Revascularization strategy following NSTEMI and predicting secondary prevention medication use: a linked data study.

Adam Livori
PhD Candidate
adam.livori1@monash.edu

Center for Medication Use and Safety, Faculty of Pharmacy and Pharmaceutical Sciences, Monash Unviersity, Melbourne, Australia

Contents

List of Figures

1 Preface

The is the protocol for the paper Revascularization strategy following NSTEMI and predicting secondary prevention medication use: a linked data study.

This protocol details the data preparation (cleaning that was undertaken from a linked dataset provided by the Victorian Department of Health as the source of Victorian Admitted Episodes Dataset (VAED), and the Centre for Vicotiran Data Linkage for provision of data linkage). This study was approved by the Human Research and Ethics Committees from the Australian Institute for Health and Welfare (AIHW) (EO2018/4/468) and Monash University (14339).

The initial cohort, with removal of nested admissions, transfers and same-day admissions was previously cleaned and presented in a previous protocol.

To generate this document, the Stata package texdoc was used, which is avaiable from here. The do file was orignially coded and then exported from the Secure Unified Research Environment (SURE), and so the histograms geenreated were exported separately and then imported via LaTex coding due to constraints on exporting raw data from SURE. Therefore, when reproducing the code, use the do file rather than copying from the LaTex document.

2 Abbreviations

- ABS: Australian Bureau of Statistics

- ACEi: Angiotensin converting enzyme inhibitor

- AF: Atrial fibrillation

- AFL: Atrial flutter

- AIHW: Australian Institute of Health and Welfare

- ARB: Angiotensin II receptor blocker

- ARIAl Accessibility/remonteness index of Australia

- ATC: Anatomical therapeutic chemical

- CABG: Coronary artery bypass graft

- CBD: Cerebrovascular disease

- CCU: Coronary care unit

- CHF: Heart failure

- CPD: Chronic pulmonary disease

- NOAC: Non-vitaminm K oral anticoagulant

- DM: Diabetis mellitus

- DMNC: DM without chronic complications

- DMWC: Diabets with chronic complications

- HIV: Human immunodeficiency virus or acquired immunodeficiency disease

- HOP: Hemiplegia or paraplegia

- HT: Hypertension

- ICD: International classification fo disease (ICD-10 Australian Modified codes were present in this dataset)

- IRSD: Index of relative socioeconomic disadvantage

- MAL : Malignancy

- MBS: Medicare benefit scheme

- MI: Myocardial infarction

- MIC: Previous MI

- MLD: Mild liver disease

- MST: Metastatic solid tumour

- NDI: National death index

- NOAC: Non-vitamin K oral anticoagulant

- NSTEMI: Non-ST elevation myocardial infarction

- P2Y12i: Adenosine diphosphate receptor (P2Y12) inhibitor

- PBS: Pharmacetical benefit scheme

- PCI: Percutaneous coronary intervention

- PDC: Proportion of days covered

- PUD: Peptic ulcer disease

- PVD: Peripheral vascular disease

- REND: Renal disease

- RHEUD: Rheumatic disease

- SLA: Statistical local area

- SLD: Severe liver disease

- STEMI: ST elevation myocardial infarction

- VAED: Victorian admitted episode dataset

- WHO: World Health Organisation

3 Introduction

The core management strategy for myocardial infarction is the use of revascularisation, either by percutaneous coronary intervention or coronary artery bypass grafts.[1] The life-prolonging benefits of these revascularisation strategies have revolutionised cardiac care following myocardial infarction.1 Following myocardial infarction, regardless of revascularisation strategy, the use secondary prevention medications (dual antiplatelet therapy; angiotensin converting enzyme inhibitors/angiotensin receptor blockers (ACEi/ARB); HMG Co-A reductase inhibitors (statins); and beta blockers) are recommended. The rationale stands that revascularisation will correct current stenoses impacting cardiac output and potential symptoms, but not prevent plaque progression and possible ventricular remodelling.2 Both strategies have their respective advantages, with selection of revascularisation strategy individualised to characteristics such as coronary anatomy, single or multi-vessel disease, comorbidities and preference of the person receiving revascularisation.[1]

The objective of this study was to determine how revascularisation strategy following NSTEMI was associated with secondary prevention medication use in the 12 months post discharge.

4 MI cohort dataset creation

4.1 Previously cleaned cohort import

We have used a previously cleaned cohort of people admitted with a prmiary diagnosis of MI from Victorian Hospitals. We cleaned nested admissions and transfers, creating unique episodes of care for each MI. This dataset was then merged with a dataset from the NDI, allowing for confirmation of dates of death during the follow up period (until 30/6/2018).

```
STOP
set rmsg on
cd "G:\Adam\Project - revasc and outcome prediction\Data"
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_ndi", clear
br
hist  admdate, color(black) graphregion(color(white)) frequency title(Histrogram of MI admissions) x
> title(MI admission date)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_miadmissions", replace
```
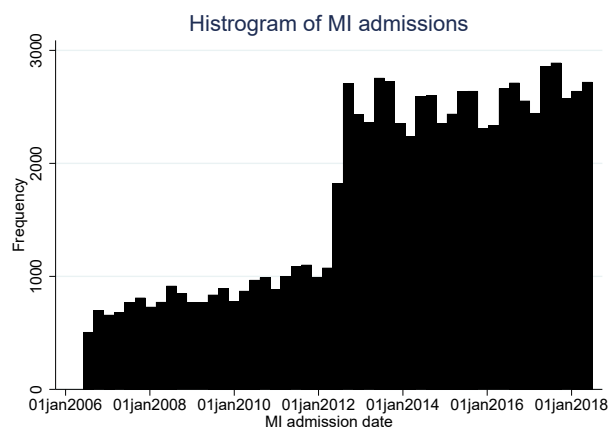

Figure 4.1: Histogram of all MI admissions from VAED dataset

4.2 Removal of nested admissions

The histogram details the total lookback avaiable for all MI admissions, however the PBS data avaiable for linkage is limited between 2012 and 2018, so our cohort was selected from these dates, with lookback to 2006 avaiable for previous MI events (which was done in previous coding protocol.

We also have a dataset that contains the MI admissions, but contains all diagnosis and procedure codes (up to 40 within each admission) for each admission which will be used for assigning cormorbidities.

```
hist  admdate if admdate >= td(1/7/2012), color(black) graphregion(color(white)) frequency title(His
> trogram of MI admissions for selected cohort) xtitle(MI admission date)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_cohortmiadmissions", replac
> e
```

4.3 Removal of nested admissions
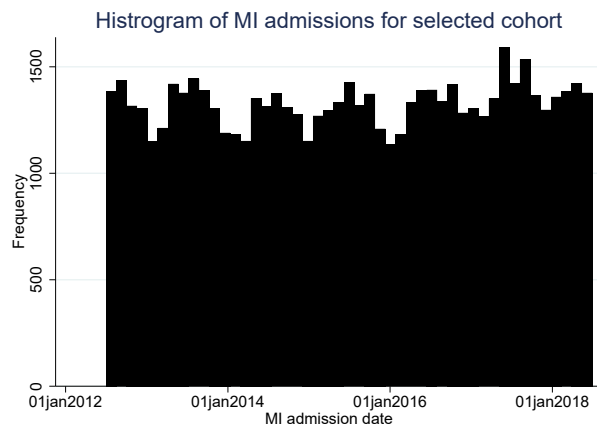
Figure 4.2: Histrogram of MI admissions for selected cohort

```
save MI_cohort_NDI, replace
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_nest_tf_free", clear
br
save MI_cohort_nest_tf_free, replace
```

5   Identification of revascularisation strategies used

5.1   Revascularisation following MI

This step followed a similar process as the comorbidity assignment, however we used the procedure codes for CABG and PCI within 30 days of admission for MI. In addition to creating a variable listing date of procedure, binary tags for PCI and CABG were created. A combined tag for either procedure was also created, noting that 122 MI admissions had both procedures within 30 days of admission.

```
use MI_proc, clear
br
foreach i in PCI CABG {
gen prev`i´ = 0
bysort ppn (admdate) : replace prev`i´ = 1 if `i´[_n-1] != .
bysort ppn (admdate) : replace prev`i´ =1 if prev`i´[_n-1] == 1
}
save MI_proc_prev, replace

use MI_proc_clean, clear

merge 1:m ppn admdate using MI_proc_prev
keep if _merge == 3
bysort ppn admdate : gen nghost = _n
br
keep if nghost == 1
drop nghost _merge

save MI_proc_all, replace
```

5.2   Removal of nested admissions

```
hist PCI if PCI > td(1/7/2012), bin(100) color(black) graphregion(color(white)) frequency title(PCIs
```
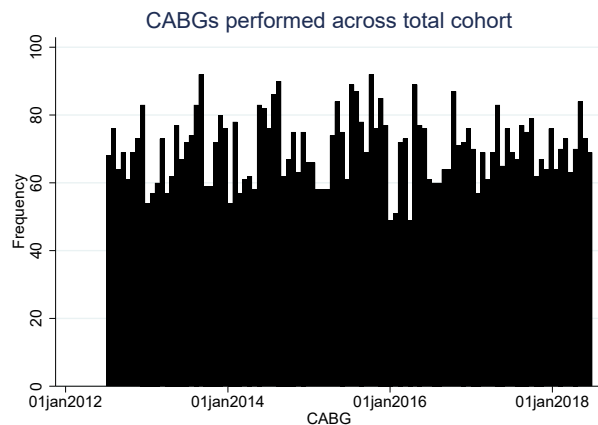
Figure 5.1: Histrogram of MI admissions with CABG for cohort

```
> performed across total cohort)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_miadmissionspci", replace
```
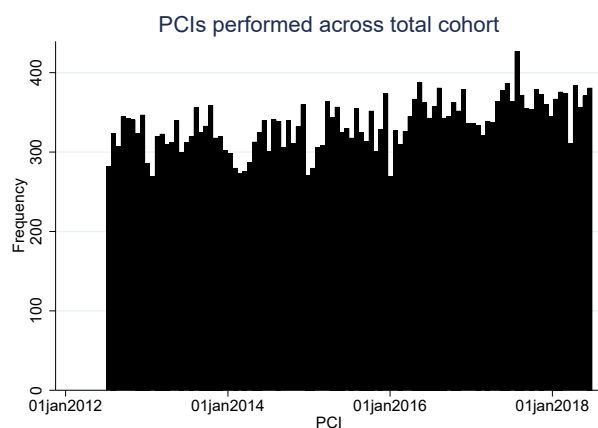

Figure 5.2: Histrogram of MI admissions with PCI for cohort

5.3 Removal of nested admissions

```
keep ppn admdate CABG PCI
save MI_proc, replace

count if CABG != .
use MI_proc, clear

quietly {
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i´
merge 1:m ppn using MI_proc
keep if _merge==3
keep ppn admdate sepdate CABG PCI
```

9

```
drop if CABG < admdate
drop if PCI <admdate
save MI_proc_merge`i´, replace
}
}
*Extract unique CABG procedures for MI admissions
clear
forval i = 1/16 {
append using MI_proc_merge`i´
}
count if CABG < admdate
count if PCI < admdate
drop PCI
drop if CABG == .
bysort ppn admdate (CABG): keep if _n == 1
save CABGppl, replace
*Extract unique PCI procedures for MI admissions
clear
forval i = 1/16 {
append using MI_proc_merge`i´
}
count if CABG < admdate
count if PCI < admdate
drop CABG
drop if PCI ==.
bysort ppn admdate (PCI): keep if _n == 1
save PCIppl, replace

*Bring PCI and CABG together
use PCIppl, clear
merge 1:1 ppn admdate using MI_cohort_ndi
drop _merge
merge 1:1 ppn admdate using CABGppl
drop _merge

*See time since MI to PCI and CABG
bysort ppn admdate (CABG PCI) : keep if _n == 1
keep if inrange(sepdate,td(1/7/2012),td(30/6/2017))
gen CABGdist = CABG - admdate
hist CABGdist, bin(30) color(black) graphregion(color(white)) frequency title(Time from MI admission
>  to CABG)
graph save "G:\Adam\Project – revasc and outcome prediction\Results\hist_timetocabg", replace
```
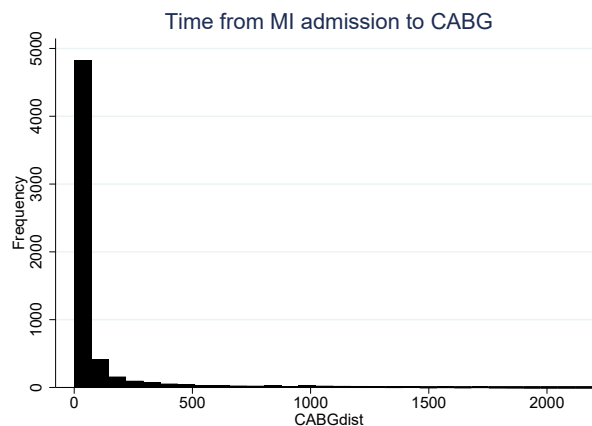

Figure 5.3: Histogram of time from admission to CABG for cohort

5.4 Removal of nested admissions

```
gen PCIdist = PCI - admdate
hist PCIdist, bin(30) color(black) graphregion(color(white)) frequency title(Time from MI admission
> to PCI)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_timetopci", replace
```
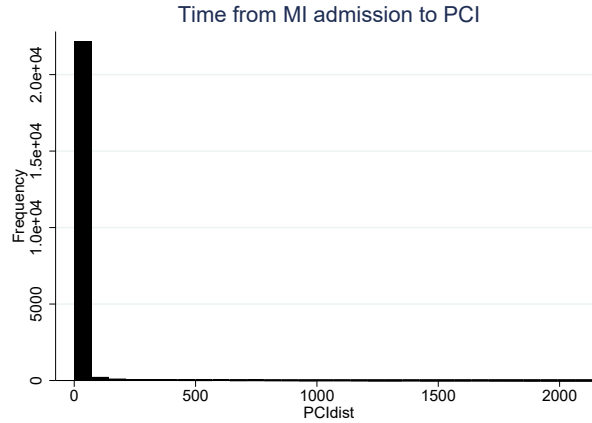

Figure 5.4: Histogram of time from admission to PCI for cohort

5.5 Removal of nested admissions

```
gen CABG_tag = 0
replace CABG_tag = 1 if CABG !=. & CABG <= sepdate + 30
ta CABG_tag
gen PCI_tag = 0
replace PCI_tag = 1 if PCI !=. & PCI <= sepdate + 30
ta PCI_tag
gen PCI_CABG_tag = 0
replace PCI_CABG_tag = 1 if PCI_tag == 1 | CABG_tag == 1
ta PCI_CABG_tag
save MI_proc_clean, replace
```

5.6 Prior revascularisation

We also wanted to know whether a person had received CABG or PCI prior to their MI, we used the same procedure codes as above, but performed a lookback across all VAED data available (earliest admission data is from 2005). Once we created the tags for previous revascularisation, we merged them back into the procedure dataset.

```
use MI_proc, clear
br
foreach i in PCI CABG {
gen prev`i´ = 0
bysort ppn (admdate) : replace prev`i´ = 1 if `i´[_n-1] != .
bysort ppn (admdate) : replace prev`i´ =1 if prev`i´[_n-1] == 1
}
save MI_proc_prev, replace
use MI_proc_clean, clear
```

11

```
merge 1:m ppn admdate using MI_proc_prev
keep if _merge == 3
bysort ppn admdate : gen nghost = _n
br
keep if nghost == 1
drop nghost _merge

save MI_proc_all, replace
```

6 Medication use

6.1 Identifying medications of interest from PBS data

This first step has been previously reported in another protocol . This includes identifying medications from antiplatelet (P2Y12i), ACEi, ARB, beta blocker and statin (which includes combination statin with ezetimibe, but not ezetimibe monotherapy). For use in the secondary analysis, we also tagged statin therapy defined as high intenstiy, which includes atorvastatin 40mg and 80mg, and rosuvastatin 20mg and 40mg. [2]

We also tagged MI admissions with previous supply of secondary prevention medications in the 60 days prior to MI admission.

```
use MI_cohort_ndi_PBS, clear
br
rename P2Y12 p2y12

*Tag high intensity statins via PBS item codes
gen hstatin = 0
replace hstatin = 1 if pbs_item_code == "08215J" | pbs_item_code == "09232X" | pbs_item_code == "085
> 21L" | pbs_item_code == "09233Y" | pbs_item_code == "10376D" | pbs_item_code == "10377E" | pbs_ite
> m_code == "09052K" | pbs_item_code == "09051J"

ta statin hstatin if statin == 1

*Supply of medications following discharge for MI
gen supplied_60 = 0
replace supplied_60 = 1 if inrange(supplydate, sepdate, sepdate+60)

*Supply of medications prior to MI admission
gen supplied_priorMI = 0
replace supplied_priorMI = 1 if inrange(supplydate, admdate-90, admdate-1)
ta supplied_priorMI supplied_60

*Combine supplied within 60 days and tag per drug class
foreach i in beta ace arb p2y12 statin ace_arb hstatin {
gen `i'_60 = 0
replace `i'_60 = 1 if `i' == 1 & supplied_60 == 1
}

*Combine supplied prior to admission and tag
foreach i in beta ace arb p2y12 statin ace_arb hstatin {
gen `i'_prior = 0
replace `i'_prior = 1 if `i' == 1 & supplied_priorMI == 1
}

gen inpt_supply = 0
replace inpt_supply = 1 if supplydate < sepdate & supplydate > = admdate
ta inpt_supply
*It appears 1.7% of dispensing after the MI admission occurred within the episode of care.

save MI_drug_all, replace
```

6.2 Prior dispesning of medications

We collapsed dispensings tagged per medication class and occurring within 90 days of admission. This set includes dispensings of all MI admissions and was merged with NDI cohort. This still contains MIs that occurred prior to and after the study inclusion period of 1/7/2012 and 30/6/2017, and will be removed when creating the analysis data set.

```
use MI_drug_all, clear

collapse (sum) beta_prior ace_prior arb_prior p2y12_prior statin_prior ace_arb_prior hstatin_prior,
> by(ppn admdate sepdate)
foreach i in  beta_prior ace_prior arb_prior p2y12_prior statin_prior ace_arb_prior hstatin_prior {
replace `i´ = 1  if `i´ >= 1
}
merge 1:1 ppn admdate using MI_cohort_NDI
keep ppn admdate beta_prior-hstatin_prior
foreach i in beta ace arb p2y12 statin ace_arb hstatin {
replace `i´_prior = 0 if `i´_prior ==.
}

save MI_drug_prior, replace
```

6.3   Initial dispesning of medications

We collapsed dispensings tagged per medication class and occurring within 60 days of discharge. This set includes dispensings of all MI admissions and was merged with NDI cohort. This still contains MIs that occurred prior to and after the study inclusion period of 1/7/2012 and 30/6/2017, and will be removed when creating the analysis data set.

```
use MI_drug_all, clear
collapse (sum) beta_60 ace_60 arb_60 p2y12_60 statin_60 ace_arb_60 hstatin_60, by(ppn admdate sepdat
> e)
foreach i in  beta_60 ace_60 arb_60 p2y12_60 statin_60 ace_arb_60 hstatin_60 {
replace `i´ = 1  if `i´ >= 1
}
merge 1:1 ppn admdate using MI_cohort_NDI
keep ppn admdate beta_60-hstatin_60
foreach i in beta ace arb p2y12 statin ace_arb hstatin {
replace `i´_60 = 0 if `i´_60 ==.
}
gen OMT  = beta_60 + p2y12_60 + statin_60 + ace_arb_60

save MI_drug_received, replace
```

Using the dataset that contains identification of the secondary prevention medication dispensed, we created class specific sets of dispensings for use in the PDC calculation steps. These sets would include all dispensings of medications within each class across the study period

```
use MI_drug_all, clear
br

gen drug_class=0
replace drug_class = 1 if p2y12==1
replace drug_class = 2 if beta==1
replace drug_class = 3 if statin==1
replace drug_class = 4 if ace ==1
replace drug_class = 5 if arb ==1
ta drug_class
save MI_drug_class, replace

forval i = 1/5 {
use MI_drug_class, clear
keep if drug_class == `i´
keep ppn admdate sepdate deathdate supplydate pbs_item_code drug_class hstatin
save MI_drug_class_`i´, replace
}
use MI_drug_class, clear
keep if ace_arb == 1
keep ppn admdate sepdate deathdate supplydate pbs_item_code drug_class hstatin
save MI_drug_class_6, replace
```

6.4 Proportions of days covered

6.4.1 PBS item packsizes for determining days supply

We reported the use of PBS information to check pack size changs across the study period in a previous protocol. This dataset was used for each drug class in order to assign packsizes for each drug class.

6.4.2 PDC calculation parameters

c We created persistence measures acrosss each drug class for each participant per MI admission. The TEN-spiders tool for reporting was used to detail the steps, definitions, and assumptions used in calculating PDC. [3]
The method for this study was done with the following steps in mind:

1. Determine each observation's study period, which in this case is 365 days (admdate + 365), unless they died prior to this date, in which case the value is death date - MI discharge date (sepdate). Noting admissions where people died within 90 days of discharge were also excluded.

2. Count the days the patient was covered by the drug based on supply date, with adjusting for credit when supply date occurs prior to previous supply ending

3. Divide number of covered days in step 2 by number of days (study date) in step 1

6.4.3 Antiplatelets

```
use MI_drug_class_1, clear
drop if deathdate <= sepdate + 90

*Firstly, our start date for each MI will be the sepdate
*Secondly, our end date is sepdate + 365
drop if supplydate <= sepdate - 60
drop if supplydate >= admdate & supplydate < sepdate
gen enddate = sepdate +365
format enddate %td

*We know need to correct study period end dates for participants who died prior to the follow up per
> iod.

replace enddate = deathdate if deathdate <= enddate
drop if supplydate >= deathdate
br
*We now calculate number of days in the study period

gen study_days = enddate-sepdate

*Merge in packsize data from PBS

merge m:1 pbs_item_code using PBS_packsizes
keep if _merge == 3
drop _merge
replace packsize = 28 if packsize == 56
ta packsize
br
bysort ppn admdate (supplydate): gen nghost=_n
ta nghost

*Therefore 30 dispensings is the most dispensings within the follow up period

drop nghost
rename packsize dayssupply
drop pbs_item_code

gen supplygap = 0
replace supplygap = admdate - supplydate if supplydate < admdate
drop if supplygap >= dayssupply
replace supplydate= sepdate - supplygap if supplygap < dayssupply & supplygap > 0

bysort ppn admdate (supplydate) : gen covered = supplydate+dayssupply if _n == 1
format covered %td

forval i = 2/30 {
bysort ppn admdate (supplydate) : replace covered = max(covered[_n-1]+dayssupply,supplydate+dayssupp
> ly) if _n == `i´
}
replace covered = enddate if covered >= enddate

gen covered1 =0
bysort ppn admdate (supplydate) : replace covered1 = 1 if supplydate <= covered[_n-1]

*check first values to ensure its not looking at the wrong line or if the first covered date is miss
> ing

bysort ppn admdate (supplydate) : gen daysmissed = supplydate[_n+1] - (covered+1) if covered1[_n+1]
> == 0

bysort ppn admdate (supplydate) : replace daysmissed = (enddate - covered) if covered1[_n+1] == .

replace daysmissed = 0 if covered < sepdate & daysmissed < study_days

count if supplydate >= sepdate+365

count if supplydate >= deathdate

*Take the last dispensing and substract the difference from enddate to be able to recalculate numera
> tor for PDC

bysort ppn admdate (supplydate) : egen total_days_missed = sum(daysmissed)

count if total_days_missed >= 365

replace total_days_missed = 365 if total_days_missed >= 365

bysort ppn admdate (supplydate) : keep if _n == 1

gen pre_supply_check = 0
```

```
replace pre_supply_check = 1 if (supplydate-1) > sepdate
gen pre_supply_missed = 0
replace pre_supply_missed = supplydate-sepdate-1 if pre_supply_check ==1

replace total_days_missed = total_days_missed + pre_supply_missed

gen days_covered = study_days - total_days_missed
replace days_covered = 0 if days_covered <0

gen PDC_365 = days_covered / 365

gen PDC_adj = days_covered / study_days

count if PDC_365 != PDC_adj & study_days == 365

save PDC_p2y12, replace
```

6.4.4 Statins

```
use MI_drug_class_3, clear
drop if deathdate <= sepdate + 90

*Firstly, our start date for each MI will be the sepdate
*Secondly, our end date is sepdate + 365
drop if supplydate <= sepdate - 60
drop if supplydate >= admdate & supplydate < sepdate
gen enddate = sepdate +365
format enddate %td
*We know need to correct study period end dates for participants who died prior to the follow up per
> iod.
replace enddate = deathdate if deathdate <= enddate
drop if supplydate >= deathdate
br
*We now calculate number of days in the study period
gen study_days = enddate-sepdate
*Merge in packsize data from PBS
merge m:1 pbs_item_code using PBS_packsizes
keep if _merge == 3
drop _merge
replace packsize = 30 if packsize == 1
ta packsize
bysort ppn admdate (supplydate): gen nghost=_n
ta nghost
*Therefore 51 dispensings is the most dispensings for an MI admission within the follow up period
drop nghost
rename packsize dayssupply
drop pbs_item_code
gen supplygap = 0
replace supplygap = admdate - supplydate if supplydate < admdate
drop if supplygap >= dayssupply
replace supplydate= sepdate - supplygap if supplygap < dayssupply & supplygap > 0
bysort ppn admdate (supplydate) : gen covered = supplydate+dayssupply if _n == 1
format covered %td
forval i = 2/48 {
bysort ppn admdate (supplydate) : replace covered = max(covered[_n-1]+dayssupply,supplydate+dayssupp
> ly) if _n == `i´
}
replace covered = enddate if covered >= enddate
gen covered1 =0
bysort ppn admdate (supplydate) : replace covered1 = 1 if supplydate <= covered[_n-1]
*check first values to ensure its not looking at the wrong line or if the first covered date is miss
> ing
bysort ppn admdate (supplydate) : gen daysmissed = supplydate[_n+1] - (covered+1) if covered1[_n+1]
> == 0
bysort ppn admdate (supplydate) : replace daysmissed = (enddate - covered) if covered1[_n+1] == .
replace daysmissed = 0 if covered < sepdate & daysmissed < study_days
count if supplydate > sepdate+365
count if supplydate >= deathdate
*Take the last dispensing and substract the difference from enddate to be able to recalculate numera
> tor for PDC
bysort ppn admdate (supplydate) : egen total_days_missed = sum(daysmissed)
count if total_days_missed > 365
replace total_days_missed = 365 if total_days_missed >= 365
bysort ppn admdate (supplydate) : keep if _n == 1
```

```
gen pre_supply_check = 0
replace pre_supply_check = 1 if (supplydate-1) > sepdate
gen pre_supply_missed = 0
replace pre_supply_missed = supplydate-sepdate-1 if pre_supply_check ==1

replace total_days_missed = total_days_missed + pre_supply_missed

gen days_covered = study_days - total_days_missed
replace days_covered = 0 if days_covered <0

gen PDC_365 = days_covered / 365

gen PDC_adj = days_covered / study_days

count if PDC_365 != PDC_adj & study_days == 365

save PDC_statin, replace
hist PDC_adj
```

6.4.5 High intensity statins

Using the previously created hstatin} tag, we removed any dispensings of non-high instensity statins

```
use MI_drug_class_3, clear
drop if deathdate <= sepdate + 90
drop if hstatin == 0

*Firstly, our start date for each MI will be the sepdate
*Secondly, our end date is sepdate + 365
drop if supplydate <= sepdate - 60
drop if supplydate >= admdate & supplydate < sepdate
gen enddate = sepdate +365
format enddate %td

*We know need to correct study period end dates for participants who died prior to the follow up per
> iod.

replace enddate = deathdate if deathdate <= enddate
drop if supplydate >= deathdate
br
*We now calculate number of days in the study period

gen study_days = enddate-sepdate

*Merge in packsize data from PBS

merge m:1 pbs_item_code using PBS_packsizes
keep if _merge == 3
drop _merge
replace packsize = 30 if packsize == 1
ta packsize
bysort ppn admdate (supplydate): gen nghost=_n
ta nghost

*Therefore 51 dispensings is the most dispensings for an MI admission within the follow up period

drop nghost
rename packsize dayssupply
drop pbs_item_code

gen supplygap = 0
replace supplygap = admdate - supplydate if supplydate < admdate
drop if supplygap >= dayssupply
replace supplydate= sepdate - supplygap if supplygap < dayssupply & supplygap > 0

bysort ppn admdate (supplydate) : gen covered = supplydate+dayssupply if _n == 1
format covered %td

forval i = 2/48 {
bysort ppn admdate (supplydate) : replace covered = max(covered[_n-1]+dayssupply,supplydate+dayssupp
> ly) if _n == `i´
}
replace covered = enddate if covered >= enddate

gen covered1 =0
bysort ppn admdate (supplydate) : replace covered1 = 1 if supplydate <= covered[_n-1]

*check first values to ensure its not looking at the wrong line or if the first covered date is miss
> ing

bysort ppn admdate (supplydate) : gen daysmissed = supplydate[_n+1] - (covered+1) if covered1[_n+1]
> == 0

bysort ppn admdate (supplydate) : replace daysmissed = (enddate - covered) if covered1[_n+1] == .

replace daysmissed = 0 if covered < sepdate & daysmissed < study_days

count if supplydate > sepdate+365

count if supplydate >= deathdate

*Take the last dispensing and substract the difference from enddate to be able to recalculate numera
> tor for PDC

bysort ppn admdate (supplydate) : egen total_days_missed = sum(daysmissed)

count if total_days_missed > 365

replace total_days_missed = 365 if total_days_missed >= 365
```

```
bysort ppn admdate (supplydate) : keep if _n == 1

gen pre_supply_check = 0
replace pre_supply_check = 1 if (supplydate-1) > sepdate
gen pre_supply_missed = 0
replace pre_supply_missed = supplydate-sepdate-1 if pre_supply_check ==1

replace total_days_missed = total_days_missed + pre_supply_missed

gen days_covered = study_days - total_days_missed
replace days_covered = 0 if days_covered <0

gen PDC_365 = days_covered / 365

gen PDC_adj = days_covered / study_days

count if PDC_365 != PDC_adj & study_days == 365

save PDC_hstatin, replace
hist PDC_adj
```

6.4.6 ACEi or ARB

```
use MI_drug_class_6, clear
drop if deathdate <= sepdate + 90

*Firstly, our start date for each MI will be the sepdate
*Secondly, our end date is sepdate + 365
drop if supplydate <= sepdate - 60
drop if supplydate >= admdate & supplydate < sepdate
gen enddate = sepdate +365
format enddate %td

*We know need to correct study period end dates for participants who died prior to the follow up per
> iod.

replace enddate = deathdate if deathdate <= enddate
drop if supplydate >= deathdate
br
*We now calculate number of days in the study period

gen study_days = enddate-sepdate

*Merge in packsize data from PBS

merge m:1 pbs_item_code using PBS_packsizes
keep if _merge == 3
drop _merge
ta packsize
replace packsize = 30 if packsize ==90
replace packsize = 38 if packsize ==1
replace packsize = 28 if packsize == 56
bysort ppn admdate (supplydate): gen nghost=_n
ta nghost

*Therefore 45 dispensings is the most dispensings for an MI admission within the follow up period

drop nghost
rename packsize dayssupply
drop pbs_item_code

gen supplygap = 0
replace supplygap = admdate - supplydate if supplydate < admdate
drop if supplygap >= dayssupply
replace supplydate= sepdate - supplygap if supplygap < dayssupply & supplygap > 0

bysort ppn admdate (supplydate) : gen covered = supplydate+dayssupply if _n == 1
format covered %td

forval i = 2/45 {
bysort ppn admdate (supplydate) : replace covered = max(covered[_n-1]+dayssupply,supplydate+dayssupp
> ly) if _n == `i´
}
replace covered = enddate if covered >= enddate

gen covered1 =0
bysort ppn admdate (supplydate) : replace covered1 = 1 if supplydate <= covered[_n-1]

*check first values to ensure its not looking at the wrong line or if the first covered date is miss
> ing

bysort ppn admdate (supplydate) : gen daysmissed = supplydate[_n+1] - (covered+1) if covered1[_n+1]
> == 0

bysort ppn admdate (supplydate) : replace daysmissed = (enddate - covered) if covered1[_n+1] == .

count if supplydate > sepdate+365

count if supplydate >= deathdate

*Take the last dispensing and substract the difference from enddate to be able to recalculate numera
> tor for PDC

bysort ppn admdate (supplydate) : egen total_days_missed = sum(daysmissed)

count if total_days_missed >= 365

replace total_days_missed = 365 if total_days_missed >= 365

bysort ppn admdate (supplydate) : keep if _n == 1

gen pre_supply_check = 0
```

```
replace pre_supply_check = 1 if (supplydate-1) > sepdate
gen pre_supply_missed = 0
replace pre_supply_missed = supplydate-sepdate-1 if pre_supply_check ==1

replace total_days_missed = total_days_missed + pre_supply_missed

gen days_covered = study_days - total_days_missed
replace days_covered = 0 if days_covered <0

gen PDC_365 = days_covered / 365

gen PDC_adj = days_covered / study_days

count if PDC_365 != PDC_adj & study_days == 365

save PDC_ace_arb, replace
hist PDC_adj
```

6.4.7 Beta blockers

Due to differences in dosing regimens with beta blockers, a data informed approach was taken to atribute days supply from the packsizes of different beta blockers. [4]

```
use MI_drug_class_2, clear
br
ta pbs_item_code
egen betagroup = group(pbs_item_code)
ta betagroup

*Therefore 28 different item codes

bysort ppn admdate pbs_item_code (supplydate) : gen nghost= _n
bysort ppn admdate pbs_item_code (supplydate): gen fill_diff = supplydate[_n+1] - supplydate
bysort betagroup : egen p75  = pctile(fill_diff), p(75)
drop nghost fill_diff
rename p75 dayssupply
drop betagroup
save MI_drug_class_2_packsize, replace

use MI_drug_class_2_packsize, clear
drop if deathdate <= sepdate + 90

*Firstly, our start date for each MI will be the sepdate
*Secondly, our end date is sepdate + 365
drop if supplydate <= sepdate - 60
drop if supplydate >= admdate & supplydate < sepdate
gen enddate = sepdate +365
format enddate %td

*We know need to correct study period end dates for participants who died prior to the follow up per
> iod.

replace enddate = deathdate if deathdate <= enddate
drop if supplydate >= deathdate
br
*We now calculate number of days in the study period

gen study_days = enddate-sepdate
bysort ppn admdate (supplydate): gen nghost=_n
ta nghost

*Therefore 44 dispensings is the most dispensings for an MI admission within the follow up period

drop nghost
drop pbs_item_code

gen supplygap = 0
replace supplygap = admdate - supplydate if supplydate < admdate
drop if supplygap >= dayssupply
replace supplydate= sepdate - supplygap if supplygap < dayssupply & supplygap > 0

bysort ppn admdate (supplydate) : gen covered = supplydate+dayssupply if _n == 1
format covered %td

forval i = 2/35 {
bysort ppn admdate (supplydate) : replace covered = max(covered[_n-1]+dayssupply,supplydate+dayssupp
> ly) if _n == `i'
}
replace covered = enddate if covered >= enddate

gen covered1 =0
bysort ppn admdate (supplydate) : replace covered1 = 1 if supplydate <= covered[_n-1]

*check first values to ensure its not looking at the wrong line or if the first covered date is miss
> ing

bysort ppn admdate (supplydate) : gen daysmissed = supplydate[_n+1] - (covered+1) if covered1[_n+1]
> == 0

bysort ppn admdate (supplydate) : replace daysmissed = (enddate - covered) if covered1[_n+1] == .

count if supplydate > sepdate+365

count if supplydate >= deathdate

*Take the last dispensing and substract the difference from enddate to be able to recalculate numera
> tor for PDC

bysort ppn admdate (supplydate) : egen total_days_missed = sum(daysmissed)
```

```
count if total_days_missed >= 365
replace total_days_missed = 365 if total_days_missed >= 365
bysort ppn admdate (supplydate) : keep if _n == 1
gen pre_supply_check = 0
replace pre_supply_check = 1 if (supplydate-1) > sepdate
gen pre_supply_missed = 0
replace pre_supply_missed = supplydate-sepdate-1 if pre_supply_check ==1

replace total_days_missed = total_days_missed + pre_supply_missed

gen days_covered = study_days - total_days_missed
replace days_covered = 0 if days_covered <0
gen PDC_365 = days_covered / 365
gen PDC_adj = days_covered / study_days
count if PDC_365 != PDC_adj & study_days == 365

save PDC_beta, replace
hist PDC_adj
```

6.5 Previous use of anticoagulants

Use of anticoagulant therapy (NOACs, warfarin, and heparins) in the 90 days prior to NSTEMI and 30 days following discharge were included as an adjustment for intial dispensing and use of secondary prevention medications. Namely, this was to address potential underuse of P2Y12i. This used the same PBS dataset as the secondary prevention medications.

```
use "G:\Jed\PBS B", clear
br
keep if atc == "B01AA03" | substr(atc,1,5) == "B01AB" | atc == "B01AX05" | atc == "B01AE07" | atc ==
>  "B01AF01" | atc == "B01AF02"
gen oac = 1
replace oac = 2 if atc == "B01AA03" | atc == "B01AE07" | atc == "B01AF01" | atc == "B01AF02"
gen supply_oac = date(date_of_supply, "DMY")
format supply_oac %td
keep ppn supply_oac oac
save oac, replace

use MI_cohort_ndi, clear
br
quietly {
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i'
merge 1:m ppn using oac
keep if _merge==3
keep if inrange(supply_oac, admdate-90, sepdate+30)
drop _merge
save MI_oac`i', replace
}
}
clear
forval i = 1/16 {
append using MI_oac`i'
}
bysort ppn admdate (supply_oac): gen nghost = _n
forval i = 1/14 {
bysort ppn admdate (supply_oac): replace oac = 2 if oac[_n+`i'] == 2
}
keep ppn admdate oac
bysort ppn admdate : keep if _n == 1
save MI_oac, replace
```

6.6 Poly-pharmacy assignment

We reviewed all PBS dispensing data within NSTEMI admissions to extract how many different medications (by ATC) code were used in the 60 days prior to NSTEMI admission. The total number of medications was calculated by looking at how many unique ATC codes were dispensed in the 60 days prior to admission. This is not a detailed polypharmacy calculation, as it does not consider cumulative use of medications as other studies have considered when polypharmacy is a primary outome. [5]

```
use MI_cohort_ndi, clear
keep if inrange(sepdate,td(1/7/2012), td(30/6/2017))
bysort ppn : keep if _n == 1
keep ppn
save MI_cohort_ppn, replace

foreach i in A B C D G H J L M N P R S V Z {
use "G:\Jed\PBS `i´", clear
merge m:1 ppn using MI_cohort_ppn
keep if _merge == 3
drop _merge
gen supplydate = date(date_of_supply, "DMY")
format supplydate %td
keep if inrange(supplydate,td(1/7/2012), td(30/6/2017))
egen pbsgroup = group(pbs_item_code)
save poly_`i´, replace
}

foreach ii in A B C D G H J L M N P R S V Z {
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i´
merge 1:m ppn using poly_`ii´
keep if _merge==3
keep if inrange(supplydate, admdate-60, admdate-1)
bysort ppn admdate pbs_item_code (supplydate) : keep if _N > 1
drop _merge
save MI_poly_`i´_`ii´, replace
}
}
foreach ii in A B C D G H J L M N P R S V Z {
use MI_poly_1_`ii´, clear
forval i = 1/16 {
append using MI_poly_`i´_`ii´
}
bysort ppn pbs_item_code (supplydate) : gen nghost= _n
bysort ppn admdate pbs_item_code (supplydate): gen fill_diff = supplydate[_n+1] - supplydate
bysort pbsgroup : egen p75  = pctile(fill_diff), p(75)
drop nghost fill_diff
rename p75 dayssupply
keep ppn admdate supplydate dayssupply atc pbs_item_code
sort ppn admdate
save MI_poly_`ii´, replace
}
use MI_poly_A, clear
foreach ii in B C D G H J L M N P R S V Z {
append using MI_poly_`ii´
}
sort ppn admdate atc supplydate
save MI_poly, replace

use MI_poly, clear
keep if supplydate >= admdate - 60
bysort ppn admdate atc (supplydate) : gen nghost = _n
keep if nghost == 1
bysort ppn admdate (atc) : gen polypharm = _N
bysort ppn admdate : keep if _n == 1
keep ppn admdate polypharm
su polypharm, detail
```

```
hist polypharm, color(black) graphregion(color(white)) title(Number of medications prior to MI) xtit
> le(Number of medications)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_polypharm", replace
save MI_poly_all, replace
```
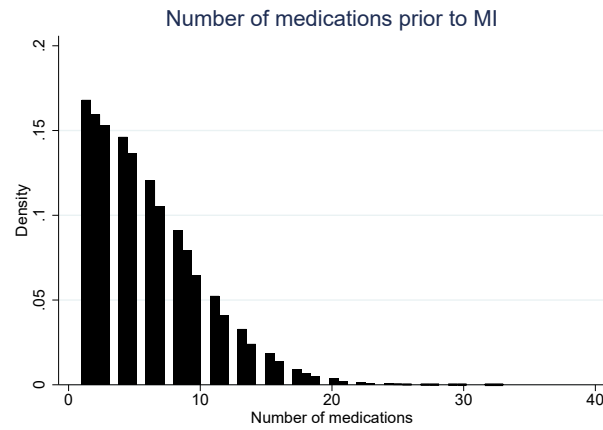
Figure 6.1: Histogram of number of regular medications pre-NSTEMI admission

7 Demographic assignment

7.1 NSTEMI classification and co-morbidity assignment

This cohort also included STEMI cases, which will not be included in this analysis due to them often requiring rescue PCI at the time of the MI, and therefore will have a disproportionate number of PCI vs CABG cases.[6] We assign comorbidity information based on ICD codes present within each index MI admission, including whether the MI was for STEMI or NSTEMI.

We were able to use the comorbidities to describe the cohort, as well as compute frailty via Charlston Comoboridity Index (CCI)[7], and use comorbidity and frailty data for adjustment in the regression analysis.

```
use MI_cohort_ndi, clear
bysort ppn (admdate) : gen nghost=_n
ta nghost
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i´
keep ppn admdate sepdate
save leanMI_`i´, replace
}
forval i=1/16 {
use leanMI_`i´, clear
*Need to rename admdate and sepdate because the merge will not allow variables of the same name
rename (admdate sepdate) (lean_admdate lean_sepdate)
merge 1:m ppn using MI_cohort_nest_tf_free
keep if _merge==3
drop _merge
keep if (admdate >= lean_admdate & sepdate <= lean_sepdate) | (sepdate==lean_sepdate) | (admdate==le
> an_admdate)

*Generate NSTEMI tag (with NSTEMI = 0 meaning STEMI and will be dropped following merge of all lean
> MI data).
gen NSTEMI = 1
replace NSTEMI = 0 if tdiag1 == "I210" | tdiag1 == "I211" | tdiag1 == "I212" | tdiag1 == "I213"

*Cardiovascular comorbidities
gen DM = 0
gen HT = 0
gen AF = 0
gen AFL = 0

*Charlson comorbidities
gen MIC = 0
gen CHF = 0
gen PVD = 0
gen CBD = 0
gen CPD = 0
gen RHEUD = 0
gen PUD = 0
gen MLD = 0
gen DMNC = 0
gen DMWC = 0
gen HOP = 0
gen REND = 0
gen MAL = 0
gen SLD = 0
gen MST = 0
gen HIV = 0

forval ii =1/40 {
replace DM = 1 if inrange(tdiag`ii´,"E10","E1499")
replace HT = 1 if inrange(tdiag`ii´,"I10","I1599") | substr(tdiag`ii´,1,4) == "U823"
replace AF = 1 if inrange(tdiag`ii´, "I48","I4899")
replace AFL =1 if substr(tdiag`ii´,1,4) == "I482" | substr(tdiag`ii´,1,4) == "I483" |substr(tdiag`ii
> ´,1,4) == "I484"
```

```
replace MIC = 1 if substr(tdiag`ii´,1,3) == "I21" | substr(tdiag`ii´,1,3) == "I22" | tdiag`ii´ == "I
> 252"
replace CHF = 1 if tdiag`ii´ == "I099" | tdiag`ii´ == "I110" | tdiag`ii´ == "I130" | tdiag`ii´ == "I
> 132" | tdiag`ii´ == "I255" | tdiag`ii´ == "I420" | inrange(tdiag`ii´,"I425","I4299") | substr(tdia
> g`ii´,1,3) == "I43" | substr(tdiag`ii´,1,3) == "I50" | tdiag`ii´ == "P290"
replace PVD = 1 if substr(tdiag`ii´,1,3) == "I70" | substr(tdiag`ii´,1,3) == "I71" | tdiag`ii´ == "I
> 731" | tdiag`ii´ == "I738" | tdiag`ii´ == "I739" | tdiag`ii´ == "I771" | tdiag`ii´ == "I790" | tdi
> ag`ii´ == "I792" | tdiag`ii´ == "K551" | tdiag`ii´ == "K558" | tdiag`ii´ == "K559" | tdiag`ii´ ==
> "Z958" | tdiag`ii´ == "Z959"
replace CBD = 1 if substr(tdiag`ii´,1,3) == "G45" | substr(tdiag`ii´,1,3) == "G46" | tdiag`ii´ == "H
> 340" | inrange(tdiag`ii´,"I60","I699")
replace CPD = 1 if tdiag`ii´ == "I278" | tdiag`ii´ == "I279" | inrange(tdiag`ii´,"J40","J4799") | in
> range(tdiag`ii´,"J60","J6799") | tdiag`ii´ == "J684" | tdiag`ii´ == "J701" | tdiag`ii´ == "J703"
replace RHEUD = 1 if substr(tdiag`ii´,1,3) == "M05" | substr(tdiag`ii´,1,3) == "M06" | tdiag`ii´ ==
> "M315" | inrange(tdiag`ii´,"M32","M3499") | tdiag`ii´ == "M351" | tdiag`ii´ == "M353" | tdiag`ii´
> == "M360"
replace PUD = 1 if inrange(tdiag`ii´,"K25","K2899")
replace MLD = 1 if substr(tdiag`ii´,1,3) == "B18" | inrange(tdiag`ii´,"K70","K7039") | tdiag`ii´ ==
> "K709" | inrange(tdiag`ii´,"K713","K7159") | tdiag`ii´ == "K717" | substr(tdiag`ii´,1,3) == "K73"
> | substr(tdiag`ii´,1,3) == "K74" | tdiag`ii´ == "K760" | inrange(tdiag`ii´,"K762","K7649") | tdiag
> `ii´ == "K768" | tdiag`ii´ == "K769" | tdiag`ii´ == "Z944"
replace DMNC = 1 if tdiag`ii´ == "E100" | tdiag`ii´ == "E101" | tdiag`ii´ == "E106" | tdiag`ii´ == "
> E108" | tdiag`ii´ == "E109" | tdiag`ii´ == "E110" | tdiag`ii´ == "E111" | tdiag`ii´ == "E116" | td
> iag`ii´ == "E118" | tdiag`ii´ == "E119" | tdiag`ii´ == "E120" | tdiag`ii´ == "E121" | tdiag`ii´ ==
> "E126" | tdiag`ii´ == "E128" | tdiag`ii´ == "E129" | tdiag`ii´ == "E130" | tdiag`ii´ == "E131" |
> tdiag`ii´ == "E136" | tdiag`ii´ == "E138" | tdiag`ii´ == "E139" | tdiag`ii´ == "E140" | tdiag`ii´
> == "E141" | tdiag`ii´ == "E146" | tdiag`ii´ == "E148" | tdiag`ii´ == "E149"
replace DMWC = 1 if inrange(tdiag`ii´,"E102","E1059") | tdiag`ii´ == "E107" | inrange(tdiag`ii´,"E11
> 2","E1159") | tdiag`ii´ == "E117" | inrange(tdiag`ii´,"E122","E1259") | tdiag`ii´ == "E127" | inra
> nge(tdiag`ii´,"E132","E1359") | tdiag`ii´ == "E137" | inrange(tdiag`ii´,"E142","E1459") | tdiag`ii
> ´ == "E147"
replace HOP = 1 if tdiag`ii´ == "G041" | tdiag`ii´ == "G114" | tdiag`ii´ == "G801" | tdiag`ii´ == "G
> 802" | substr(tdiag`ii´,1,3) == "G81" | substr(tdiag`ii´,1,3) == "G82" | inrange(tdiag`ii´,"G83","
> G8349") | tdiag`ii´ == "G839"
replace REND = 1 if tdiag`ii´ == "I120" | tdiag`ii´ == "I131" | inrange(tdiag`ii´,"N032","N0379") |
> inrange(tdiag`ii´,"N052","N0579") | substr(tdiag`ii´,1,3) == "N18" | substr(tdiag`ii´,1,3) == "N19
> " | tdiag`ii´ == "N250" | inrange(tdiag`ii´,"Z49","Z4929") | tdiag`ii´ == "Z940" | tdiag`ii´ == "Z
> 992"
replace MAL = 1 if inrange(tdiag`ii´,"C00","C2699") | inrange(tdiag`ii´,"C30","C3499") | inrange(tdi
> ag`ii´,"C37","C4199") | substr(tdiag`ii´,1,3) == "C43" | inrange(tdiag`ii´,"C45","C5899") | inrang
> e(tdiag`ii´,"C60","C7699") | inrange(tdiag`ii´,"C81","C8599") | substr(tdiag`ii´,1,3) == "C88" | i
> nrange(tdiag`ii´,"C90","C9799")
replace SLD = 1 if tdiag`ii´ == "I850" | tdiag`ii´ == "I859" | tdiag`ii´ == "I864" | tdiag`ii´ == "I
> 982" | tdiag`ii´ == "K704" | tdiag`ii´ == "K711" | tdiag`ii´ == "K721" | tdiag`ii´ == "K729" | tdi
> ag`ii´ == "K765" | tdiag`ii´ == "K766" | tdiag`ii´ == "K767"
replace MST = 1 if inrange(tdiag`ii´,"C77","C8099")
replace HIV = 1 if inrange(tdiag`ii´,"B20","B2299") | substr(tdiag`ii´,1,3) == "B24"
}
collapse (sum) NSTEMI-HIV, by(ppn lean_admdate)
foreach var of varlist NSTEMI-HIV {
replace `var´ = 1 if `var´ > 1
}
rename lean_admdate admdate
save MIcomorbid_`i´, replace
}
forval i=1/16 {
append using MIcomorbid_`i´
}
bysort ppn admdate : keep if _n == 1
save MIcomorbidtotal, replace
```

7.2 Remoteness assignment

We used previously reported methods to assign a value of remoteness using the statistical local area of each indivudal at the time of their index MI. This remoteness value is mean Accessibility/Remoteness Index of Australia.[8]

In the previous study that focussed on remoteness, our protocol detailed that those MI admissions with missing SLAs and therefore matching ARIA values would be exlcuded. However, for this study, we instead assigned the 3,601 MI admissions without a SLA the median ARIA value for the remainder of the Victorian cohort within the study period.

```
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_ndi_location.dta", clear
keep ppn admdate sla state

merge 1:1 ppn admdate using MI_cohort_ndi
drop _merge

keep if state == 2
keep if inrange(sepdate,td(1/7/2012),td(30/6/2017))

merge m:1 sla using "G:\Adam\Project 2  - location and MI outcomes\Data\sla_location_key.dta"
drop if _merge == 2
drop SLAname Min Max RRMAClass _merge

egen median_ARIA  = pctile(Mean), p(50)

replace Mean = median_ARIA if Mean ==.

rename Mean mean_ARIA

keep ppn admdate state ARIA mean_ARIA

save MIremoteness, replace
```

7.3 Socioeconomic disadvantage assignment

The IRSD score for the postcode where the indiviudal was residing at the time closest to their MI was used for this step. Obtaining postcode information and linking to a dataset containing the corresponding IRSD value has been reported in the previous protocol. This dataset has already excluded people outside of the study period, but does include non-Victorians. These MI admissions will be exlcuded during the preparation of the analysis dataset. There were originally 2,146 admissions missing postcode data, and the mean IRSD was assigned to those with missing postcode. In this cohort, there were only 1,748 admissions from the original dataset with missing postcode data that were present in this dataset.

```
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_SES.dta", clear

merge 1:1 ppn admdate using MI_cohort_ndi

keep if _merge == 3

keep ppn admdate IRSD_score

save MIses, replace
```

8 Clinical outcomes at 12 months

Using the NDI and VAED datasets, we determined the following clinical outcomes:

- All-cause death

- Cardiovascular death

- Recurrent MI

- Stroke

- Revascularisation

8.1 All-cause death

Information on any reported deaths is already present following the merge with the NDI dataset. This date (if present) will be used for all-cause death.

8.2 Cardiovascular death

This step involved merging a previous dataset containing the MI cohort from a previous study protocol. black) This data was sourced from the NDI with a cardiovascular primary cause of death allowing for identifcation of cardiovascular death.

```
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_NDI_cause.dta", clear
br
keep ppn admdate sepdate underlying_cause_of_death othercauses deathdate
ta underlying_cause_of_death
gen cvdeath_tag = 0
replace cvdeath_tag = 1 if inrange(underlying_cause_of_death,"I10","I99")
gen cvdeath = deathdate if cvdeath_tag == 1
format cvdeath %td
keep ppn admdate sepdate cvdeath cvdeath_tag
keep if cvdeath_tag == 1
gen cvdeath_12mo = 0
replace cvdeath_12mo = 1 if cvdeath <= sepdate + 365
drop sepdate
hist cvdeath,  bin(100) color(black) graphregion(color(white)) frequency title(Admissions with cardi
> ovasular primary cause of death from MI cohort)  xtitle(Cardiovascular death)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_cvd", replace
save outcome_cvdeath, replace
```
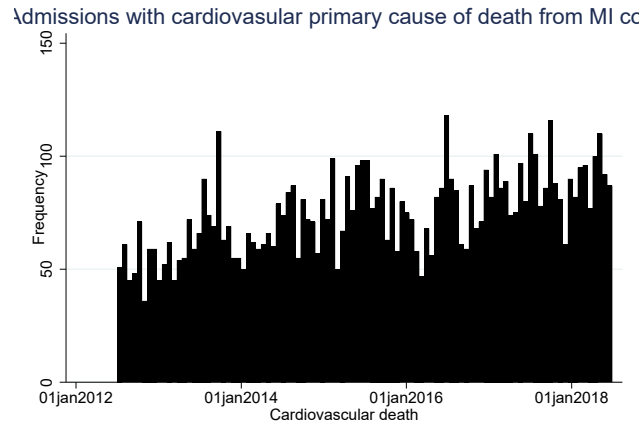
Figure 8.1: Histogram of cardiovascular deaths following NSTEMI

8.3 Stroke admission

```
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_raw.dta", clear
br
keep ppn admdate sepdate tdiag1
gen stroke_tag = 0
replace stroke_tag = 1 if inrange(tdiag1, "I60","I698")
ta stroke_tag
gen stroke = admdate if stroke_tag==1
format stroke %td
drop tdiag1
keep if stroke_tag == 1
hist stroke,  bin(100) color(black) graphregion(color(white)) frequency xtitle(Stroke admissions)  t
> itle(Admissions with primary diagnosis of stroke from MI cohort)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_strokeadm", replace
save stroke, replace
```
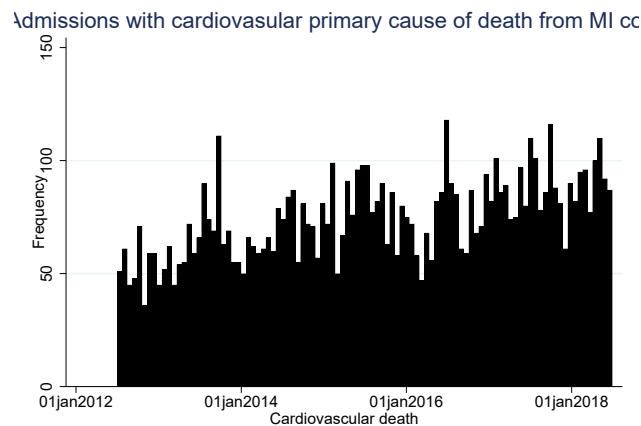

Figure 8.2: Histogram of cardiovascular deaths following NSTEMI

35

```
quietly {
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i´
merge 1:m ppn using stroke
keep if _merge==3
keep ppn admdate sepdate stroke stroke_tag
drop if stroke <= sepdate
save stroke_merge`i´, replace
}
}
clear
forval i = 1/16 {
append using stroke_merge`i´
}
br
count if stroke < sepdate
bysort ppn admdate (stroke) : keep if _n==1
gen stroke_12mo = 0
replace stroke_12mo = 1 if stroke <= sepdate + 365
drop sepdate
save outcome_stroke, replace
```

8.4   recurrent MI admission

```
use MI_cohort_ndi, clear
br
keep ppn admdate sepdate
bysort ppn (admdate) : gen recurrentMI = admdate[_n+1]
format recurrentMI %td
keep if recurrentMI !=.
gen recurrentMI_tag = 1
gen recurrentMI_12mo = 0
replace recurrentMI_12mo = 1 if recurrentMI <= sepdate + 365
drop sepdate
order ppn admdate recurrentMI_tag
hist recurrentMI,  bin(100) color(black) graphregion(color(white)) frequency xtitle(Recurrent MI adm
> issions)  title(Admissions of recurrent MI from MI cohort)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_recurrentmi", replace
save outcome_recurrentMI, replace
```
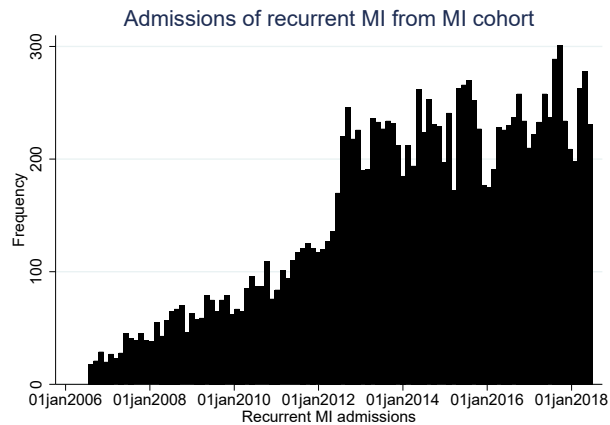

Figure 8.3: Histogram of recurrent myocardial infarction following NSTEMI

8.5 Admission for revascularisation

```
use "G:\Adam\Project 2  - location and MI outcomes\Data\MI_cohort_raw.dta", clear
br
keep ppn admdate sepdate toper*
gen CABG =.
gen PCI =.
order ppn admdate sepdate PCI CABG
quietly {
forval i = 1/40 {
replace CABG = sepdate if inrange(toper`i´,3849700,3849707) | inrange(toper`i´,3850000,3850305)
replace PCI = sepdate if inrange(toper`i´,3830600,3830605) | toper`i´==3830000 |toper`i´==3830300 |
> toper`i´==3830301 | inrange(toper`i´, 3830900,3831801) | inrange(toper`i´,9021800,9021802)
}
}
format CABG %td
format PCI %td
keep if PCI != . | CABG != .
drop toper* sepdate
gen revasc = PCI if PCI != .
replace revasc = CABG if CABG !=. & CABG < PCI
count if revasc == .
format revasc %td
save pci_cabg_raw, replace

quietly {
forval i = 1/16 {
use MI_cohort_ndi, clear
bysort ppn (admdate) : keep if _n == `i´
merge 1:m ppn using pci_cabg_raw
keep if _merge==3
keep ppn admdate sepdate PCI CABG revasc
drop if revasc <= sepdate
save revasc_merge`i´, replace
}
}
clear
forval i = 1/16 {
append using revasc_merge`i´
}
br
count if revasc > sepdate
bysort ppn admdate (revasc) : gen nghost = _n
keep if nghost == 1
gen revasc_tag = 1
gen revasc_12mo = 0
replace revasc_12mo = 1 if revasc <= sepdate + 365 & revasc >= sepdate + 30
drop sepdate
order ppn admdate revasc_tag
hist revasc,  bin(100) color(black) graphregion(color(white)) frequency xtitle(Revascularisation adm
> issions)  title(Admissions of revascularisation (PCI/CABG) following MI)
graph save "G:\Adam\Project - revasc and outcome prediction\Results\hist_revascadm", replace
save outcome_revasc, replace
```
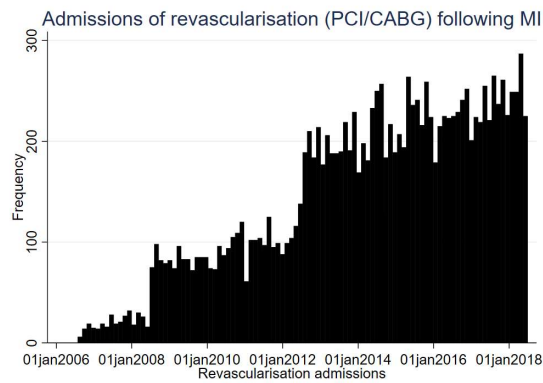
Figure 8.4: Histogram of admissions for revascularisation following NSTEMI

9 Data analysis

9.1 Creation of analysis dataset

Each of the separate datasetes throughout cohort creation, medication use and demographic information assignment were merged into one dataset. The final dataset contains information regarding all Victorian NSTEMI admissions.

```
use MI_cohort_ndi, clear
br
keep if inrange(sepdate,td(1/7/2012), td(30/6/2017))
gen death_time = deathdate - sepdate
gen hosp_mort = 0
replace hosp_mort = 1 if death_time < 1
replace hosp_mort = 2 if death_time >= 1 & death_time <=90
ta hosp_mort
drop death_time sepmode
rename priorMI prevMI
gen priorMI = 0
replace priorMI = 1 if prevMI > 0
save MI_ads0, replace

use MI_proc_all, clear
save MI_ads1, replace

use MI_drug_prior, clear
save MI_ads2, replace

use MI_drug_received, clear
save MI_ads3, replace

use PDC_p2y12, clear
keep ppn admdate PDC_adj
rename PDC_adj PDC_p2y12
save MI_ads4, replace

use PDC_statin, clear
keep ppn admdate PDC_adj
rename PDC_adj PDC_statin
save MI_ads5, replace

use PDC_hstatin, clear
keep ppn admdate PDC_adj
rename PDC_adj PDC_hstatin
save MI_ads6, replace

use PDC_ace_arb, clear
```

```
keep ppn admdate PDC_adj
rename PDC_adj PDC_ace_arb
save MI_ads7, replace

use PDC_beta, clear
keep ppn admdate PDC_adj
rename PDC_adj PDC_beta
save MI_ads8, replace

use MIcomorbidtotal, clear
save MI_ads9, replace

use MIremoteness, clear
save MI_ads10, replace

use MIses, clear
save MI_ads11, replace

use outcome_cvdeath, clear
save MI_ads12, replace

use outcome_stroke, clear
save MI_ads13, replace

use outcome_recurrentMI, clear
save MI_ads14, replace

use outcome_revasc, clear
save MI_ads15, replace

use MI_oac, clear
gen anticoag = 1
save MI_ads16, replace

use MI_poly_all, clear
save MI_ads17, replace

use MI_ads0, clear
br
forval i = 1/17 {
merge 1:1 ppn admdate using MI_ads`i´
drop if _merge == 2
drop _merge
}
br
*create age group stratification
replace agegroup = substr(agegroup,1,2)
destring agegroup, replace force
ta agegroup
*create death tags at 90 days and 365 days
gen dead90 = 0
replace dead90 = 1 if deathdate <= sepdate + 90
gen dead_12mo = 0
replace dead_12mo = 1 if deathdate <= sepdate + 365
*remove non-Victorians
drop if state != 2
drop state
*create values for oac were data is missing
replace oac = 0 if oac == .
replace anticoag = 0 if anticoag == .
*create values for polypharm were data is missing
replace polypharm = 0 if polypharm == .
*replace PDC_adj values with 0 if alive but no dispensing
foreach i in p2y12 statin ace_arb beta hstatin {
replace PDC_`i´ = 0 if hosp_mort == 0 & PDC_`i´ ==.
}
save ads_revasc, replace
```

39

9.2 Result tables

9.3 Tables of total population characteristics

The analysis dataset contains information on STEMI as well as MI admissions where no revascularisation was performed. For the total table, we will exclude STEMI but include those without revasculrisation as a third arm to show those with no revasculisation. We created a variable to identity PCI only, CABG (with or without PCI), and no revascularisation.

```
use ads_revasc, clear
forval i = 35(10)85 {
replace agegroup = agegroup - 5 if agegroup == `i´
}
gen rid = 0
replace rid = 1 if PCI_tag == 1
replace rid = 2 if CABG_tag == 1
br
drop if NSTEMI == 0
save ads_revasc_age10, replace

use ads_revasc_age10, clear

gen tahelp = 1
ta tahelp rid,                                                    matcell(B1)
ta sex rid if sex == 1,                                  matcell(B2)
ta agegroup rid,                                                 matcell(B3)
ta priorMI rid if priorMI == 1,                   matcell(B4)
ta CBD rid if CBD== 1,                                   matcell(B5)
ta HT rid if HT== 1,                                     matcell(B6)
ta AF rid if AF== 1,                                     matcell(B7)
ta DM rid if DM== 1,                                     matcell(B8)
ta CHF rid if CHF== 1,                                   matcell(B9)
ta CPD rid if CPD == 1,                                  matcell(B10)
ta PVD rid if PVD == 1,                                  matcell(B11)
ta REND rid if REND == 1,                                matcell(B12)
ta prevPCI rid if prevPCI == 1,                   matcell(B14)
ta prevCABG rid if prevCABG == 1,                 matcell(B15)
ta hosp_mort rid if hosp_mort == 1,         matcell(B16)
ta hosp_mort rid if hosp_mort == 2,         matcell(B17)
ta ARIA rid, rowsort                                     matcell(B18)
ta p2y12_prior rid if p2y12_prior == 1 ,      matcell(B19)
ta statin_prior rid if statin_prior == 1,     matcell(B20)
ta ace_arb_prior rid if ace_arb_prior == 1, matcell(B21)
ta beta_prior rid if beta_prior == 1,         matcell(B22)
ta p2y12_60 rid if p2y12_60 == 1,                 matcell(B23)
ta statin_60 rid if statin_60 == 1,               matcell(B24)
ta hstatin_60 rid if hstatin_60 == 1,       matcell(B28)
ta ace_arb_60 rid if ace_arb_60 == 1,       matcell(B25)
ta beta_60 rid if beta_60 == 1,                   matcell(B26)
ta OMT rid,                                                  matcell(B27)

matrix B = (B1\B2\B3\B4\B5\B6\B7\B8\B9\B10\B11\B12\B14\B15\B16\B17\B18\B19\B20\B21\B22\B23\B24\B28\B
> 25\B26\B27)
matrix list B
clear
svmat B
br
rename (B1 B2 B3) (NONE PCI CABG)
egen Total = max(NONE + PCI + CABG)
gen Totalproportion = string((100 * (NONE + PCI + CABG) / Total), "%3.0f")+"%"
replace Total = NONE + PCI + CABG

foreach i in NONE PCI CABG {
egen `i´Total = max(`i´)
gen `i´proportion1 = (100 * `i´ / `i´Total)
replace `i´proportion1 = `i´Total / Total * 100 if `i´proportion1 == 100
gen `i´proportion = string(`i´proportion1, "%3.0f")+"%"
drop `i´proportion1
```

```stata
}

gen id = _n
gen name = ""
replace name = "Male" if                                                      _n == 2
replace name = "Aged 30-39" if                                          _n == 3
replace name = "Aged 40-49" if                                          _n == 4
replace name = "Aged 50-54" if                                          _n == 5
replace name = "Aged 60-64" if                                          _n == 6
replace name = "Aged 70-74" if                                          _n == 7
replace name = "Aged 80+" if                                            _n == 8
replace name = "Previous myocardial infarction" if              _n == 9
replace name = "Previous stroke" if                                _n == 10
replace name = "Hypertension" if                                    _n == 11
replace name = "Atrial fibrillation" if                           _n == 12
replace name = "Diabetes" if                                        _n == 13
replace name = "Heart failure" if                                   _n == 14
replace name = "Chronic pulmonary disease" if             _n == 15
replace name = "Peripeheral vascular disease" if          _n == 16
replace name = "Renal disease" if                                   _n == 17
replace name = "Previous PCI" if                                    _n == 18
replace name = "Previous CABG" if                                   _n == 19
replace name = "Died during admission" if                    _n == 20
replace name = "Died within 90 days of discharge" if    _n == 21
replace name = "Metropolitan" if                                    _n == 22
replace name = "Inner regional" if                                  _n == 23
replace name = "Outer regional" if                                  _n == 24
replace name = "Prior P2Y12 inhibitor" if                 _n == 25
replace name = "Prior Statin" if                                    _n == 26
replace name = "Prior ACE inhibitor or ARB" if           _n == 27
replace name = "Prior beta blocker"  if                     _n == 28
replace name = "P2Y12 inhibitor"  if                        _n == 29
replace name = "Statin" if                                                  _n == 30
replace name = "High intensity statin" if                    _n == 31
replace name = "ACE inhibitor or ARB" if                     _n == 32
replace name = "Beta blocker" if                                    _n == 33
replace name = "No medication classes dispensed" if      _n == 34
replace name = "One medication classes dispensed" if   _n == 35
replace name = "Two medication classes dispensed" if   _n == 36
replace name = "Three medication classes dispensed" if  _n == 37
replace name = "Four medication classes dispensed"  if  _n == 38

drop NONETotal PCITotal CABGTotal
order name Total Totalproportion NONE* PCI* CABG*
save Totaldemo1, replace

*Create polypharmacy table data
use ads_revasc_age10, clear
su(polypharm), detail
matrix B = (r(p50)\r(p25)\r(p75))
su(polypharm) if rid == 0, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(polypharm) if rid == 1, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(polypharm) if rid == 2, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))

mat list B
clear
svmat B
foreach i in B1 B2 B3 B4  {
gen `i´s = string(`i´)
drop `i´
gen `i´ = `i´ + " (" + `i´[_n+1] + "-" + `i´[_n+2] + ")"
drop `i´s
}
drop if _n != 1
rename (B1 B2 B3 B4) (Total NONE PCI CABG)
gen name = "Median number of medications (IQR)"
order name
```

```
save polytot, replace

*Create IRSD table data
use ads_revasc_age10, clear
su(IRSD_score), detail
matrix B = (r(p50)\r(p25)\r(p75))
su(IRSD_score) if rid == 0, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(IRSD_score) if rid == 1, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(IRSD_score) if rid == 2, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))

mat list B
clear
svmat B
foreach i in B1 B2 B3 B4  {
gen `i´s = string(`i´)
drop `i´
gen `i´ = `i´ + " (" + `i´[_n+1] + "-" + `i´[_n+2] + ")"
drop `i´s
}
drop if _n != 1
rename (B1 B2 B3 B4) (Total NONE PCI CABG)
gen name = "Median IRSD score (IQR)"
order name
save irsdtot, replace

*add in median IRSD data and summary labels
use Totaldemo1, clear
foreach i in Total NONE PCI CABG {
gen `i´s = string(`i´)
drop `i´
rename `i´s `i´
}
foreach i in Total NONE PCI CABG {
gen `i´s = `i´ + " " + "(" + `i´proportion + ")"
drop `i´ `i´proportion
rename `i´s `i´
}
append using irsdtot
append using polytot
replace id = 24.5 if id == . & name == "Median IRSD score (IQR)"
replace id = 28.5 if id == .
sort id
drop id
gen category = ""
replace category = "Baseline characteristics" if _n == 2
replace category = "Co-morbidities" if _n == 9
replace category = "Mortality" if _n == 20
replace category = "Remoteness and socio-economic status" if _n == 22
replace category = "Medications dispensed within 90 days prior to MI admission" if _n == 26
replace category = "Medications dispensed within 60 days following MI admission" if _n == 31
order category
save table_totaldemo, replace
use table_totaldemo, clear
```

9.4 Tables of analysed population characteristics

```
use ads_revasc_age10, clear
drop if rid == 0
keep if hosp_mort == 0

gen tahelp = 1
ta tahelp rid,                                                    matcell(B1)
ta sex rid if sex == 1,                              matcell(B2)
ta agegroup rid,                                              matcell(B3)
ta priorMI rid if priorMI == 1,                      matcell(B4)
ta CBD rid if CBD== 1,                               matcell(B5)
ta HT rid if HT== 1,                                 matcell(B6)
ta AF rid if AF== 1,                                 matcell(B7)
ta DM rid if DM== 1,                                 matcell(B8)
ta CHF rid if CHF== 1,                               matcell(B9)
ta CPD rid if CPD == 1,                              matcell(B10)
ta PVD rid if PVD == 1,                              matcell(B11)
ta REND rid if REND == 1,                            matcell(B12)
ta prevPCI rid if prevPCI == 1,              matcell(B14)
ta prevCABG rid if prevCABG == 1,            matcell(B15)
ta hosp_mort rid if hosp_mort == 1,        matcell(B16)
ta hosp_mort rid if hosp_mort == 2,        matcell(B17)
ta ARIA rid, rowsort                                 matcell(B18)
ta p2y12_prior rid if p2y12_prior == 1 ,     matcell(B19)
ta statin_prior rid if statin_prior == 1,    matcell(B20)
ta ace_arb_prior rid if ace_arb_prior == 1, matcell(B21)
ta beta_prior rid if beta_prior == 1,        matcell(B22)
ta p2y12_60 rid if p2y12_60 == 1,                    matcell(B23)
ta statin_60 rid if statin_60 == 1,                  matcell(B24)
ta hstatin_60 rid if hstatin_60 == 1,        matcell(B28)
ta ace_arb_60 rid if ace_arb_60 == 1,        matcell(B25)
ta beta_60 rid if beta_60 == 1,                      matcell(B26)
ta OMT rid,                                          matcell(B27)

matrix B = (B1\B2\B3\B4\B5\B6\B7\B8\B9\B10\B11\B12\B14\B15\B18\B19\B20\B21\B22\B23\B24\B28\B25\B26\B
> 27)
matrix list B
clear
svmat B
br

rename (B1 B2) (PCI CABG)
egen Total = max(PCI + CABG)
gen Totalproportion = string((100 * (PCI + CABG) / Total), "%3.0f")+"%"
replace Total = PCI + CABG

foreach i in PCI CABG {
egen `i'Total = max(`i')
gen `i'proportion1 = (100 * `i' / `i'Total)
replace `i'proportion1 = `i'Total / Total * 100 if `i'proportion1 == 100
gen `i'proportion = string(`i'proportion1, "%3.0f")+"%"
drop `i'proportion1
}

gen id = _n
gen name = ""
replace name = "Male" if                                          _n == 2
replace name = "Aged 30-39" if                                _n == 3
replace name = "Aged 40-49" if                                _n == 4
replace name = "Aged 50-54" if                                _n == 5
replace name = "Aged 60-64" if                                _n == 6
replace name = "Aged 70-74" if                                _n == 7
replace name = "Aged 80+" if                                  _n == 8
replace name = "Previous myocardial infarction" if         _n == 9
replace name = "Previous stroke" if                          _n == 10
replace name = "Hypertension" if                             _n == 11
replace name = "Atrial fibrillation" if                  _n == 12
replace name = "Diabetes" if                                 _n == 13
replace name = "Heart failure" if                            _n == 14
```

```
replace name = "Chronic pulmonary disease" if                    _n == 15
replace name = "Peripeheral vascular disease" if                 _n == 16
replace name = "Renal disease" if                                      _n == 17
replace name = "Previous PCI" if                                       _n == 18
replace name = "Previous CABG" if                                      _n == 19
replace name = "Metropolitan" if                                       _n == 20
replace name = "Inner regional" if                                     _n == 21
replace name = "Outer regional" if                                     _n == 22
replace name = "Prior P2Y12 inhibitor" if                        _n == 23
replace name = "Prior Statin" if                                     _n == 24
replace name = "Prior ACE inhibitor or ARB" if              _n == 25
replace name = "Prior beta blocker"  if                          _n == 26
replace name = "P2Y12 inhibitor"  if                             _n == 27
replace name = "Statin" if                                                   _n == 28
replace name = "High intensity statin" if                      _n == 29
replace name = "ACE inhibitor or ARB" if                       _n == 30
replace name = "Beta blocker" if                                     _n == 31
replace name = "No medication classes dispensed" if        _n == 32
replace name = "One medication classes dispensed" if   _n == 33
replace name = "Two medication classes dispensed" if   _n == 34
replace name = "Three medication classes dispensed" if _n == 35
replace name = "Four medication classes dispensed"  if _n == 36

drop PCITotal CABGTotal
order name Total Totalproportion PCI* CABG*
save Analdemo1, replace

*Create polypharmacy table data
use ads_revasc_age10, clear
su(polypharm), detail
matrix B = (r(p50)\r(p25)\r(p75))
su(polypharm) if rid == 1, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(polypharm) if rid == 2, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))

mat list B
clear
svmat B
foreach i in B1 B2 B3 {
gen `i´s = string(`i´)
drop `i´
gen `i´ = `i´ + " (" + `i´[_n+1] + "-" + `i´[_n+2] + ")"
drop `i´s
}
drop if _n != 1
rename (B1 B2 B3) (Total PCI CABG)
gen name = "Median number of medications (IQR)"
order name
save polyanal, replace

*Create IRSD table data
use ads_revasc_age10, clear
su(IRSD_score), detail
matrix B = (r(p50)\r(p25)\r(p75))
su(IRSD_score) if rid == 1, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))
su(IRSD_score) if rid == 2, detail
matrix B = (B,(r(p50)\r(p25)\r(p75)))

mat list B
clear
svmat B
foreach i in B1 B2 B3 {
gen `i´s = string(`i´)
drop `i´
gen `i´ = `i´ + " (" + `i´[_n+1] + "-" + `i´[_n+2] + ")"
drop `i´s
}
drop if _n != 1
```

```
rename (B1 B2 B3) (Total PCI CABG)
gen name = "Median IRSD score (IQR)"
order name
save irsdanal, replace

*add in median IRSD data and summary labels
use Analdemo1, clear
foreach i in Total PCI CABG {
gen `i´s = string(`i´)
drop `i´
rename `i´s `i´
}
foreach i in Total PCI CABG {
gen `i´s = `i´ + " " + "(" + `i´proportion + ")"
drop `i´ `i´proportion
rename `i´s `i´
}
append using irsdanal
append using polyanal
replace id = 22.5 if id == .
replace id = 26.5 if id == 22.5 & name == "Median number of medications (IQR)"
sort id
drop id
gen category = ""
replace category = "Baseline characteristics" if _n == 2
replace category = "Co-morbidities" if _n == 9
replace category = "Remoteness and socio-economic status" if _n == 20
replace category = "Medications dispensed within 90 days prior to MI admission" if _n == 24
replace category = "Medications dispensed within 60 days following MI admission" if _n == 29
order category
save table_analdemo, replace
use table_analdemo, clear
```

9.5 Tables of clinical outcomes at 12 months for analysed cohort

```
use ads_revasc_age10, clear
keep if rid != 0

gen tahelp = 1
ta tahelp rid,                                                          matcell(B1)
ta dead_12mo rid if dead_12mo == 1,                           matcell(B2)
ta cvdeath_12mo rid if cvdeath_12mo ==1,                 matcell(B3)
ta recurrentMI_12mo rid if recurrentMI_12mo == 1,     matcell(B4)
ta stroke_12mo rid if stroke_12mo == 1,                  matcell(B5)
ta revasc_12mo rid if revasc_12mo == 1,                  matcell(B6)

matrix B = (B1\B2\B3\B4\B5\B6)
matrix list B
clear
svmat B
br

rename (B1 B2) (PCI CABG)
egen Total = max(PCI + CABG)
gen Totalproportion = string((100 * (PCI + CABG) / Total), "%3.0f")+"%"
replace Total = PCI + CABG

foreach i in PCI CABG {
egen `i´Total = max(`i´)
gen `i´proportion1 = (100 * `i´ / `i´Total)
replace `i´proportion1 = `i´Total / Total * 100 if `i´proportion1 == 100
gen `i´proportion = string(`i´proportion1, "%3.0f")+"%"
drop `i´proportion1
}

gen id = _n
gen name = ""
replace name = "All-cause death" if                              _n == 2
replace name = "Cardiovascular death" if                 _n == 3
replace name = "Recurrent MI" if                             _n == 4
replace name = "Stroke" if                                        _n == 5
replace name = "Revascularisation" if                    _n == 6

foreach i in Total PCI CABG {
gen `i´s = string(`i´)
drop `i´
rename `i´s `i´
}
foreach i in Total PCI CABG {
gen `i´s = `i´ + " " + "(" + `i´proportion + ")"
drop `i´ `i´proportion
rename `i´s `i´
}

order name Total PCI CABG
drop PCITotal CABGTotal id

save table_outcomes_12mo, replace
use table_outcomes_12mo, clear
```

9.6 Tables of clinical outcomes at 12 months for analysed cohort

```
use ads_revasc_age10, clear
keep if rid != 0
gen dead_tag = 0
replace dead_tag = 1 if deathdate != .

gen tahelp = 1
ta tahelp rid,                                                          matc
> ell(B1)
ta dead_tag rid if dead_tag == 1,                            matcell(B2)
ta cvdeath_tag rid if cvdeath_tag ==1,                  matcell(B3)
ta recurrentMI_tag rid if recurrentMI_tag == 1,     matcell(B4)
```

```stata
ta stroke_tag rid if stroke_tag == 1,                                   matcell(B5)
ta revasc_tag rid if revasc_tag == 1,                                   matcell(B6)

matrix B = (B1\B2\B3\B4\B5\B6)
matrix list B
clear
svmat B
br

rename (B1 B2) (PCI CABG)
egen Total = max(PCI + CABG)
gen Totalproportion = string((100 * (PCI + CABG) / Total), "%3.0f")+"%"
replace Total = PCI + CABG

foreach i in PCI CABG {
egen `i´Total = max(`i´)
gen `i´proportion1 = (100 * `i´ / `i´Total)
replace `i´proportion1 = `i´Total / Total * 100 if `i´proportion1 == 100
gen `i´proportion = string(`i´proportion1, "%3.0f")+"%"
drop `i´proportion1
}

gen id = _n
gen name = ""
replace name = "All-cause death" if                                     _n == 2
replace name = "Cardiovascular death" if                              _n == 3
replace name = "Recurrent MI" if                                       _n == 4
replace name = "Stroke" if                                                _n == 5
replace name = "Revascularisation" if                              _n == 6

foreach i in Total PCI CABG {
gen `i´s = string(`i´)
drop `i´
rename `i´s `i´
}
foreach i in Total PCI CABG {
gen `i´s = `i´ + " " + "(" + `i´proportion + ")"
drop `i´ `i´proportion
rename `i´s `i´
}

order name Total PCI CABG
drop PCITotal CABGTotal id

save table_outcomes_all, replace
use table_outcomes_all, clear
```

9.7 Predicted probabilty of dispensing at 60 days at means of co-variates stratified by revascularisation

We predicted probabilities of dispensing for each medication class using logistic regression, stratified by reveascularisation strategy. We adjusted with mean values of the spline effects of age, ARIA and IRSD, and binary effects of CBD, HT, AF, DM, CHF, CPD, REND, and dispensing of anticoagulation in the 60 days prior to admission.

```
*Full analysis
*Create age group sections to allow for age spline creation
use ads_revasc, clear
drop if hosp_mort != 0
drop if NSTEMI == 0
gen rid = 0
replace rid = 1 if PCI_tag == 1
replace rid = 2 if CABG_tag == 1
drop if rid == 0
br
gen agespline = agegroup + 2.5 if agegroup != 85
replace agespline = agegroup + 5 if agegroup == 85
save ads_revasc_spline, replace

*Creation of prediction data set, using marginal effects at the means of independent variables for P
> CI and CABG. These are located in the appendices of the manuscript.
forval ii = 1/2 {
use ads_revasc_spline, clear
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND IRSD_score mean_ARIA anticoag polyp
> harm p2y12_60 statin_60 hstatin_60 ace_arb_60 beta_60 {
su(`i´) if rid == `ii´
local m`i´ = r(mean)
}
clear
set obs 1
gen rid = `ii´
br
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND IRSD_score mean_ARIA anticoag polyp
> harm p2y12_60 statin_60 hstatin_60 ace_arb_60 beta_60 {
gen `i´ = `m`i´´
}
mkspline ARIAS= mean_ARIA, cubic knots(0 0.05 0.5 2)
mkspline ages = agespline, cubic knots(32.5 62.5 72.5 77.5)
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)
save analysis_adherence_set_`ii´, replace
}
append using analysis_adherence_set_1
gen revasc = ""
replace revasc = "PCI" if rid == 1
replace revasc = "CABG" if rid  == 2
order revasc
sort rid
save covariate_means, replace
use covariate_means, clear
*Set up splines followed by logistic regression model construction for PCI and CABG groups

forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

logistic `i´_60 c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD i.PVD
> i.REND i.anticoag c.polypharm if rid ==`ii´, coef
```

```
use analysis_adherence_set_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_60_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_1, replace

clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_2, replace

*Create table of both revasc strategy predictions
use disp_predict_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using disp_predict_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
save disp_predict, replace
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_disp_predict, replace

*Create bar graph
use disp_predict_1, clear
append using disp_predict_2
append using disp_predict_2
rename class Class
sort Class rid
gen classrid = _n
drop if classrid == 3 | classrid ==  6 | classrid ==   9 | classrid ==   12 | classrid ==   15

twoway ///
(bar A classrid if Class == 1 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 1 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 2 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 2 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 3 & rid ==1, barw(0.8) col(dknavy%90)) ///
```

```
(bar A classrid if Class == 3 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 4 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 4 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 5 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 5 & rid ==2, barw(0.8) col(red%80)) ///
(rcap ll ul classrid, col(black)), graphregion(color(white)) ///
yscale(range(0,1)) ylabel(0 "0.0"  0.2 "0.2"  0.4 "0.4"  0.6 "0.6"  0.8 "0.8"  1.0 "1.0") ytitle("Pr
> obability of dispensing within 60 days post discharge")  ///
xlabel(1.5 "P2Y12i" 4.5 "Statin" 7.5 "High intensity statin" 10.5 "ACEi/ARB" 13.5 "Beta blocker") xt
> itle("") legend(order(1 "PCI" 2 "CABG"))
graph save "G:\Adam\Project - revasc and outcome prediction\Results\fig_disp", replace
```

9.7.1 Predited dispensing of P2Y12i per year for CABG

Due to the difference noted in P2Y12 dispensing, and the introduction of P2Y12 recommendations from 2012, year on year dispesning was predicted for this class of medications in those receiving CABG.

```
use ads_revasc_spline, clear
ta rid

gen sepyear = 1
replace sepyear = 2 if spedate >= td(1/7/2013)
replace sepyear = 3 if spedate >= td(1/7/2014)
replace sepyear = 4 if spedate >= td(1/7/2015)
replace sepyear = 5 if spedate >= td(1/7/2016)

order ppn agregroup sex admdate sepdate sepyear

ta p2y12_60 sepyear if rid == 2, col chi

forval ii = 1/5 {
use ads_revasc_spline, clear

gen sepyear = 1
replace sepyear = 2 if spedate >= td(1/7/2013)
replace sepyear = 3 if spedate >= td(1/7/2014)
replace sepyear = 4 if spedate >= td(1/7/2015)
replace sepyear = 5 if spedate >= td(1/7/2016)

*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

logistic p2y12_60 c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD i.PV
> D i.REND i.anticoag c.polypharm if rid == 2 & sepyear == `ii´, coef

use analysis_adherence_set_2, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save disp_predict_p2y12_year`ii´, replace

}

use disp_predict_p2y12_year1, clear
forval ii = 2/5 {
append using disp_predict_p2y12_year`ii´
}
gen sepyear = _n
label define sepyear 1 "2012-2013" 2 "2013-2014" 3 "2014-2015" 4 "2015-2016" 5 "2016-2017"
label values sepyear sepyear
order sepyear A ll ul
gen Astring = string(A, "%3.2f")
gen llstring = string(ll, "%3.2f")
gen ulstring = string(ul, "%3.2f")
gen P2Y12 = Astring + " " + "(" + llstring + "_" + ulstring + ")"
keep sepyear P2Y12
save table_disp_predict_p2y12_years, replace
```

9.8 Predicted PDC at means of co-variates stratified by revascularisation

The same prediction data sets created in the initial dispensing analysis, however we applied a fractional regression analysis with a logit link function in order to predict the proportion of dats covered for each drug class, using the same adjusting variables.

```
*Set up splines followed by logistic regression model construction for PCI and CABG groups

forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

fracreg logit PDC_`i´ c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD
> i.PVD i.REND  i.anticoag c.polypharm if rid ==`ii´

use analysis_adherence_set_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_pdc_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta {
append using `i´_pdc_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_1, replace

clear
foreach i in p2y12 statin hstatin  ace_arb beta {
append using `i´_pdc_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_2, replace

*Create table of both revasc strategy predictions
use pdc_predict_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using pdc_predict_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
```

```
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_pdc_predict, replace

*Create bar graph for both revasc strategy predictions
use pdc_predict_1, clear
append using pdc_predict_2
append using pdc_predict_2
rename class Class
sort Class rid
gen classrid = _n
drop if classrid == 3 | classrid ==  6 | classrid ==   9 | classrid ==   12 | classrid ==   15

twoway ///
(bar A classrid if Class == 1 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 1 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 2 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 2 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 3 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 3 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 4 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 4 & rid ==2, barw(0.8) col(red%80)) ///
(bar A classrid if Class == 5 & rid ==1, barw(0.8) col(dknavy%90)) ///
(bar A classrid if Class == 5 & rid ==2, barw(0.8) col(red%80)) ///
(rcap ll ul classrid, col(black)), graphregion(color(white)) ///
yscale(range(0,1)) ylabel(0 "0.0"  0.2 "0.2"  0.4 "0.4"  0.6 "0.6"  0.8 "0.8"  1.0 "1.0") ytitle("PD
> C at 12 months post discharge")  ///
xlabel(1.5 "P2Y12i" 4.5 "Statin" 7.5 "High intensity statin" 10.5 "ACEi/ARB" 13.5 "Beta blocker") xt
> itle("") legend(order(1 "PCI" 2 "CABG") ring(0) position(12) region(lcolor(white) color(none)))
graph save "G:\Adam\Project - revasc and outcome prediction\Results\fig_pdc", replace
```

9.9 Sensitivity analysis

There were a number of senstivity analyses performed to confirm the results generated. These included:

1. Inclusion of first MI and no prior revascularisation only

2. Inclusion of revascularisation that occurred during indexed MI admission

3. Inclusion of revascularisation within 90 days

It should be noted that the third sensitivity analysis being conducted includes events that occurred after the 60 dispensing cut off point for the first outcome, however as this is a sensitivity analysis, the methdological issue is acknowledged but not considered an error for this purpose.

9.9.1 Inclusion of first MI only

Creation

```
use ads_revasc_age10, clear
drop if rid == 0
keep if hosp_mort == 0
keep if priorMI == 0
keep if prevPCI == 0
keep if prevCABG == 0
```

Therefore, 2,097 admissions involved a prior MI within the cohort window (2012-2017), and are removed from this analysis. The same analysis for predicted dispensing and PDC is then performed.

Initial dispensing

```
gen agespline = agegroup + 2.5 if agegroup != 85
replace agespline = agegroup + 5 if agegroup == 85
save ads_revasc_spline_sa_nopriormi, replace

*Creation of prediction data set, using marginal effects at the means of independent variables for P
> CI and CABG. These are located in the appendices of the manuscript.

forval ii = 1/2 {
use ads_revasc_spline_sa_nopriormi, clear
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND  IRSD_score mean_ARIA anticoag poly
> pharm {
su(`i') if rid == `ii'
local m`i' = r(mean)
}
clear
set obs 1
gen rid = `ii'
br
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND  IRSD_score mean_ARIA anticoag poly
> pharm {
gen `i' = `m`i''
}
mkspline ARIAS= mean_ARIA, cubic knots(0 0.05 0.5 2)
mkspline ages = agespline, cubic knots(32.5 62.5 72.5 77.5)
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)
save analysis_adherence_set_sa_nopriormi_`ii', replace
}
append using analysis_adherence_set_sa_nopriormi_1
gen revasc = ""
replace revasc = "PCI" if rid == 1
replace revasc = "CABG" if rid  == 2
order revasc
sort rid
save covariate_means_sa_nopriormi, replace
use covariate_means_sa_nopriormi, clear
*Set up splines followed by logistic regression model construction for PCI and CABG groups

forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_nopriormi, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

logistic `i'_60 c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD i.PVD
> i.REND  i.anticoag c.polypharm if rid ==`ii', coef

use analysis_adherence_set_sa_nopriormi_`ii', clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i'_60_sa_nopriormi_`ii', replace

}
}
```

```
clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_nopriormi_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_sa_nopriormi_1, replace

clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_nopriormi_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_sa_nopriormi_2, replace

*Create table of both revasc strategy predictions
use disp_predict_sa_nopriormi_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using disp_predict_sa_nopriormi_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_disp_predict_sa_nopriormi, replace
```

PDC at 12 months

```
forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_nopriormi, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

fracreg logit PDC_`i´ c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD
> i.PVD i.REND  i.anticoag c.polypharm if rid ==`ii´

use analysis_adherence_set_sa_nopriormi_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_pdc_sa_nopriormi_`ii´, replace

}
}
```

```
clear
foreach i in p2y12 statin hstatin ace_arb beta {
append using `i´_pdc_sa_nopriormi_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_sa_nopriormi_1, replace

clear
foreach i in p2y12 statin hstatin  ace_arb beta {
append using `i´_pdc_sa_nopriormi_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_sa_nopriormi_2, replace

*Create table of both revasc strategy predictions
use pdc_predict_sa_nopriormi_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using pdc_predict_sa_nopriormi_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_pdc_predict_sa_nopriormi, replace
```

9.9.2 Inclusion of revascularisation that occurred during indexed MI admission

This sensitivity analysis reduced the time allowed for revascularisation to be considered part of the index MI to just the admission only. New set of covariates were created as part of this analysis and initial dispensing and PDC were predicted. **Cohort creation**

```
use ads_revasc_age10, clear
drop if rid == 0
keep if hosp_mort == 0

count if PCI < admdate
count if CABG < admdate

gen sa_pci0 = 0
replace sa_pci0 = 1 if PCI <= sepdate
ta sa_pci0 PCI_tag

gen sa_cabg0 = 0
replace sa_cabg0 = 1 if CABG <= sepdate
ta sa_cabg0 CABG_tag

ta sa_cabg0 rid
drop if rid == 2 & sa_cabg0 == 0

ta sa_pci0 rid
drop if rid == 1 & sa_pci0 == 0
```

It can be seen that when adjusting the inclusion to be restricted to only the index MI admission, there are 442 PCI and 288 CABG that are performed within 30 days of the index NSTEMI admission. These will be excluded from the analysis. **Initial dispensing**

```
gen agespline = agegroup + 2.5 if agegroup != 85
replace agespline = agegroup + 5 if agegroup == 85
save ads_revasc_spline_sa_admrevasc, replace

*Creation of prediction data set, using marginal effects at the means of independent variables for P
> CI and CABG. These are located in the appendices of the manuscript.

forval ii = 1/2 {
use ads_revasc_spline_sa_admrevasc, clear
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND IRSD_score mean_ARIA anticoag polyp
> harm {
su(`i´) if rid == `ii´
local m`i´ = r(mean)
}
clear
set obs 1
gen rid = `ii´
br
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND IRSD_score mean_ARIA anticoag polyp
> harm {
gen `i´ = `m`i´´
}
mkspline ARIAS= mean_ARIA, cubic knots(0 0.05 0.5 2)
mkspline ages = agespline, cubic knots(32.5 62.5 72.5 77.5)
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)
save analysis_adherence_set_sa_admrevasc_`ii´, replace
}
append using analysis_adherence_set_sa_admrevasc_1
gen revasc = ""
replace revasc = "PCI" if rid == 1
replace revasc = "CABG" if rid  == 2
order revasc
sort rid
save covariate_means_sa_admrevasc, replace
use covariate_means_sa_admrevasc, clear
*Set up splines followed by logistic regression model construction for PCI and CABG groups

forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_admrevasc, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

logistic `i´_60 c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD i.PVD
> i.REND i.anticoag c.polypharm if rid ==`ii´, coef

use analysis_adherence_set_sa_admrevasc_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_60_sa_admrevasc_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_admrevasc_1
}
gen diff = A - ll
```

```
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_sa_admrevasc_1, replace

clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_admrevasc_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_sa_admrevasc_2, replace

*Create table of both revasc strategy predictions
use disp_predict_sa_admrevasc_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using disp_predict_sa_admrevasc_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_disp_predict_sa_admrevasc, replace
```

PDC at 12 months

```
forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_admrevasc, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

fracreg logit PDC_`i´ c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD
> i.PVD i.REND i.anticoag c.polypharm if rid ==`ii´

use analysis_adherence_set_sa_admrevasc_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_pdc_sa_admrevasc_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta {
append using `i´_pdc_sa_admrevasc_1
}
gen diff = A - ll
```

```
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_sa_admrevasc_1, replace

clear
foreach i in p2y12 statin hstatin  ace_arb beta {
append using `i´_pdc_sa_admrevasc_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_sa_admrevasc_2, replace

*Create table of both revasc strategy predictions
use pdc_predict_sa_admrevasc_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using pdc_predict_sa_admrevasc_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_pdc_predict_sa_admrevasc, replace
```

9.9.3   Inclusion of revascularisation within 90 days

This analysis extends the window of inclusion for revascularisation to 90 days, which is reflective of the cutoffs used by the Victorian Cardiac Outcome Registry for PCI. [9]

```
use ads_revasc_age10, clear
drop if rid == 0
keep if hosp_mort == 0

count if CABG < admdate
count if PCI < admdate

gen sa_pci90 = 0
replace sa_pci90 = 1 if PCI <= sepdate + 90
ta sa_pci90 PCI_tag
replace PCI_tag = 1 if sa_pci90 == 1

gen sa_cabg90 = 0
replace sa_cabg90 = 1 if CABG <= sepdate + 90
ta sa_cabg90 CABG_tag
replace CABG_tag = 1 if sa_cabg90 == 1

ta sa_cabg90 rid
replace rid = 2 if rid == 1 & sa_cabg90 == 1 & CABG > PCI

ta sa_pci90 rid
replace rid = 1 if rid == 2 & sa_pci90 == 1 & PCI > CABG
```

It can be seen that when adjusting the inclusion to be include revascularisation up to 90 days following speration, there were 4 extra admissions classified as PCI, 149 extra admissions for CABG, and 5 admissions that were originally CABG who received subsequent PCI and therefore were classified as PCI. **Initial dispensing**

```
gen agespline = agegroup + 2.5 if agegroup != 85
replace agespline = agegroup + 5 if agegroup == 85
```

```
save ads_revasc_spline_sa_90revasc, replace
*Creation of prediction data set, using marginal effects at the means of independent variables for P
> CI and CABG. These are located in the appendices of the manuscript.
forval ii = 1/2 {
use ads_revasc_spline_sa_90revasc, clear
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND  IRSD_score mean_ARIA anticoag poly
> pharm {
su(`i´) if rid == `ii´
local m`i´ = r(mean)
}
clear
set obs 1
gen rid = `ii´
br
foreach i in agespline sex priorMI CBD HT AF DM CHF CPD PVD REND IRSD_score mean_ARIA anticoag polyp
> harm {
gen `i´ = `m`i´´
}
mkspline ARIAS= mean_ARIA, cubic knots(0 0.05 0.5 2)
mkspline ages = agespline, cubic knots(32.5 62.5 72.5 77.5)
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)
save analysis_adherence_set_sa_90revasc_`ii´, replace
}
append using analysis_adherence_set_sa_90revasc_1
gen revasc = ""
replace revasc = "PCI" if rid == 1
replace revasc = "CABG" if rid  == 2
order revasc
sort rid
save covariate_means_sa_90revasc, replace
use covariate_means_sa_90revasc, clear
*Set up splines followed by logistic regression model construction for PCI and CABG groups
forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_90revasc, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

logistic `i´_60 c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD i.PVD
> i.REND i.anticoag c.polypharm if rid ==`ii´, coef

use analysis_adherence_set_sa_90revasc_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_60_sa_90revasc_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_90revasc_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
```

```stata
label values class class
order class A ll ul diff rid
save disp_predict_sa_90revasc_1, replace

clear
foreach i in p2y12 statin hstatin ace_arb beta  {
append using `i´_60_sa_90revasc_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save disp_predict_sa_90revasc_2, replace

*Create table of both revasc strategy predictions
use disp_predict_sa_90revasc_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using disp_predict_sa_90revasc_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_disp_predict_sa_90revasc, replace
```

PDC at 12 months

```stata
forval ii = 1/2 {
foreach i in p2y12 statin ace_arb beta hstatin {
use ads_revasc_spline_sa_90revasc, clear
*create splines for remoteness
mkspline ARIAS = mean_ARIA, cubic knots(0 0.05 0.5 2)
*create splines for ages
mkspline ages = agegroup, cubic knots(32.5 62.5 72.5 77.5)
drop agespline
*create splines for IRSD
mkspline IRSDspline = IRSD_score, cubic knots(780 964 1004 1051)

fracreg logit PDC_`i´ c.ARIAS* c.ages* sex c.IRSDspline* i.priorMI i.CBD i.HT i.AF i.DM i.CHF i.CPD
> i.PVD i.REND i.anticoag c.polypharm if rid ==`ii´

use analysis_adherence_set_sa_90revasc_`ii´, clear
*create prediction variable from model, then transform into probabability
predict A
predict xb, xb
*create SE to build confidence intervals
predict B, stdp
gen ll = xb - (invnormal(0.975)*B)
gen ul = xb + (invnormal(0.975)*B)
replace ll = invlogit(ll)
replace ul = invlogit(ul)
keep A ul ll rid

save `i´_pdc_sa_90revasc_`ii´, replace

}
}
clear
foreach i in p2y12 statin hstatin ace_arb beta {
append using `i´_pdc_sa_90revasc_1
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
```

```
label values class class
order class A ll ul diff rid
save pdc_predict_sa_90revasc_1, replace

clear
foreach i in p2y12 statin hstatin  ace_arb beta {
append using `i´_pdc_sa_90revasc_2
}
gen diff = A - ll
gen class = _n
label define class 1 "P2Y12 inhibitor" 2 "Statin" 4 "ACE inhibitor or ARB" 5 "Beta blocker" 3 "High
> intensity statin"
label values class class
order class A ll ul diff rid
save pdc_predict_sa_90revasc_2, replace

*Create table of both revasc strategy predictions
use pdc_predict_sa_90revasc_1, clear
rename (A ll ul rid) (A1 ll1 ul1 rid1)
merge 1:1 class using pdc_predict_sa_90revasc_2
rename (A ll ul rid class) (A2 ll2 ul2 rid2 Class)
gen A2sum = string(A2, "%03.2f")
gen ul2sum = string(ul2, "%03.2f")
gen ll2sum = string(ll2, "%03.2f")
gen A1sum = string(A1, "%03.2f")
gen ul1sum = string(ul1, "%03.2f")
gen ll1sum = string(ll1, "%03.2f")
gen PCI = A1sum + " " + "(" + ll1sum + "-" + ul1sum + ")"
gen CABG = A2sum + " " + "(" + ll2sum + "-" + ul2sum + ")"
keep Class PCI CABG
save table_pdc_predict_sa_90revasc, replace
```
Combine all sensitivity analysis with original analysis

```
br
use table_disp_predict, clear
use table_disp_predict_sa_90revasc, clear
use table_disp_predict_sa_admrevasc, clear
use table_disp_predict_sa_nopriormi, clear

foreach i in disp pdc {
foreach ii in PCI CABG {
use table_`i´_predict, clear
keep Class `ii´
rename `ii´ Primary
merge 1:1 Class using table_`i´_predict_sa_nopriormi
keep Class Primary `ii´
rename `ii´ Sensitivity_1
merge 1:1 Class using table_`i´_predict_sa_admrevasc
keep Class Primary Sensitivity_1 `ii´
rename `ii´ Sensitivity_2
merge 1:1 Class using table_`i´_predict_sa_90revasc
keep Class Primary Sensitivity_1 Sensitivity_2 `ii´
rename `ii´ Sensitivity_3
save table_`i´_sens_`ii´_all, replace
}
}

}
```

References

[1] Jennifer S. Lawton, Jacqueline E. Tamis-Holland, Sripal Bangalore, Eric R. Bates, Theresa M. Beckie, James M. Bischoff, John A. Bittl, Mauricio G. Cohen, J. Michael DiMaio, Creighton W. Don, Stephen E. Fremes, Mario F. Gaudino, Zachary D. Goldberger, Michael C. Grant, Jang B. Jaswal, Paul A. Kurlansky, Roxana Mehran, Thomas S. Metkus, Lorraine C. Nnacheta, Sunil V. Rao, Frank W. Sellke, Garima Sharma, Celina M. Yong, and Brittany A. Zwischenberger. 2021 acc/aha/scai guideline for coronary artery revascularization: A report of the american college of cardiology/american heart association joint committee on clinical practice guidelines. *Journal of the American College of Cardiology*, 79:e21–e129, 1 2022.

[2] J P Collet, H Thiele, E Barbato, O Barthelemy, J Bauersachs, D L Bhatt, P Dendale, M Dorobantu, T Edvardsen, T Folliguet, C P Gale, M Gilard, A Jobs, P Juni, E Lambrinou, B S Lewis, J Mehilli, E Meliga, B Merkely, C Mueller, M Roffi, F H Rutten, D Sibbing, and G C M Siontis. 2020 esc guidelines for the management of acute coronary syndromes in patients presenting without persistent st-segment elevation. *Eur Heart J*, 2020. Collet, Jean-Philippe Thiele, Holger Barbato, Emanuele Barthelemy, Olivier Bauersachs, Johann Bhatt, Deepak L Dendale, Paul Dorobantu, Maria Edvardsen, Thor Folliguet, Thierry Gale, Chris P Gilard, Martine Jobs, Alexander Juni, Peter Lambrinou, Ekaterini Lewis, Basil S Mehilli, Julinda Meliga, Emanuele Merkely, Bela Mueller, Christian Roffi, Marco Rutten, Frans H Sibbing, Dirk Siontis, George C M eng England Eur Heart J. 2020 Aug 29. pii: 5898842. doi: 10.1093/eurheartj/ehaa575.

[3] Lachlan L. Dalli, Monique F. Kilkenny, Isabelle Arnet, Frank M. Sanfilippo, Doyle M. Cummings, Moira K. Kapral, Joosup Kim, Jan Cameron, Kevin Y. Yap, Melanie Greenland, and Dominique A. Cadilhac. Towards better reporting of the proportion of days covered method in cardiovascular medication adherence: A scoping review and new tool ten-spiders, 10 2022.

[4] L Mellish, E A Karanges, M J Litchfield, A L Schaffer, B Blanch, B J Daniels, A Segrave, and S A Pearson. The australian pharmaceutical benefits scheme data collection: a practical guide for researchers. *BMC Res Notes*, 8:634, 2015. Mellish, Leigh Karanges, Emily A Litchfield, Melisa J Schaffer, Andrea L Blanch, Bianca Daniels, Benjamin J Segrave, Alicia Pearson, Sallie-Anne eng Research Support, Non-U.S. Gov't England BMC Res Notes. 2015 Nov 2;8:634. doi: 10.1186/s13104-015-1616-8.

[5] Nashwa Masnoon, Sepehr Shakib, Lisa Kalisch-Ellett, and Gillian E. Caughey. What is polypharmacy? a systematic review of definitions, 10 2017.

[6] Yasuaki Takeji, Hiroki Shiomi, Takeshi Morimoto, Ko Yamamoto, Yukiko Matsumura-Nakano, Kazuya Nagao, Ryoji Taniguchi, Kyohei Yamaji, Tomohisa Tada, Eri Toda Kato, Yusuke Yoshikawa, Yuki Obayashi, Satoru Suwa, Moriaki Inoko, Natsuhiko Ehara, Toshihiro Tamura, Tomoya Onodera, Hiroki Watanabe, Mamoru Toyofuku, Kenji Nakatsuma, Hiroki Sakamoto, Kenji Ando, Yutaka Furukawa, Yukihito Sato, Yoshihisa Nakagawa, Kazushige Kadota, and Takeshi Kimura. Differences in mortality and causes of death between stemi and nstemi in the early and late phases after acute myocardial infarction. *PLoS ONE*, 16, 11 2021.

[7] Hude Quan, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L Duncan Saunders, Cynthia A Beck, Thomas E Feasby, and William A

Ghali. Coding algorithms for defining comorbidities in icd-9-cm and icd-10 administrative data, 2005.

[8] Australian Institute of Health and Welfare. Rural and remote health, 2022.

[9] J Lefkovitz, A Brennan, D Dinh, H Carruthers, J Doyle, M Lucas, D Stubb, and C Reid. The victorian cardiac outcomes registry annual report 2022, 8 2022.