST==1 & DM==0 & DT==0 & (DME==1 | DTE==1)
replace MIDHS = 1 if MI==1 & ST==0 & DM==1 & DT==0 & STE==. & DTE==.
replace MIDHS = 0.5 if (MI==1 & ST==0 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & MIE==1 & DTE==.)
replace MIDHS = 0.5 if MI==1 & ST==0 & DM==1 & DT==0 & (STE==1 | DTE==1)
replace STDHS = 1 if MI==0 & ST==1 & DM==1 & DT==0 & MIE==. & DTE==.
replace STDHS = 0.5 if (MI==0 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | (MI==0 & ST==0 & DM==1 &
> DT==0 & STE==1 & DTE==.)
```

```
replace STDHS = 0.5 if MI==0 & ST==1 & DM==1 & DT==0 & (MIE==1 | DTE==1)
replace MSDHS = 1 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==.
replace MSDHS = 0.5 if ///
(MI==1 & ST==1 & DM==0 & DT==0 & DME==1 & DTE==.) | ///
(MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE==.) | ///
(MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==.)
replace MSDHS = 0.5 if MI==1 & ST==1 & DM==1 & DT==0 & DTE==1
gen double HEAHSQL = HEAHS*UT
gen double MIOHSQL = MIOHS*UT*0.79
gen double STOHSQL = STOHS*UT*0.65
gen double DMOHSQL = DMOHS*UT*0.785
gen double MISHSQL = MISHS*UT*0.65
gen double MIDHSQL = MIDHS*UT*(0.785-0.055)
gen double STDHSQL = STDHS*UT*(0.785-0.164)
gen double MSDHSQL = MSDHS*UT*(0.785-0.164)
replace MIOHSQL = MIOHSQL-0.01125 if MIE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.01125 if MIE == 1 & DTE==.
replace MIDHSQL = MIDHSQL-0.01125 if MIE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.01125 if MIE == 1 & DTE==.
replace STOHSQL = STOHSQL-0.03 if STE == 1 & DTE==.
replace MISHSQL = MISHSQL-0.03 if STE == 1 & DTE==.
replace STDHSQL = STDHSQL-0.03 if STE == 1 & DTE==.
replace MSDHSQL = MSDHSQL-0.03 if STE == 1 & DTE==.

gen double MIOHSHC = MIOHS*3304 if sex == 0
gen double STOHSHC = STOHS*7021 if sex == 0
gen double DMOHSHC = DMOHS*2546 if sex == 0
gen double MISHSHC =MISHS*14442 if sex == 0
gen double MIDHSHC = MIDHS*4511 if sex == 0
gen double STDHSHC = STDHS*10014 if sex == 0
gen double MSDHSHC =MSDHS*14442 if sex == 0
replace MIOHSHC = MIOHS*2917 if sex == 1
replace STOHSHC = STOHS*7351 if sex == 1
replace DMOHSHC = DMOHS*2170 if sex == 1
replace MISHSHC =MISHS*12616 if sex == 1
replace MIDHSHC = MIDHS*3917 if sex == 1
replace STDHSHC = STDHS*10494 if sex == 1
replace MSDHSHC =MSDHS*12616 if sex == 1
gen double ACMIC = 2212 if MIE == 1 & DTE == .
replace ACMIC = 515 if MIE == 1 & DTE == 1
gen double ACSTC = 4626 if STE == 1 & DTE ==.
replace ACSTC = 3886 if STE == 1 & DTE == 1
gen double DRUGSHC = 0
replace DRUGSHC = DRUGSHC+3975 if LPT == 1
gen LPATHC = 40 if LPAT==1
gen WFP_GP = .
replace WFP_GP = 0.806 if sex == 0 & inrange(age,40,49)
replace WFP_GP = 0.659 if sex == 0 & inrange(age,50,65)
replace WFP_GP = 0.900 if sex == 1 & inrange(age,40,49)
replace WFP_GP = 0.743 if sex == 1 & inrange(age,50,65)


gen HEAHS_WFP = WFP_GP
gen MIOHS_WFP = 1-(1.46*(1-WFP_GP))
gen STOHS_WFP = 1-(1.92*(1-WFP_GP))
gen DMOHS_WFP = WFP_GP-0.037 if  sex == 0 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.002 if sex == 0 & inrange(age,50,65)
replace DMOHS_WFP = WFP_GP-0.039 if sex == 1 & inrange(age,40,49)
replace DMOHS_WFP = WFP_GP-0.115 if sex == 1 & inrange(age,50,65)
gen MISHS_WFP = 1-(1.92*(1-WFP_GP))
gen MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MIDHS_WFP = 1-(1.46*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
gen STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace STDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
```

```
gen MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.037 if sex == 0 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.002 if sex == 0 & inrange(age,50,65)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.039 if sex == 1 & inrange(age,40,49)
replace MSDHS_WFP = 1-(1.92*(1-WFP_GP))-0.115 if sex == 1 & inrange(age,50,65)
replace MIOHS_WFP= MIOHS_WFP*((365.25-21.5)/365.25)
replace STOHS_WFP= STOHS_WFP*((365.25-21.5)/365.25)
replace DMOHS_WFP= DMOHS_WFP*((365.25-6)/365.25)
replace MISHS_WFP= MISHS_WFP*((365.25-21.5)/365.25)
replace MIDHS_WFP= MIDHS_WFP*((365.25-21.5-6)/365.25)
replace STDHS_WFP= STDHS_WFP*((365.25-21.5-6)/365.25)
replace MSDHS_WFP= MSDHS_WFP*((365.25-21.5-6)/365.25)
replace MIOHS_WFP = MIOHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace STOHS_WFP = STOHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==0 & DT==0 & STE==1 & DTE==
> .
replace MIDHS_WFP = MIDHS_WFP*((365.25-55)/365.25) if MI==1 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace STDHS_WFP = STDHS_WFP*((365.25-90)/365.25) if MI==0 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MISHS_WFP = MISHS_WFP*((365.25-55)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE==
> .
replace MSDHS_WFP = MSDHS_WFP*((365.25-90)/365.25) if MI==1 & ST==1 & DM==1 & DT==0 & STE==1 & DTE==
> .
replace MIOHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace STOHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==0 & DT==0 & MIE==1 & DTE
> ==.
replace MISHS_WFP = MIOHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==0 & DT==0 & STE==1 & DTE
> ==.
replace MIDHS_WFP = MIDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace STDHS_WFP = STDHS_WFP*((182.625-90)/182.625) if MI==0 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-55)/182.625) if MI==0 & ST==1 & DM==1 & DT==0 & MIE==1 & DTE
> ==.
replace MSDHS_WFP = MSDHS_WFP*((182.625-90)/182.625) if MI==1 & ST==0 & DM==1 & DT==0 & STE==1 & DTE
> ==.

gen WFP = 0 if age < 66
foreach var of varlist HEAHS-MSDHS {
replace WFP = WFP+(`var´*`var´_WFP) if `var´!=.
}
gen double INDC = (WFP_GP-WFP)*34855
gen DC = 1/((1.035)^(cycle))
gen N = 1 if cycle == 0
forval c = 1/9 {
if `c´ == 1 {
local CC = "UK"
local HS = 5493
local PP = 0.65
local AW = 57323
}
if `c´ == 2 {
local CC = "Austria"
local HS = 7275
local PP = 0.70
local AW = 71340
}
if `c´ == 3 {
local CC = "France"
local HS = 6630
local PP = 0.67
```

```
local AW = 60086
}
if `c´ == 4 {
local CC = "Germany"
local HS = 8011
local PP = 0.69
local AW = 65822
}
if `c´ == 5 {
local CC = "Italy"
local HS = 4291
local PP = 0.60
local AW = 50263
}
if `c´ == 6 {
local CC = "The Netherlands"
local HS = 6729
local PP = 0.73
local AW = 71776
}
if `c´ == 7 {
local CC = "Spain"
local HS = 4432
local PP = 0.58
local AW = 50859
}
if `c´ == 8 {
local CC = "Poland"
local HS = 2973
local PP = 1.79
local AW = 40023
}
if `c´ == 9 {
local CC = "USA"
local HS = 12555
local PP = 1
local AW = 79882
}
foreach var of varlist HEAHS-MSDHSQL {
gen double `var´_DC_`c´ = `var´*DC
}
foreach var of varlist MIOHSHC-LPATHC {
gen double `var´_DC_`c´ = `var´*DC*(`HS´/5493)*(`PP´/0.65)
}
gen double INDC_DC_`c´ = INDC*DC*(`HS´/5493)*(`PP´/0.65)*(`AW´/57323)
}
collapse (sum) N HEAHS_DC_1-INDC_DC_9
forval c = 1/9 {
gen double QLY_`c´ = HEAHSQL_DC_`c´+MIOHSQL_DC_`c´+STOHSQL_DC_`c´+DMOHSQL_DC_`c´+MISHSQL_DC_`c´+MIDH
> SQL_DC_`c´+STDHSQL_DC_`c´+MSDHSQL_DC_`c´
gen double HCC_`c´ = MIOHSHC_DC_`c´+STOHSHC_DC_`c´+DMOHSHC_DC_`c´+MISHSHC_DC_`c´+MIDHSHC_DC_`c´+STDH
> SHC_DC_`c´+MSDHSHC_DC_`c´+ACMIC_DC_`c´+ACSTC_DC_`c´+DRUGSHC_DC_`c´+LPATHC_DC_`c´
gen double IND_`c´ = INDC_DC_`c´
gen double TSC_`c´ = HCC_`c´+INDC_DC_`c´
}
keep QLY_1-TSC_9
expand 32
gen stat = ""
gen double val =.
local j = 1
foreach var of varlist QLY_1-TSC_9 {
replace stat = "`var´" if _n == `j´
replace val = `var´[1] if _n == `j´
local j = `j´+1
}
keep stat val
rename val val`i´
save tabres_`i´_CA_2sp, replace
```

```
}
}
use tabres_0_CA_2sp, clear
merge 1:1 _n using tabres_1_CA_2sp, nogen
gen double diff = val1-val0
gen njm = _n
gen country = substr(stat,-1,1)
expand 3 if substr(stat,1,3)=="TSC"
gen njmm = _n
bysort country (njm njmm) : replace val0 = . if _n>=5
bysort country (njm njmm) : replace val1 = . if _n>=5
bysort country (njm njmm) : replace diff = diff[2]/diff[1] if _n==5
bysort country (njm njmm) : replace diff = diff[4]/diff[1] if _n==6
bysort country (njm njmm) : drop if _n == 1
gen row = ""
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Euro)" if _n == 1
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Euro)" if _n == 2
bysort country (njm njmm) : replace row = "Total costs (2023 Euro)" if _n == 3
bysort country (njm njmm) : replace row = "ICER (2023 Euro per QALY)" if _n == 4
bysort country (njm njmm) : replace row = "SICER (2023 Euro per QALY)" if _n == 5
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 GBP)" if _n == 1 & country =
> = "1"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 GBP)" if _n == 2 & country ==
> "1"
bysort country (njm njmm) : replace row = "ICER (2023 GBP per QALY)" if _n == 4 & country == "1"
bysort country (njm njmm) : replace row = "SICER (2023 GBP per QALY)" if _n == 5 & country == "1"
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 Zloty)" if _n == 1 & country
>  == "8"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 Zloty)" if _n == 2 & country =
> = "8"
bysort country (njm njmm) : replace row = "Total costs (2023 Zloty)" if _n == 3 & country == "8"
bysort country (njm njmm) : replace row = "ICER (2023 Zloty per QALY)" if _n == 4 & country == "8"
bysort country (njm njmm) : replace row = "SICER (2023 Zloty per QALY)" if _n == 5 & country == "8"
bysort country (njm njmm) : replace row = "Total healthcare costs (2023 USD)" if _n == 1 & country =
> = "9"
bysort country (njm njmm) : replace row = "Total indirect costs (2023 USD)" if _n == 2 & country ==
> "9"
bysort country (njm njmm) : replace row = "Total costs (2023 USD)" if _n == 3 & country == "9"
bysort country (njm njmm) : replace row = "ICER (2023 USD per QALY)" if _n == 4 & country == "9"
bysort country (njm njmm) : replace row = "SICER (2023 USD per QALY)" if _n == 5 & country == "9"
gen C = ""
bysort country (njm njmm) : replace C = "UK" if _n == 1 & country == "1"
bysort country (njm njmm) : replace C = "Austria" if _n == 1 & country == "2"
bysort country (njm njmm) : replace C = "France" if _n == 1 & country == "3"
bysort country (njm njmm) : replace C = "Germany" if _n == 1 & country == "4"
bysort country (njm njmm) : replace C = "Italy" if _n == 1 & country == "5"
bysort country (njm njmm) : replace C = "The Netherlands" if _n == 1 & country == "6"
bysort country (njm njmm) : replace C = "Spain" if _n == 1 & country == "7"
bysort country (njm njmm) : replace C = "Poland" if _n == 1 & country == "8"
bysort country (njm njmm) : replace C = "USA" if _n == 1 & country == "9"
order C row val0 val1 diff
tostring val0-diff, replace force format(%15.0fc)
replace val0 = "" if val0 == "."
replace val1 = "" if val1 == "."
drop stat-njmm
export delimited using CSV/CAres2sp.csv, delimiter(":") novarnames replace
```

Table 12.3: Cost adaptation results. Study 2. Primary prevention population.

Country	Outcome	Control	Intervention	Difference
UK	Total healthcare costs (2023 GBP)	83,493,274	106,653,325	23,160,051
	Total indirect costs (2023 GBP)	57,084,779	56,925,554	-159,225
	Total costs (2023 Euro)	140,578,053	163,578,879	23,000,826
	ICER (2023 GBP per QALY)			422,978
	SICER (2023 GBP per QALY)			420,070
Austria	Total healthcare costs (2023 Euro)	119,085,689	152,118,657	33,032,968
	Total indirect costs (2023 Euro)	101,328,730	101,046,097	-282,633
	Total costs (2023 Euro)	220,414,419	253,164,754	32,750,335
	ICER (2023 Euro per QALY)			603,289
	SICER (2023 Euro per QALY)			598,127
France	Total healthcare costs (2023 Euro)	103,876,394	132,690,483	28,814,089
	Total indirect costs (2023 Euro)	74,444,065	74,236,421	-207,644
	Total costs (2023 Euro)	178,320,460	206,926,904	28,606,444
	ICER (2023 Euro per QALY)			526,239
	SICER (2023 Euro per QALY)			522,446
Germany	Total healthcare costs (2023 Euro)	129,260,061	165,115,280	35,855,219
	Total indirect costs (2023 Euro)	101,478,807	101,195,756	-283,052
	Total costs (2023 Euro)	230,738,869	266,311,036	35,572,167
	ICER (2023 Euro per QALY)			654,832
	SICER (2023 Euro per QALY)			649,663
Italy	Total healthcare costs (2023 Euro)	60,205,796	76,906,175	16,700,379
	Total indirect costs (2023 Euro)	36,093,303	35,992,629	-100,674
	Total costs (2023 Euro)	96,299,099	112,898,804	16,599,705
	ICER (2023 Euro per QALY)			305,003
	SICER (2023 Euro per QALY)			303,164
The Netherlands	Total healthcare costs (2023 Euro)	114,868,758	146,731,998	31,863,241
	Total indirect costs (2023 Euro)	98,337,938	98,063,647	-274,291
	Total costs (2023 Euro)	213,206,696	244,795,646	31,588,950
	ICER (2023 Euro per QALY)			581,926
	SICER (2023 Euro per QALY)			576,917
Spain	Total healthcare costs (2023 Euro)	60,111,322	76,785,495	16,674,173
	Total indirect costs (2023 Euro)	36,463,976	36,362,268	-101,708
	Total costs (2023 Euro)	96,575,298	113,147,763	16,572,465
	ICER (2023 Euro per QALY)			304,524
	SICER (2023 Euro per QALY)			302,667
Poland	Total healthcare costs (2023 Zloty)	124,444,720	158,964,220	34,519,500
	Total indirect costs (2023 Zloty)	59,405,415	59,239,718	-165,698
	Total costs (2023 Zloty)	183,850,136	218,203,938	34,353,802
	ICER (2023 Zloty per QALY)			630,438
	SICER (2023 Zloty per QALY)			627,412

Abbreviations: QALY – Quality-adjusted life years; ICER – Incremental cost-effectiveness ratio; SICER – Incremental cost-effectiveness ratio (societal perspective).

Table 12.4: Cost adaptation results. Study 2. Secondary prevention population.

Country	Outcome	Control	Intervention	Difference
UK	Total healthcare costs (2023 GBP)	680,327,985	686,593,168	6,265,183
	Total indirect costs (2023 GBP)	416,096,120	414,397,455	-1,698,665
	Total costs (2023 Euro)	1,096,424,105	1,100,990,623	4,566,518
	ICER (2023 GBP per QALY)			23,056
	SICER (2023 GBP per QALY)			16,805
Austria	Total healthcare costs (2023 Euro)	970,345,548	979,281,521	8,935,973
	Total indirect costs (2023 Euro)	738,594,284	735,579,057	-3,015,227
	Total costs (2023 Euro)	1,708,939,832	1,714,860,578	5,920,746
	ICER (2023 Euro per QALY)			32,885
	SICER (2023 Euro per QALY)			21,789
France	Total healthcare costs (2023 Euro)	846,415,702	854,210,397	7,794,695
	Total indirect costs (2023 Euro)	542,629,530	540,414,307	-2,215,223
	Total costs (2023 Euro)	1,389,045,231	1,394,624,703	5,579,472
	ICER (2023 Euro per QALY)			28,685
	SICER (2023 Euro per QALY)			20,533
Germany	Total healthcare costs (2023 Euro)	1,053,249,357	1,062,948,796	9,699,439
	Total indirect costs (2023 Euro)	739,688,210	736,668,517	-3,019,693
	Total costs (2023 Euro)	1,792,937,566	1,799,617,313	6,679,747
	ICER (2023 Euro per QALY)			35,694
	SICER (2023 Euro per QALY)			24,582
Italy	Total healthcare costs (2023 Euro)	490,574,698	495,092,431	4,517,733
	Total indirect costs (2023 Euro)	263,087,353	262,013,329	-1,074,024
	Total costs (2023 Euro)	753,662,050	757,105,759	3,443,709
	ICER (2023 Euro per QALY)			16,625
	SICER (2023 Euro per QALY)			12,673
The Netherlands	Total healthcare costs (2023 Euro)	935,984,741	944,604,283	8,619,542
	Total indirect costs (2023 Euro)	716,794,130	713,867,900	-2,926,230
	Total costs (2023 Euro)	1,652,778,870	1,658,472,182	5,693,312
	ICER (2023 Euro per QALY)			31,720
	SICER (2023 Euro per QALY)			20,952
Spain	Total healthcare costs (2023 Euro)	489,804,900	494,315,544	4,510,644
	Total indirect costs (2023 Euro)	265,789,219	264,704,165	-1,085,054
	Total costs (2023 Euro)	755,594,119	759,019,709	3,425,590
	ICER (2023 Euro per QALY)			16,599
	SICER (2023 Euro per QALY)			12,606
Poland	Total healthcare costs (2023 Zloty)	1,014,012,527	1,023,350,632	9,338,105
	Total indirect costs (2023 Zloty)	433,011,448	431,243,729	-1,767,720
	Total costs (2023 Zloty)	1,447,023,975	1,454,594,360	7,570,385
	ICER (2023 Zloty per QALY)			34,365
	SICER (2023 Zloty per QALY)			27,859

Abbreviations: QALY – Quality-adjusted life years; ICER – Incremental cost-effectiveness ratio; SICER – Incremental cost-effectiveness ratio (societal perspective).

References

1. Claudia Lamina, Florian Kronenberg, et al. Estimation of the required lipoprotein (a)-lowering therapeutic effect size for reduction in coronary heart disease outcomes: a Mendelian randomization analysis. *JAMA cardiology*, 4(6):575–579, 2019.

2. Susanna C Larsson, Dipender Gill, Amy M Mason, Tao Jiang, Magnus Bäck, Adam S Butterworth, and Stephen Burgess. Lipoprotein (a) in Alzheimer, atherosclerotic, cerebrovascular, thrombotic, and valvular disease: Mendelian randomization investigation. *Circulation*, 141(22):1826–1828, 2020.

3. Paul Welsh, Claire Welsh, Carlos A Celis-Morales, Rosemary Brown, Frederick K Ho, Lyn D Ferguson, Patrick B Mark, James Lewsey, Stuart R Gray, Donald M Lyall, et al. Lipoprotein (a) and cardiovascular disease: prediction, attributable risk fraction, and estimating benefits from novel interventions. *European Journal of Preventive Cardiology*, 28(18):1991–2000, 2021.

4. Børge G Nordestgaard, M John Chapman, Kausik Ray, Jan Borén, Felicita Andreotti, Gerald F Watts, Henry Ginsberg, Pierre Amarenco, Alberico Catapano, Olivier S Descamps, et al. Lipoprotein (a) as a cardiovascular risk factor: current status. *European heart journal*, 31(23):2844–2853, 2010.

5. Michelle L O'Donoghue, Robert S Rosenson, Baris Gencer, J Antonio G López, Norman E Lepor, Seth J Baum, Elmer Stout, Daniel Gaudet, Beat Knusel, Julia F Kuder, et al. Small interfering RNA to reduce lipoprotein (a) in cardiovascular disease. *New England Journal of Medicine*, 387(20):1855–1864, 2022.

6. Steven E Nissen, Helle Linnebjerg, Xi Shen, Kathy Wolski, Xiaosu Ma, Shufen Lim, Laura F Michael, Giacomo Ruotolo, Grace Gribble, Ann Marie Navar, et al. Lepodisiran, an extended-duration short interfering RNA targeting lipoprotein (a): a randomized dose-ascending clinical trial. *JAMA*, 330(21):2075–2083, 2023.

7. Xing L Wang, David EL Wilcken, and Nicholas PB Dudman. Early expression of the apolipoprotein (a) gene: relationships between infants' and their parents' serum apolipoprotein (a) levels. *Pediatrics*, 89(3):401–406, 1992.

8. Nader Rifai, Gerardo Heiss, and Karl Doetsch. Lipoprotein (a) at birth, in blacks and whites. *Atherosclerosis*, 92(2-3):123–129, 1992.

9. Catherine J McNeal. Lipoprotein (a): its relevance to the pediatric population. *Journal of Clinical Lipidology*, 9(5):S57–S66, 2015.

10. Eric Boerwinkle, Carla C Leffert, Jingping Lin, Carolin Lackner, Giulia Chiesa, Helen H Hobbs, et al. Apolipoprotein (a) gene accounts for greater than 90% of the variation in plasma lipoprotein (a) concentrations. *The Journal of clinical investigation*, 90(1):52–60, 1992.

11. Frank LJ Visseren, François Mach, Yvo M Smulders, David Carballo, Konstantinos C Koskinas, Maria Bäck, Athanase Benetos, Alessandro Biffi, Jose-Manuel Boavida, Davide Capodanno, et al. 2021 ESC Guidelines on cardiovascular disease prevention in clinical practice:

Developed by the Task Force for cardiovascular disease prevention in clinical practice with representatives of the European Society of Cardiology and 12 medical societies With the special contribution of the European Association of Preventive Cardiology (EAPC). *European heart journal*, 42(34):3227–3337, 2021.

12. ESC Cardiovasc Risk Collaboration, SCORE2 Working Group, et al. SCORE2 risk prediction algorithms: new models to estimate 10-year risk of cardiovascular disease in Europe. *European Heart Journal*, 42(25):2439–2454, 2021.

13. Cholesterol Treatment Trialists'(CTT) Collaborators: Efficacy and safety of cholesterol-lowering treatment: prospective meta-analysis of data from 90,056 participants in 14 randomised trials of statins. *Lancet*, 366:1267–1278, 2005.

14. Stephen P Adams, Michael Tsang, and James M Wright. Atorvastatin for lowering lipids. *Cochrane Database of Systematic Reviews*, (3), 2015.

15. SPRINT Research Group. A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine*, 373(22):2103–2116, 2015.

16. Anne Langsted, Pia R Kamstrup, and Børge G Nordestgaard. High lipoprotein (a) and high risk of mortality. *European heart journal*, 40(33):2760–2770, 2019.

17. Florian Kronenberg, Samia Mora, Erik SG Stroes, Brian A Ference, Benoit J Arsenault, Lars Berglund, Marc R Dweck, Marlys Koschinsky, Gilles Lambert, François Mach, et al. Lipoprotein (a) in atherosclerotic cardiovascular disease and aortic stenosis: a European Atherosclerosis Society consensus statement. *European heart journal*, 43(39):3925–3946, 2022.

18. Gregory A Roth, George A Mensah, Catherine O Johnson, Giovanni Addolorato, Enrico Ammirati, Larry M Baddour, Noël C Barengo, Andrea Z Beaton, Emelia J Benjamin, Catherine P Benziger, et al. Global burden of cardiovascular diseases and risk factors, 1990–2019: update from the GBD 2019 study. *Journal of the American College of Cardiology*, 76(25):2982–3021, 2020.

19. Peter WF Wilson, Ralph B D'Agostino, Daniel Levy, Albert M Belanger, Halit Silbershatz, and William B Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–1847, 1998.

20. David C Goff Jr, Donald M Lloyd-Jones, Glen Bennett, Sean Coady, Ralph B D'agostino, Raymond Gibbons, Philip Greenland, Daniel T Lackland, Daniel Levy, Christopher J O'donnell, et al. 2013 ACC/AHA guideline on the assessment of cardiovascular risk: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. *Circulation*, 129(25_suppl_2):S49–S73, 2014.

21. Zanfina Ademi, Jedidiah I Morton, Danny Liew, Stephen J Nicholls, Sophia Zoungas, and Brian A Ference. Integrating the biology of cardiovascular disease into the epidemiology of economic decision modelling via mendelian randomisation. *Pharmacoeconomics*, 40(11):1033–1042, 2022.

22. Jedidiah I Morton, Clara Marquina, Melanie Lloyd, Gerald F Watts, Sophia Zoungas, Danny Liew, and Zanfina Ademi. Lipid-lowering strategies for primary prevention of coronary heart disease in the uk: a cost-effectiveness analysis. *Pharmacoeconomics*, 42(1):91–107, 2024.

23. Brian A. Ference, Deepak L. Bhatt, Alberico L. Catapano, Chris J. Packard, Ian Graham, Stephen Kaptoge, Thatcher B. Ference, Qi Guo, Ulrich Laufs, Christian T. Ruff, Arjen Cupido, G. Kees Hovingh, John Danesh, Michael V. Holmes, George Davey Smith, Kausik K. Ray, Stephen J. Nicholls, and Marc S. Sabatine. Association of Genetic Variants Related to Combined Exposure to Lower Low-Density Lipoproteins and Lower Systolic Blood Pressure With Lifetime Risk of Cardiovascular Disease. *JAMA*, 322(14):1381–1391, 10 2019.

24. C Arden Pope, Robert D Brook, Richard T Burnett, and Douglas W Dockery. How is cardiovascular disease mortality risk affected by duration and intensity of fine particulate matter exposure? An integration of the epidemiologic evidence. *Air Quality, Atmosphere & Health*, 4:5–14, 2011.

25. Stephen Burgess, Brian A Ference, James R Staley, Daniel F Freitag, Amy M Mason, Sune F Nielsen, Peter Willeit, Robin Young, Praveen Surendran, Savita Karthikeyan, et al. Association of LPA variants with risk of coronary disease and the implications for lipoprotein (a)-lowering therapies: a Mendelian randomization analysis. *JAMA cardiology*, 3(7):619–627, 2018.

26. Michael D Shapiro and Deepak L Bhatt. "Cholesterol-years" for ASCVD risk prediction and treatment. *Journal of the American College of Cardiology*, 76(13):1517–1520, 2020.

27. Robyn E Wootton, Rebecca C Richmond, Bobby G Stuijfzand, Rebecca B Lawn, Hannah M Sallis, Gemma MJ Taylor, Gibran Hemani, Hannah J Jones, Stanley Zammit, George Davey Smith, et al. Evidence for causal effects of lifetime smoking on risk for depression and schizophrenia: a Mendelian randomisation study. *Psychological medicine*, 50(14):2435–2443, 2020.

28. J Zhang and K F Yu. What's the relative risk? A method of correcting the odds ratio in cohort studies of common outcomes. *JAMA*, 280(19):1690–1691, Nov 1998.

29. Jedidiah I Morton, Danny Liew, and Zanfina Ademi. A causal model for primary prevention of cardiovascular disease: the health economic model for the primary prevention of cardiovascular disease (HEM-PPCVD). *Value in Health*, 2024.

30. Brian A. Ference, Wonsuk Yoo, Issa Alesh, Nitin Mahajan, Karolina K. Mirowska, Abhishek Mewada, Joel Kahn, Luis Afonso, Kim Allan Williams, and John M. Flack. Effect of Long-Term Exposure to Lower Low-Density Lipoprotein Cholesterol Beginning Early in Life on the Risk of Coronary Heart Disease. *Journal of the American College of Cardiology*, 60(25):2631–2639, 2012.

31. Michael V Holmes, Folkert W Asselbergs, Tom M Palmer, Fotios Drenos, Matthew B Lanktree, Christopher P Nelson, Caroline E Dale, Sandosh Padmanabhan, Chris Finan, Daniel I Swerdlow, et al. Mendelian randomization of blood lipids for coronary heart disease. *European heart journal*, 36(9):539–550, 2015.

32. Brian A Ference, Jennifer G Robinson, Robert D Brook, Alberico L Catapano, M John Chapman, David R Neff, Szilard Voros, Robert P Giugliano, George Davey Smith, Sergio Fazio, et al. Variation in PCSK9 and HMGCR and risk of cardiovascular disease and diabetes. *New England Journal of Medicine*, 375(22):2144–2153, 2016.

33. Brian A Ference, Stevo Julius, Nitin Mahajan, Phillip D Levy, Kim Allan Williams Sr, and John M Flack. Clinical effect of naturally random allocation to lower systolic blood pressure beginning before the development of hypertension. *Hypertension*, 63(6):1182–1188, 2014.

34. Hannah Higgins, Amy M Mason, Susanna C Larsson, Dipender Gill, Claudia Langenberg, and Stephen Burgess. Estimating the population benefits of blood pressure lowering: a wide-angled Mendelian randomization study in UK Biobank. *Journal of the American Heart Association*, 10(17):e021098, 2021.

35. Michael G Levin, Derek Klarin, Themistocles L Assimes, Matthew S Freiberg, Erik Ingelsson, Julie Lynch, Pradeep Natarajan, Christopher O'Donnell, Daniel J Rader, Philip S Tsao, et al. Genetics of smoking and risk of atherosclerotic cardiovascular diseases: a Mendelian randomization study. *JAMA network open*, 4(1):e2034461–e2034461, 2021.

36. Susanna C Larsson, Amy M Mason, Magnus Bäck, Derek Klarin, Scott M Damrauer, Million Veteran Program, Karl Michaëlsson, and Stephen Burgess. Genetic predisposition to smoking in relation to 14 cardiovascular diseases. *European heart journal*, 41(35):3304–3310, 2020.

37. Stephanie Ross, Hertzel C Gerstein, John Eikelboom, Sonia S Anand, Salim Yusuf, and Guillaume Paré. Mendelian randomization analysis supports the causal role of dysglycaemia and diabetes in the risk of coronary artery disease. *European heart journal*, 36(23):1454–1462, 2015.

38. Wei Gan, Fiona Bragg, Robin G Walters, Iona Y Millwood, Kuang Lin, Yiping Chen, Yu Guo, Julien Vaucher, Zheng Bian, Derrick Bennett, et al. Genetic predisposition to type 2 diabetes and risk of subclinical atherosclerosis and cardiovascular diseases among 160,000 Chinese adults. *Diabetes*, 68(11):2155–2164, 2019.

39. Shiu Lun Au Yeung, Shan Luo, and C Mary Schooling. The impact of glycated hemoglobin (HbA1c) on cardiovascular disease risk: a Mendelian randomization study using UK Biobank. *Diabetes Care*, 41(9):1991–1997, 2018.

40. Nina A Hilkens, Barbara Casolla, Thomas W Leung, and Frank-Erik de Leeuw. Stroke. *The Lancet*, 2024.

41. Shuai Yuan, Bowen Tang, Jie Zheng, and Susanna C Larsson. Circulating lipoprotein lipids, apolipoproteins and ischemic stroke. *Annals of neurology*, 88(6):1229–1236, 2020.

42. George Hindy, Gunnar Engström, Susanna C Larsson, Matthew Traylor, Hugh S Markus, Olle Melander, and Marju Orho-Melander. Role of blood lipids in the development of ischemic stroke and its subtypes: a Mendelian randomization study. *Stroke*, 49(4):820–827, 2018.

43. Elsa Valdes-Marquez, Sarah Parish, Robert Clarke, Traiani Stari, Bradford B Worrall, Jemma C Hopewell, et al. Relative effects of LDL-C on ischemic stroke and coronary disease: a Mendelian randomization study. *Neurology*, 92(11):e1176–e1187, 2019.

44. Eric Yuk Fai Wan, Wing Tung Fung, C Mary Schooling, Shiu Lun Au Yeung, Man Ki Kwok, Esther Yee Tak Yu, Yuan Wang, Esther Wai Yin Chan, Ian Chi Kei Wong, and Cindy Lo Kuen Lam. Blood pressure and risk of cardiovascular disease in UK Biobank: a mendelian randomization study. *Hypertension*, 77(2):367–375, 2021.

45. Io Ieong Chan, Man Ki Kwok, and C Mary Schooling. The total and direct effects of systolic and diastolic blood pressure on cardiovascular disease and longevity using Mendelian randomisation. *Scientific Reports*, 11(1):21799, 2021.

46. Robert Clarke, Neil Wright, Robin Walters, Wei Gan, Yu Guo, Iona Y Millwood, Ling Yang, Yiping Chen, Sarah Lewington, Jun Lv, et al. Genetically Predicted Differences in Systolic Blood Pressure and Risk of Cardiovascular and Noncardiovascular Diseases: A Mendelian Randomization Study in Chinese Adults. *Hypertension*, 80(3):566–576, 2023.

47. Susanna C Larsson, Stephen Burgess, and Karl Michaëlsson. Smoking and stroke: a Mendelian randomization study. *Annals of neurology*, 86(3):468–471, 2019.

48. Eric L Harshfield, Marios K Georgakis, Rainer Malik, Martin Dichgans, and Hugh S Markus. Modifiable lifestyle factors and risk of stroke: a Mendelian randomization analysis. *Stroke*, 52(3):931–936, 2021.

49. Susanna C Larsson, Robert A Scott, Matthew Traylor, Claudia C Langenberg, George Hindy, Olle Melander, Marju Orho-Melander, Sudha Seshadri, Nicholas J Wareham, Hugh S Markus, et al. Type 2 diabetes, glucose, insulin, BMI, and ischemic stroke subtypes: Mendelian randomization study. *Neurology*, 89(5):454–460, 2017.

50. Bowen Liu, Amy M Mason, Luanluan Sun, Emanuele Di Angelantonio, Dipender Gill, and Stephen Burgess. Genetically predicted type 2 diabetes mellitus liability, glycated hemoglobin and cardiovascular diseases: a wide-angled Mendelian randomization study. *Genes*, 12(10):1644, 2021.

51. Marios K Georgakis, Eric L Harshfield, Rainer Malik, Nora Franceschini, Claudia Langenberg, Nicholas J Wareham, Hugh S Markus, and Martin Dichgans. Diabetes mellitus, glycemic traits, and cerebrovascular disease: a Mendelian randomization study. *Neurology*, 96(13):e1732–e1742, 2021.

52. Guido J Falcone, Elayna Kirsch, Julian N Acosta, Rommell B Noche, Audrey Leasure, Sandro Marini, Jaeyoon Chung, Magdy Selim, James F Meschia, Devin L Brown, et al. Genetically elevated LDL associates with lower risk of intracerebral hemorrhage. *Annals of neurology*, 88(1):56–66, 2020.

53. Zhou Yu, Linjing Zhang, Gan Zhang, Kailin Xia, Qiong Yang, Tao Huang, and Dongsheng Fan. Lipids, Apolipoproteins, Statins, and Intracerebral Hemorrhage: A Mendelian Randomization Study. *Annals of neurology*, 92(3):390–399, 2022.

54. Rachael C Aikens, Wei Zhao, Danish Saleheen, Muredach P Reilly, Stephen E Epstein, Emmi Tikkanen, Veikko Salomaa, and Benjamin F Voight. Systolic blood pressure and risk of type 2 diabetes: a Mendelian randomization study. *Diabetes*, 66(2):543–550, 2017.

55. Dianjianyi Sun, Tao Zhou, Yoriko Heianza, Xiang Li, Mengyu Fan, Vivian A Fonseca, and Lu Qi. Type 2 diabetes and hypertension: a study on bidirectional causality. *Circulation research*, 124(6):930–937, 2019.

56. Zhihong Zhu, Zhili Zheng, Futao Zhang, Yang Wu, Maciej Trzaskowski, Robert Maier, Matthew R Robinson, John J McGrath, Peter M Visscher, Naomi R Wray, et al. Causal associations between risk factors and common diseases inferred from GWAS summary data. *Nature communications*, 9(1):1–12, 2018.

57. Pia R Kamstrup and Børge G Nordestgaard. Lipoprotein (a) concentrations, isoform size, and risk of type 2 diabetes: a Mendelian randomisation study. *The lancet Diabetes & endocrinology*, 1(3):220–227, 2013.

58. Zheng Ye, Philip C Haycock, Deepti Gurdasani, Cristina Pomilla, S Matthijs Boekholdt, Sotirios Tsimikas, Kay-Tee Khaw, Nicholas J Wareham, Manjinder S Sandhu, and Nita G Forouhi. The association between circulating lipoprotein (a) and type 2 diabetes: is it causal? *Diabetes*, 63(1):332–342, 2014.

59. Connor A Emdin, Amit V Khera, Pradeep Natarajan, Derek Klarin, Hong-Hee Won, Gina M Peloso, Nathan O Stitziel, Akihiro Nomura, Seyedeh M Zekavat, Alexander G Bick, et al. Phenotypic characterization of genetically lowered human lipoprotein (a) levels. *Journal of the American College of Cardiology*, 68(25):2761–2772, 2016.

60. Nikolaus Buchmann, Markus Scholz, Christina M Lill, Ralph Burkhardt, Rahel Eckardt, Kristina Norman, Markus Loeffler, Lars Bertram, Joachim Thiery, Elisabeth Steinhagen-Thiessen, et al. Association between lipoprotein (a) level and type 2 diabetes: no evidence for a causal role of lipoprotein (a) and insulin. *Acta Diabetologica*, 54:1031–1038, 2017.

61. Susanna C Larsson, Lijuan Wang, Xue Li, Fangyuan Jiang, Xiangjun Chen, and Christos S Mantzoros. Circulating lipoprotein (a) levels and health outcomes: Phenome-wide Mendelian randomization and disease-trajectory analyses. *Metabolism*, 137:155347, 2022.

62. Susanna C Larsson and Stephen Burgess. Appraising the causal role of smoking in multiple diseases: A systematic review and meta-analysis of Mendelian randomization studies. *EBioMedicine*, 82:104154, 2022.

63. Christopher S Thom, Zhuoran Ding, Michael G Levin, Scott M Damrauer, Kyung Min Lee, Julie Lynch, Kyong-Mi Chang, Philip S Tsao, Kelly Cho, Peter WF Wilson, et al. Genetic determinants of increased body mass index mediate the effect of smoking on increased risk for type 2 diabetes but not coronary artery disease. *Human Molecular Genetics*, 29(19):3327–3337, 2020.

64. Shuai Yuan and Susanna C Larsson. An atlas on risk factors for type 2 diabetes: a wide-angled Mendelian randomisation study. *Diabetologia*, 63:2359–2371, 2020.

65. Naveed Sattar, David Preiss, Heather M Murray, Paul Welsh, Brendan M Buckley, Anton JM de Craen, Sreenivasa Rao Kondapally Seshasai, John J McMurray, Dilys J Freeman, J Wouter Jukema, Peter W Macfarlane, Chris J Packard, David J Stott, Rudi G Westendorp, James Shepherd, Barry R Davis, Sara L Pressel, Roberto Marchioli, Rosa Maria Marfisi, Aldo P Maggioni, Luigi Tavazzi, Gianni Tognoni, John Kjekshus, Terje R Pedersen, Thomas J Cook, Antonio M Gotto, Michael B Clearfield, John R Downs, Haruo Nakamura, Yasuo Ohashi, Kyoichi Mizuno, Kausik K Ray, and Ian Ford. Statins and risk of incident diabetes: a collaborative meta-analysis of randomised statin trials. *The Lancet*, 375(9716):735–742, 2010.

66. David Preiss, Sreenivasa Rao Kondapally Seshasai, Paul Welsh, Sabina A Murphy, Jennifer E Ho, David D Waters, David A DeMicco, Philip Barter, Christopher P Cannon, Marc S Sabatine, et al. Risk of incident diabetes with intensive-dose compared with moderate-dose statin therapy: a meta-analysis. *JAMA*, 305(24):2556–2564, 2011.

67. Christina Reith, David Preiss, Lisa Blackwell, Jonathan Emberson, Enti Spata, Kelly Davies, Heather Halls, Charlie Harper, Lisa Holland, Kate Wilson, et al. Effects of statin therapy on diagnoses of new-onset diabetes and worsening glycaemia in large-scale randomised blinded statin trials: an individual participant data meta-analysis. *The Lancet Diabetes & Endocrinology*, 12(5):306–319, 2024.

68. Opeyemi Soremekun, Ville Karhunen, Yiyan He, Skanda Rajasundaram, Bowen Liu, Apostolos Gkatzionis, Chisom Soremekun, Brenda Udosen, Hanan Musa, Sarah Silva, et al. Lipid traits and type 2 diabetes risk in African ancestry individuals: A Mendelian Randomization study. *EBioMedicine*, 78:103953, 2022.

69. Luca A Lotta, Stephen J Sharp, Stephen Burgess, John RB Perry, Isobel D Stewart, Sara M Willems, Jian'an Luan, Eva Ardanaz, Larraitz Arriola, Beverley Balkau, et al. Association between low-density lipoprotein cholesterol–lowering genetic variants and risk of type 2 diabetes: a meta-analysis. *JAMA*, 316(13):1383–1391, 2016.

70. Ralph A DeFronzo. From the triumvirate to the ominous octet: a new paradigm for the treatment of type 2 diabetes mellitus. *Diabetes*, 58(4):773–795, 2009.

71. Brian A Ference, Henry N Ginsberg, Ian Graham, Kausik K Ray, Chris J Packard, Eric Bruckert, Robert A Hegele, Ronald M Krauss, Frederick J Raal, Heribert Schunkert, et al. Low-density lipoproteins cause atherosclerotic cardiovascular disease. 1. Evidence from genetic, epidemiologic, and clinical studies. A consensus statement from the European Atherosclerosis Society Consensus Panel. *European heart journal*, 38(32):2459–2472, 2017.

72. Marianne Benn, Anne Tybjærg-Hansen, Stefan Stender, Ruth Frikke-Schmidt, and Børge G Nordestgaard. Low-density lipoprotein cholesterol and the risk of cancer: a mendelian randomization study. *Journal of the National Cancer Institute*, 103(6):508–519, 2011.

73. Brandon L Pierce, Peter Kraft, and Chenan Zhang. Mendelian randomization studies of cancer risk: a literature review. *Current epidemiology reports*, 5:184–196, 2018.

74. Shuai Yuan, Siddhartha Kar, Paul Carter, Mathew Vithayathil, Amy M Mason, Stephen Burgess, and Susanna C Larsson. Is type 2 diabetes causally associated with cancer risk?

Evidence from a two-sample Mendelian randomization study. *Diabetes*, 69(7):1588–1596, 2020.

75. Jonathan Pearson-Stuttard, Nikos Papadimitriou, Georgios Markozannes, Sofia Cividini, Artemisia Kakourou, Dipender Gill, Evangelos C Rizos, Grace Monori, Heather A Ward, Maria Kyrgiou, et al. Type 2 diabetes and cancer: an umbrella review of observational and mendelian randomization studies. *Cancer Epidemiology Biomarkers & Prevention*, 30(6):1218–1228, 2021.

76. Helene M Flatby, Humaira Rasheed, Anuradha Ravi, Laurent F Thomas, Kristin V Liyanarachi, Jan E Afset, Andrew T DeWan, Ben M Brumpton, Kristian Hveem, Bjørn O Åsvold, et al. Risk of lower respiratory tract infections: a genome-wide association study with Mendelian randomization analysis in three independent European populations. *Clinical Microbiology and Infection*, 28(5):732–e1, 2022.

77. Huachen Wang, Zheng Guo, Yulu Zheng, Chunyan Yu, Haifeng Hou, and Bing Chen. No casual relationship between T2DM and the risk of infectious diseases: A two-sample mendelian randomization study. *Frontiers in Genetics*, 12:720874, 2021.

78. Seyedeh M Zekavat, Michael Honigberg, James P Pirruccello, Puja Kohli, Elizabeth W Karlson, Christopher Newton-Cheh, Hongyu Zhao, and Pradeep Natarajan. Elevated blood pressure increases pneumonia risk: epidemiological association and mendelian randomization in the UK Biobank. *Med*, 2(2):137–148, 2021.

79. Catherine King, Anwar Mulugeta, Farhana Nabi, Robert Walton, Ang Zhou, and Elina Hyppönen. Mendelian randomization case-control PheWAS in UK Biobank shows evidence of causality for smoking intensity in 28 distinct clinical conditions. *EClinicalMedicine*, 26:100488, 2020.

80. Hongxiang Zhu, Xiaohui Zhan, Congjie Wang, Yuying Deng, Xiaoping Li, Linru Song, and Lingyan Zhao. Causal Associations Between Tobacco, Alcohol Use and Risk of Infectious Diseases: A Mendelian Randomization Study. *Infectious Diseases and Therapy*, 12(3):965–977, 2023.

81. Daniel B Rosoff, Joyce Yoo, and Falk W Lohoff. Smoking is significantly associated with increased risk of COVID-19 and other respiratory infections. *Communications Biology*, 4(1):1230, 2021.

82. Ben M Brumpton, Lars G Fritsche, Jie Zheng, Jonas Bille Nielsen, Maria Mannila, Ida Surakka, Humaira Rasheed, Gunnhild Åberge Vie, Sarah E Graham, Maiken Elvestad Gabrielsen, et al. Variation in Serum PCSK9 (Proprotein Convertase Subtilisin/Kexin Type 9), Cardiovascular Disease Risk, and an Investigation of Potential Unanticipated Effects of PCSK9 Inhibition: A Genome-Wide Association Study and Mendelian Randomization Study in the HUNT, Norway. *Circulation: Genomic and Precision Medicine*, 12(1):e002335, 2019.

83. Thomas H Julian, Sarah Boddy, Mahjabin Islam, Julian Kurz, Katherine J Whittaker, Tobias Moll, Calum Harvey, Sai Zhang, Michael P Snyder, Christopher McDermott, et al. A review of Mendelian randomization in amyotrophic lateral sclerosis. *Brain*, 145(3):832–842, 2022.

84. Kailin Xia, Linjing Zhang, Lu Tang, Tao Huang, and Dongsheng Fan. Assessing the role of blood pressure in amyotrophic lateral sclerosis: a Mendelian randomization study. *Orphanet Journal of Rare Diseases*, 17(1):56, 2022.

85. Sara Bandres-Ciga, Alastair J Noyce, Gibran Hemani, Aude Nicolas, Andrea Calvo, Gabriele Mora, ITALSGEN Consortium, Alessandro Arosio, Marco Barberis, Ilaria Bartolomei, et al. Shared polygenic risk and causal inferences in amyotrophic lateral sclerosis. *Annals of neurology*, 85(4):470–481, 2019.

86. Ping Zeng and Xiang Zhou. Causal effects of blood lipids on amyotrophic lateral sclerosis: a Mendelian randomization study. *Human molecular genetics*, 28(4):688–697, 2019.

87. Xu Chen, Solmaz Yazdani, Fredrik Piehl, Patrik KE Magnusson, and Fang Fang. Polygenic link between blood lipids and amyotrophic lateral sclerosis. *Neurobiology of aging*, 67:202–e1, 2018.

88. Yangfan Zhao and Gagliano Taliun SA. Lipid-lowering drug targets and Parkinson's disease: A sex-specific Mendelian randomization study. *Frontiers in Neurology*, 13:940118–940118, 2022.

89. Marianne Benn, Børge G Nordestgaard, Ruth Frikke-Schmidt, and Anne Tybjærg-Hansen. Low LDL cholesterol, PCSK9 and HMGCR genetic variation, and risk of Alzheimer's disease and Parkinson's disease: Mendelian randomisation study. *bmj*, 357, 2017.

90. Dylan M Williams, Sara Bandres-Ciga, Karl Heilbron, David Hinds, Alastair J Noyce, 23andMe Research Team, and International Parkinson's Disease Genomics Consortium.

91. Ge Liu, Mingjian Shi, Jonathan D Mosley, Chunhua Weng, Yanfei Zhang, Ming Ta Michael Lee, Gail P Jarvik, Hakon Hakonarson, Bahram Namjou-Khales, Patrick Sleiman, et al. A mendelian randomization approach using 3-HMG-coenzyme-A reductase gene variation to evaluate the association of statin-induced low-density lipoprotein cholesterol lowering with noncardiovascular disease phenotypes. *JAMA network open*, 4(6):e2112820–e2112820, 2021.

92. Kye Won Park, Yun Su Hwang, Seung Hyun Lee, Sungyang Jo, and Sun Ju Chung. The effect of blood lipids, type 2 diabetes, and body Mass Index on Parkinson's disease: a Korean mendelian randomization study. *Journal of Movement Disorders*, 16(1):79, 2023.

93. Zheng Jiang, Xiao-Jing Gu, Wei-Ming Su, Qing-Qing Duan, Yan-Lin Ren, Ju-Rong Li, Li-Yi Chi, Yi Wang, Bei Cao, and Yong-Ping Chen. Protective effect of antihypertensive drugs on the risk of Parkinson's disease lacks causal evidence from mendelian randomization. *Frontiers in Pharmacology*, 14, 2023.

94. Harneek Chohan, Konstantin Senkevich, Radhika K Patel, Jonathan P Bestwick, Benjamin M Jacobs, Sara Bandres Ciga, Ziv Gan-Or, and Alastair J Noyce. Type 2 diabetes as a determinant of Parkinson's disease risk and progression. *Movement disorders*, 36(6):1420–1429, 2021.

95. Benjamin Mappin-Kasirer, Hongchao Pan, Sarah Lewington, Jennifer Kizza, Richard Gray, Robert Clarke, and Richard Peto. Tobacco smoking and the risk of Parkinson disease: A 65-year follow-up of 30,000 male British doctors. *Neurology*, 94(20):e2132–e2138, 2020.

96. Johanna Sieurin, Yiqiang Zhan, Nancy L Pedersen, and Karin Wirdefeldt. Neuroticism, smoking, and the risk of Parkinson's disease. *Journal of Parkinson's Disease*, 11(3):1325–1334, 2021.

97. Carmen Domínguez-Baleón, Jue-Sheng Ong, Clemens R Scherzer, Miguel E Rentería, and Xianjun Dong. Understanding the effect of smoking and drinking behavior on Parkinson's disease risk: a Mendelian randomization study. *Scientific Reports*, 11(1):13980, 2021.

98. Karl Heilbron, Melanie P Jensen, Sara Bandres-Ciga, Pierre Fontanillas, Cornelis Blauwendraat, Mike A Nalls, Andrew B Singleton, George Davey Smith, Paul Cannon, Alastair J Noyce, et al. Unhealthy behaviours and risk of Parkinson's disease: a Mendelian randomisation study. *Journal of Parkinson's Disease*, 11(4):1981–1993, 2021.

99. Søren D Østergaard, Shubhabrata Mukherjee, Stephen J Sharp, Petroula Proitsi, Luca A Lotta, Felix Day, John RB Perry, Kevin L Boehme, Stefan Walter, John S Kauwe, et al. Associations between potentially modifiable risk factors and Alzheimer disease: a Mendelian randomization study. *PLoS medicine*, 12(6):e1001841, 2015.

100. Dylan M Williams, Chris Finan, Amand F Schmidt, Stephen Burgess, and Aroon D Hingorani. Lipid lowering and Alzheimer disease risk: A mendelian randomization study. *Annals of neurology*, 87(1):30–39, 2020.

101. Jiang-Shan Tan, Meng-Jin Hu, Yan-Min Yang, and Yue-Jin Yang. Genetic predisposition to low-density lipoprotein cholesterol may increase risks of both individual and Familial Alzheimer's Disease. *Frontiers in Medicine*, 8:798334, 2022.

102. Yuesong Pan, Hao Li, Yilong Wang, Xia Meng, and Yongjun Wang. Causal effect of Lp (a)[lipoprotein (a)] level on ischemic stroke and Alzheimer disease: a Mendelian randomization study. *Stroke*, 50(12):3532–3539, 2019.

103. Susanna C Larsson, Matthew Traylor, Rainer Malik, Martin Dichgans, Stephen Burgess, and Hugh S Markus. Modifiable pathways in Alzheimer's disease: Mendelian randomisation analysis. *bmj*, 359, 2017.

104. Ya-Nan Ou, Yu-Xiang Yang, Xue-Ning Shen, Ya-Hui Ma, Shi-Dong Chen, Qiang Dong, Lan Tan, and Jin-Tai Yu. Genetically determined blood pressure, antihypertensive medications, and risk of Alzheimer's disease: a Mendelian randomization study. *Alzheimer's Research & Therapy*, 13(1):1–9, 2021.

105. Venexia M Walker, Patrick G Kehoe, Richard M Martin, and Neil M Davies. Repurposing antihypertensive drugs for the prevention of Alzheimer's disease: a Mendelian randomization study. *International journal of epidemiology*, 49(4):1132–1140, 2020.

106. Jesper Qvist Thomassen, Janne Schurmann Tolstrup, Marianne Benn, and Ruth Frikke-Schmidt. Type-2 diabetes and risk of dementia: observational and Mendelian randomisation studies in 1 million individuals. *Epidemiology and psychiatric sciences*, 29:e118, 2020.

107. Yuesong Pan, Weiqi Chen, Hongyi Yan, Mengxing Wang, and Xianglong Xiang. Glycemic traits and Alzheimer's disease: a Mendelian randomization study. *Aging (Albany NY)*, 12(22):22688, 2020.

108. Nikki McCaffrey, Billingsley Kaambwa, David C Currow, and Julie Ratcliffe. Health-related quality of life measured using the EQ-5D–5L: South Australian population norms. *Health and quality of life outcomes*, 14(1):1–12, 2016.

109. Roberta Ara and John E. Brazier. Populating an Economic Model with Health State Utility Values: Moving toward Better Practice. *Value in Health*, 13(5):509–518, 2010.

110. Amélie Beaudet, John Clegg, Per-Olof Thuresson, Adam Lloyd, and Phil McEwan. Review of utility values for economic modeling in type 2 diabetes. *Value in Health*, 17(4):462–470, 2014.

111. Marissa Blieden Betts, Pratik Rane, Evelien Bergrath, Madhura Chitnis, Mohit Kumar Bhutani, Claudia Gulea, Yi Qian, and Guillermo Villa. Utility value estimates in cardiovascular disease and the effect of changing elicitation methods: a systematic literature review. *Health and Quality of Life Outcomes*, 18(1):251, 2020.

112. Raed A Joundi, Joel Adekanye, Alexander A Leung, Paul Ronksley, Eric E Smith, Alexander D Rebchuk, Thalia S Field, Michael D Hill, Stephen B Wilton, and Lauren C Bresee. Health State Utility Values in People With Stroke: A Systematic Review and Meta-Analysis. *Journal of the American Heart Association*, 11(13):e024296, 2022.

113. E Anne Lacey, R June Musgrave, Jenny V Freeman, Angela M Tod, and Peter Scott. Psychological morbidity after myocardial infarction in an area of deprivation in the UK: evaluation of a self-help package. *European journal of cardiovascular nursing*, 3(3):219–224, 2004.

114. Jedidiah I Morton, Clara Marquina, Jonathan E Shaw, Danny Liew, Kevan R Polkinghorne, Zanfina Ademi, and Dianna J Magliano. Projecting the incidence and costs of major cardiovascular and kidney complications of type 2 diabetes with widespread SGLT2i and GLP-1 RA use: a cost-effectiveness analysis. *Diabetologia*, 66(4):642–656, 2023.

115. Crystal Man Ying Lee, Ruth Colagiuri, Dianna J Magliano, Adrian J Cameron, Jonathan Shaw, Paul Zimmet, and Stephen Colagiuri. The cost of diabetes in adults in Australia. *Diabetes research and clinical practice*, 99(3):385–390, 2013.

116. Clara Marquina, Stella Talic, Sandra Vargas-Torres, Marjana Petrova, Dina Abushanab, Alice Owen, Sean Lybrand, David Thomson, Danny Liew, Ella Zomer, et al. Future burden of cardiovascular disease in Australia: impact on health and economic outcomes between 2020 and 2029. *European Journal of Preventive Cardiology*, 29(8):1212–1219, 2022.

117. Linda J Cobiac, Anne Magnus, Jan J Barendregt, Rob Carter, and Theo Vos. Improving the cost-effectiveness of cardiovascular disease prevention in Australia: a modelling study. *BMC public health*, 12(1):1–10, 2012.

118. The Pharmaceutical Benefits Scheme. PBS and RPBS Section 85 Date of Supply Data. 2023. Available at: https://www.pbs.gov.au/info/statistics/dos-and-dop/dos-and-dop [Accessed: 5 January 2024].

119. Australian Institute of Health and Welfare. Health expenditure Australia 2022-23. Available at: https://www.aihw.gov.au/reports/health-welfare-expenditure/health-expenditure-australia-2022-23/data [Accessed: 10 December 2024].

120. Public Health England. The health and social care costs of a selection of health conditions and multi-morbidities. July 2020. Available at: https://www.gov.uk/government/publications/health-and-social-care-costs-of-a-selection-of-health-conditions [Accessed: 12 January 2024].

121. National Health Service. 2021/22 National Cost Collection for the NHS. Available at: https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/ Accessed 12 January 2024.

122. National Health Service. Prescription Services. NHS Electronic Drug Tariff June 2023. Available at: https://www.nhsbsa.nhs.uk/pharmacies-gp-practices-and-appliance-contractors/drug-tariff/back-copies-drug-tariff Accessed 10 December 2024.

123. National Health Service Dictionary of Medicines and Devices. Actual Medicinal Product pack (AMPP). Leqvio 284mg/1.5ml solution for injection pre-filled syringes (Novartis Pharmaceuticals UK Ltd) 1 pre-filled disposable injection. Available at: https://dmd-browser.nhsbsa.nhs.uk/amp/view/160939 [Accessed 10 January 2024].

124. National Health Service. 2022/23 National Cost Collection for the NHS. Available at: https://www.england.nhs.uk/costing-in-the-nhs/national-cost-collection/ Accessed 10 December 2024.

125. Personal Social Services Research Unit, University of Kent. Unit Costs of Health and Social Care 2023. 12 Sources of Information. 12.1 Inflation indices. 12.1.1 The NHS Cost Inflation Index (NHSCII). Available at: https://kar.kent.ac.uk/105685/ Accessed 10 December 2024 January 2024.

126. WB Van den Hout. The value of productivity: human-capital versus friction-cost method. *Annals of the rheumatic diseases*, 69(Suppl 1), 2010.

127. Australian Bureau of Statistics. Average Weekly Earnings, Australia. May 2024. Available at: https://www.abs.gov.au/statistics/labour/earnings-and-working-conditions/average-weekly-earnings-australia/may-2024 [Accessed 4/11/2024].

128. Office for National Statistics, UK. Average weekly earnings in Great Britain: September 2024. Available at: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/bulleti [Accessed 18/9/2024].

129. Australian Bureau of Statistics. Labour Force, Australia, Detailed. September 2024. Available at: https://www.abs.gov.au/statistics/labour/employment-and-unemployment/labour-force-australia-detailed/latest-release [Accessed 4/11/2024].

130. Office for National Statistics, UK. A05 SA: Employment, unemployment and economic inactivity by age group (seasonally adjusted): September 2024. Available at: https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/datase [Accessed 18/9/2024].

131. Marian U Worcester, Peter C Elliott, Alyna Turner, Jeremy J Pereira, Barbara M Murphy, Michael R Le Grande, Katherine L Middleton, Hema S Navaratnam, John K Nguyen, Robert W Newman, et al. Resumption of work after acute coronary syndrome or coronary artery bypass graft surgery. *Heart, Lung and Circulation*, 23(5):444–453, 2014.

132. Kathryn Radford, Mary I Grant, Emma J Sinclair, Jade Kettlewell, and Connor Watkin. Describing return to work after stroke: a feasibility trial of 12-month outcomes. *Journal of Rehabilitation Medicine*, 52:1–8, 2020.

133. Xue Song, Ruben GW Quek, Shravanthi R Gandra, Katherine A Cappell, Robert Fowler, and Ze Cong. Productivity loss and indirect costs associated with cardiovascular events and related clinical procedures. *BMC health services research*, 15:1–14, 2015.

134. Marie-Claude Breton, Line Guenette, Mohamed Amine Amiche, Jeanne-Francoise Kayibanda, Jean-Pierre Gregoire, and Jocelyne Moisan. Burden of diabetes on the ability to work: a systematic review. *Diabetes care*, 36(3):740–749, 2013.

135. Muhammad Shahdaat Bin Sayeed, Grace Joshy, Ellie Paige, Emily Banks, and Rosemary Korda. Cardiovascular disease subtypes, physical disability and workforce participation: A cross-sectional study of 163,562 middle-aged Australians. *Plos one*, 16(4):e0249738, 2021.

136. Xiaohui Zhang, Xueyan Zhao, and Anthony Harris. Chronic diseases and labour force participation in Australia. *Journal of Health Economics*, 28(1):91–108, 2009.

137. Department of Health. The Pharmaceutical Benefits Advisory Committee (PBAC) Guidelines. https://pbac.pbs.gov.au [Accessed 28 September 2021].

138. Guide to the methods of technology appraisal. *National Institute for Health and Clinical Excellence (NICE) London, UK*, 2013.

139. Laura Catherine Edney, Hossein Haji Ali Afzali, Terence Chai Cheng, and Jonathan Karnon. Estimating the reference incremental cost-effectiveness ratio for the Australian health system. *Pharmacoeconomics*, 36:239–252, 2018.

140. Kingshuk Pal, Laura Horsfall, Manuj Sharma, Irwin Nazareth, and Irene Petersen. Time trends in the incidence of clinically diagnosed type 2 diabetes and pre-diabetes in the UK 2009–2018: a retrospective cohort study. *BMJ Open Diabetes Research and Care*, 9(1):e001989, 2021.

141. Australian Institute of Health and Welfare (2009) Chronic disease and participation in work. Catalogue number PHE 109. Australian Institute of Health and Welfare, Canberra.

142. Bendix Carstensen. Age–period–cohort models for the Lexis diagram. *Statistics in medicine*, 26(15):3018–3045, 2007.

143. Frank E Harrell. *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*, volume 608. Springer, 2001.

144. Estimates of the population for the UK, England, Wales, Scotland and Northern Ireland: Mid-2013 edition of this dataset Superseded. Available at:

https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/da
[Accessed 16 June 2023].

145. Olivier S Descamps, Monique Bruniaux, Pierre-Francois Guilmot, René Tonglet, and Francis R Heller. Lipoprotein concentrations in newborns are associated with allelic variations in their mothers. *Atherosclerosis*, 172(2):287–298, 2004.

146. Brian K. Kit, Margaret D. Carroll, David A. Lacher, Paul D. Sorlie, Janet M. DeJesus, and CyntjL. Ogden. Trends in Serum Lipids Among US Youths Aged 6 to 19 Years, 1988-2010. *JAMA*, 308(6):591–600, 08 2012.

147. National High Blood Pressure Education Program Working Group on High Blood Pressure in Children and Adolescents. The fourth report on the diagnosis, evaluation, and treatment of high blood pressure in children and adolescents. *Pediatrics*, 114(Supplement_2):555–576, 2004.

148. Hongwei Ji, Andy Kim, Joseph E Ebinger, Teemu J Niiranen, Brian L Claggett, C Noel Bairey Merz, and Susan Cheng. Sex differences in blood pressure trajectories over the life course. *JAMA cardiology*, 5(3):255–262, 2020.

149. Bendix Carstensen, Pernille Falberg Rønn, and Marit Eika Jørgensen. Prevalence, incidence and mortality of type 1 and type 2 diabetes in Denmark 1996–2016. *BMJ Open Diabetes Research and Care*, 8(1):e001071, 2020.

150. Manuj Sharma, Irwin Nazareth, and Irene Petersen. Trends in incidence, prevalence and prescribing in type 2 diabetes mellitus between 2000 and 2013 in primary care: a retrospective cohort study. *BMJ open*, 6(1):e010210, 2016.

151. United Kingdom Office for National Statistics. Adult Smoking Habits in the UK: 2021. Available at: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandlifeexpecta
[Accesed: 19 July 2023].

152. Meredith S Duncan, Ramachandran S Vasan, and Vanessa Xanthakis. Trajectories of blood lipid concentrations over the adult life course and risk of cardiovascular disease and all-cause mortality: observations from the Framingham study over 35 years. *Journal of the American Heart Association*, 8(11):e011433, 2019.

153. Peter P. Toth, Craig Granowitz, Michael Hull, Amy Anderson, and Sephy Philip. Long-term statin persistence is poor among high-risk patients with dyslipidemia: a real-world administrative claims analysis. *Lipids in Health and Disease*, 18(1):175, 2019.

154. Stella Talic, Clara Marquina, Richard Ofori-Asenso, Marjana Petrova, Danny Liew, Alice J Owen, Sean Lybrand, David Thomson, Jenni Ilomäki, Ella Zomer, and Zanfina Ademi. Switching, Persistence and Adherence to Statin Therapy: a Retrospective Cohort Study Using the Australian National Pharmacy Data. *Cardiovasc Drugs Ther*, Jun 2021.

155. Richard Ofori-Asenso, Avtar Jakhu, Ella Zomer, Andrea J Curtis, Maarit Jaana Korhonen, Mark Nelson, Manoj Gambhir, Andrew Tonkin, Danny Liew, and Sophia Zoungas. Adherence and Persistence Among Statin Users Aged 65 Years and Over: A Systematic Review and Meta-analysis. *The Journals of Gerontology: Series A*, 73(6):813–819, 09 2017.

156. Hao Sen Andrew Fang, Qiao Gao, Mong Li Lee, Wynne Hsu, and Ngiap Chuan Tan. LDL-cholesterol change and goal attainment following statin intensity titration among Asians in primary care: a retrospective cohort study. *Lipids Health Dis*, 20(1):2, Jan 2021.

157. Dirk De Bacquer, Delphine De Smedt, Željko Reiner, Lale Tokgözoğlu, Els Clays, Kornelia Kotseva, Lars Rydén, David Wood, and Guy De Backer. Percentage low-density lipoprotein-cholesterol response to a given statin dose is not fixed across the pre-treatment range: Real world evidence from clinical practice: Data from the ESC-EORP EUROASPIRE V Study. *European Journal of Preventive Cardiology*, 27(15):1630–1636, 10 2020.

158. MR Law, NJ Wald, JK Morris, and RE Jordan. Value of low dose combination treatment with blood pressure lowering drugs: analysis of 354 randomised trials. *BMJ*, 326(7404):1427, 2003.

159. Stephen P Adams, Sarpreet S Sekhon, and James M Wright. Rosuvastatin for lowering lipids. *Cochrane Database of Systematic Reviews*, (11), 2014.

160. Zanfina Ademi, Yuki Tomonaga, Joris van Stiphout, Dominik Glinz, Viktoria Gloy, Heike Raatz, Heiner C Bucher, and Matthias Schwenkglenks. Adaptation of cost-effectiveness analyses to a single country: the case of bariatric surgery for obesity and overweight. *Swiss medical weekly*, 148:w14626, 2018.

161. Organisation for Economic Co-operation and Development. Health Spending. Available from: https://www.oecd.org/en/data/indicators/health-spending.html [Accessed 6 November 2024].

162. Organisation for Economic Co-operation and Development. Purchasing power parities (PPP). Available from: https://www.oecd.org/en/data/indicators/purchasing-power-parities-ppp.html [Accessed 19 November 2024].

163. Organisation for Economic Co-operation and Development. Average Annual Wages. Available from: https://www.oecd.org/en/data/indicators/average-annual-wages.html [Accessed 6 November 2024].

164. Andres Pichon-Riviere, Michael Drummond, Alfredo Palacios, Sebastián Garcia-Marti, and Federico Augustovski. Determining the efficiency path to universal health coverage: cost-effectiveness thresholds for 174 countries based on growth in life expectancy and health expenditures. *The Lancet Global Health*, 11(6):e833–e842, 2023.