/\*\*\*\*\*\*\*\*\*\*\*\* FFR \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

ANALYTIC PLAN

Cohort denominator

All people who had a coronary angiography 2012-2013. Coronary angiography will be defined as any of the following HCPCs codes reported in 20% Medicare Part B (Carrier) files:

Coronary angiography (excluding evaluation of bypass grafts): (HCPCS) 93454, 93456, 93458, 93460

Include PCI: (HCPCS) 92920, 92921, 92924, 92925, 92928, 92929, 92933, 92934, 92943, 92944, 92973, 92974, 92980, 92981, 92982, 92984, 92995, 92996

Only index procedures will be kept. Duplicates within 2012-2013 time span will be dropped from analysis.

All benes within denominator will have had continuous FFS eligibility 3months prior to, including, and 1 month after index coronary angiography.

Procedure and Diagnosis Exclusions

Exclusions include patients who had AMI, CABG, valve procedures and valve disorders.

ICD-9 Diagnosis and Procedure Codes will be pulled from the linedgns variable of the Part B files as well as from DGNS\_CD1 through DGNS\_CD25 of the MedPAR files. Patients for whom any of the following codes have been reported 365 days prior to the index coronary angiography date will be dropped.

Definition: 0 ≤ (angio\_dt – proc/dx\_dt) ≤ 365

AMI: (ICD-9 DX) 41001, 41011, 41021, 41031, 41041, 41051, 41061, 41071, 41081, 41091,

41000, 41010, 41020, 41030, 41040, 41050, 41060, 41070, 41080, 41090

Valve Procedures: (ICD 9 SG) 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511, 3512, 3513, 3514, 3520, 3521, 3522, 3523, 3524, 3525, 3526, 3527, 3528, 3596, 3597, 3599

Valve disorders: (ICD-9 DX) (single) 4240, 4241, 4242, 4243, 42490, 42491, 42499 (combined mitral/aortic) 3960, 3961, 3962, 3963, 3969 (tricuspid valve and pulmonary) 3970, 3971 (rheumatic aortic valve) 3950, 3951, 3952, 3959 (presence of valve prosthesis) 99602, 99671

HCPCS procedure codes will be pulled from Part B files only.

CABG: (HCPCS) 33510, 33511, 33512, 33513, 33514, 33516, [33517, 33518, 33519, 33521, 33522, 33523] 33533, 33534, 33535, 33536, 35600, [33572, 4110F]

Additionally, CCW files will be used to identify AMIs occurring in the year prior to the index coronary angiography.

ER Exclusions

Exclusions include patients admitted from an emergency department or transferred from an outpatient emergency department.

ER exclusions will be pulled from the inpatient MedPAR files as well as the Outpatient files.

Outpatient files: Revenue Center Codes 0450-0459 (Emergency room), 0981 (Professional fees-Emergency room).

Inpatient MedPAR: Emergency Room Charge Amount > $0

Inpatient Transfers (MedPAR files): Destination Code “dstntncd” = 02, 05, 09

To identify ER patients in the outpatient data who may have been admitted to the hospital, outpatient ER visits will be linked by date (inpatient admission within one day of outpatient rev. date) to Medpar files with transfer destination code “dstntncd” = 02, 05, 09

02 = Discharged/transferred to other short term

general hospital for inpatient care.

05 = Discharged/transferred to another type

of institution for inpatient care (including

distinct parts). NOTE: Effective 1/2005,

psychiatric hospital or psychiatric distinct

part unit of a hospital will no longer be

identified by this code. New code is '65'.

09 = Admitted as an inpatient to this

hospital (effective 3/1/91). In situa-

tions where a patient is admitted before

midnight of the third day following the

day of an outpatient service, the out-

patient services are considered inpatient.

Patients within the MedPAR file with ER\_AMT >0$, or patients identified as having been transferred from an outpatient ER will be dropped from analysis.

Definition: -1 day ≤ (corangio\_dt – ER\_dt) ≤ 3

Analytic Stratifications

Stratify final population by Intervention v. No Intervention. Intervention is defined as PCI or CABG occurring 30 days following index coronary angiography.

Definition: 0 days ≤ (intervention dt – corangio dt) ≤ 30 days.

PCI: (HCPCS) 92920, 92921, 92924, 92925, 92928, 92929, 92933, 92934, 92943, 92944, 92973, 92974, 92980, 92981, 92982, 92984, 92995, 92996.

CABG: (HCPCS) 33510, 33511, 33512, 33513, 33514, 33516, [33517, 33518, 33519, 33521, 33522, 33523] 33533, 33534, 33535, 33536, 35600, [33572, 4110F]

Stratify both Intervention and No-Intervention populations by whether or not they underwent FRR at the time of the index coronary angiography.

Definition 3 days ≤ (FFR dt – corangio dt) ≤ 3 days

FFR: (HCPCS) 93571, 93572

Further stratify the stratified populations by whether or not they underwent stress testing 90 days prior to index coronary angiography.

Definition: 0 days ≤ (corangio dt – stress dt) ≤ 90 days

Stress testing HCPS codes will be pulled from Part B files, ICD 9 procedure codes will be pulled from Part B file and MedPAR files.

Stress Testing: (HCPCS) 93350, 93351, [78466, 78468, 78469, 78472, 78473, 78481, 78494, 78496, 78460, 78461, 78464, 78465, 78478, 78480, 78483,] 78451, 78452, [78453, 78454, 78491, 78492,] 93015, 93016, 93017, 93018, [J2785, J0150, J0151, J0152, J0153, J1245, J1250, J0395, G8965;]

(ICD 9 SG) 89.41, 89.42, 89.43, 89.44

\*/

/\*

Summary of Diagnosis and Procedure Codes Used

Coronary angiography (excluding evaluation of bypass grafts): 93454, 93456, 93458, 93460.

AMI: 410xx

Coronary artery atherosclerosis: 41400, 41401, 41402, 41403, 41404, 41405, 41406, 41407.

\*PCI: 92920, 92921, 92924, 92925, 92928, 92929, 92933, 92934, 92943, 92944, 92973, 92974, 92980, 92981, 92982, 92984, 92995, 92996.

CABG: 00566\*\*, 00567\*\*, 33510, 33511, 33512, 33513, 33514, 33516, [33517, 33518, 33519, 33521, 33522, 33523] 33533, 33534, 33535, 33536, 35600, [33572, 4110F]

Valve Procedures: 3500, 3501, 3502, 3503, 3504, 3505, 3506, 3507, 3508, 3509, 3510, 3511, 3512, 3513, 3514, 3520, 3521, 3522, 3523, 3524, 3525, 3526, 3527, 3528, 3596, 3597, 3599

Valve disorders: (single) 4240, 4241, 4242, 4243, 42490, 42491, 42499 (combined mitral/aortic) 3960, 3961, 3962, 3963, 3969 (tricuspid valve and pulmonary) 3970, 3971 (rheumatic aortic valve) 3950, 3951, 3952, 3959 (presence of valve prosthesis) 99602, 99671

FFR: 93571, 93572.

Stress Testing: 93350, 93351, [78466, 78468,78469, 78472, 78473, 78481, 78494, 78496, 78460, 78461, 78464, 78465, 78478, 78480, 78483,] 78451, 78452, [78453, 78454, 78491, 78492,] 93015, 93016, 93017, 93018, [J2785, J0150, J0151, J0152, J0153, J1245, J1250, J0395, G8965;] 89.41, 89.42, 89.43, 89.44

\*Removed: 92927, 92928 (presence of bypass grafts)

\*\* anesthesia related to CABG

/\*\*\*COHORT CONTRUCTION STEPS\*\*\*/

**1.** Create Procedure Cohort

a. Pull relevant procedure codes (defined above) from Medicare Pt B Carrier files

b. Create variable names for procedures

c. Remove duplicate procedures - only keep index procedure, final file named corangio&yr.3

**2.** Construct FFS cohort steps

a. Read in denominator files **2011**-**2014**

b. Only OASI

c. Only Elderly (>**65** years old)

d. Only FFS eligible in given month

e. DROP EXCLUSIONS - Create new descriptive variables

f. Convert character zip to numeric

g. Create FFS dataset for all years combined containing only necessary variables for determining continuous eligibility later

h. Save combined-year FFS dataset as ffs.ffs08thru14

i. Save to FFS folder as ffs.ffs(yr)

**3.** Merge Coronary Angio and FFS cohorts

a. keep only FFS eligible coronary angiographies - merge by month and then stack, named final files eligangio(yr)

**4.** Merge in CCW with eligangio dataset and exclude benes who had CCW AMI in the year prior to index coronary angiography

a. Read in CCW files and bene xwalk for **2011** to **2013**

b. Note - Bene\_id\_14049 variable links stein **20**% Medicare **data**, bene\_id\_21074 variable links CCW **data**

c. Merge eligangio(yr) with benexwalk by bene\_id\_14049

d. Separate by year, merge eligangio(yr) with cc(yr) by bene\_id\_21074, name cceligangio(yr)

e. Define chronic conditions on whether claims were met (AMI = **1**,**3**)

f. drop AMIs from cohort which met claims conditions

g. stack cceligangio(yr)s

h. save final stacked file named eligangio to temp folder

**5.** Keep only benes with continuous ffs eligibility **3** months prior to and **1** month after index coronary angio

a. Merge eligangio dataset with combined year FFS dataset, drop if bene is not FFS eligible **3** months prior to and **1** month

after index coronary angio or if they died within **100** days after coronary angio

b. keep only index procedures for all years

c. Save final file nodupeligangio in temp folder

**6.** Read in and create datasets for Medpar exclusion procedures (Same as in part B but from Medpar file)

a. read in medpar files, rename dgnscd vars so all years match

b. Output separate datasets for AMI, valve procedures, and stress tests found in any of the **25** diagnosis codes

c. Assign variable names to output codes, keep only merging variables and newly assigned diagnosis variable names

d. Merge ami, valve, stress datasets back to original medpar dataset for each year to add in new indicator diagnosis variables,

final files named mp(yr), saved to temp folder as mpdx(yr)

**7.** ER Exclusions cohort contruction - patients admitted or transferred from an ED

a. read in outpatient files

b. identify ED outpatient visits

c. identify inpatient transfers from medpar file

d. identify inpatient transfers from outpatient ED

e. merge OP ED transfers into latest full medpar file

f. create dataset containing only merging variables and new variables for inpatient ER or outpatient ER transfers\*/

g. keep only ER exclusions and stack separate year datasets into one to save to iac folder\*/

h. remove duplicate observations with the same admission date break stacked dataset back up by year of admission

**8.** Assemble all Part B, Medpar, ER exclusions, outcomes (revasc and FFR), and covariates (prior stress) to be merged into eligible bene cohort, drop exclusions

a. Make separate datasets for each Downstream/Concurrent/Past Procedures from coronary angio cohort (from Part B file) - PCI, CABG, FFR, ami, stress, and valve procedures

b. Make separate datasets for each Downstream/Concurrent/Past procedure/diagnosis from medpar exclusion cohort - ami, stress, valve

c. Combine like indicators from Medpar and Part B - stress, ami, valve, remove duplicates

d. output dataset of ER Exclusions from latest eligible angio cohort (nodupeligangio) if ER visit happened **3** days before or **1** day after angio

e. merge identified ER exclusions into original eligible cohort and drop ER exclusions

f. Repeat steps d. and e. for AMI **1** year prior, CABG **1** year prior, and valve procedures **1** year prior

g. Create dataset of variable for revasc if bene had PCI or CABG **30** days after angio

h. output dataset of downstream revasc from latest eligible angio cohort (novalve4) if revasc happened within **30** days after angio

i. merge identified revasc into last eligible cohort

j. Repeat steps g. through i. for creating a variable for FFR - if FFR occurred **3** days before or after angio, and for stress **90** days prior to angio

k. save in final folder as finalcohort

**9.** Create variable for geographic regions, tweak covariates, perform analysis

a. define regions

b. Combine stress or ffr into one variable

c. Exploratory Descriptive Analysis

d. Newest final model with sex ref = male instrad and agegrp = **65**-**75**, **75**-**85**, >**85** instead

e. Final Descriptive Analysis

f. Exploratory Analytical Models

g. Final Analytical Model

\*/

/\*Assign libraries\*/

libname iac 'W:\Jesse\IAC';

libname stdall 'W:\Medicare\_Claims';

libname vingt 'O:\Medicare\_20%\_Stein\Denominator';

libname vingtMP 'o:\Medicare\_20%\_Stein\MedPAR';

libname vingtOP 'O:\Medicare\_20%\_Stein\Outpatient';

libname vingtPB 'O:\Medicare\_20%\_Stein\PartB';

libname ffr 'W:\Jesse\FFR';

libname holl 'O:\Hollenbeck\_20%\2013\_Original';

libname kaytemp 'W:\Jesse\FFR\Joseph and Bradley March 17\TEMP DATASETS';

libname kayfinal 'W:\Jesse\FFR\Joseph and Bradley March 17\FINAL DATASETS';

libname venk 'O:\Murthy\_CardiacTesting';

libname ffs 'W:\Jesse\FFS\FFS COHORT';

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. Create Procedure Cohort\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*a. Pull relevant procedure codes (defined above) from Medicare Pt B Carrier files

Data Structure of PtB files: Data is line level, such that an individual HCPC is represented

in each line. Benes may have multiple claims and claims may have multiple lines. Not all lines are unique.

Same day duplicate procedures will be dropped later.

Part B is used to pull CPT (labeled as HCPCs) codes for relevant procedures. While most relevant diagnoses will be

pulled later from the Medpar files, they were also pulled from the linedgns variable of the Part B files to

help capture additional diagnoses of AMI, stress tests, or valve disorders. This would have only contributed

marginal new observations to the cohort as these diagnoses would be more thoroughly documented in Medpar files.

Having constructed the cohort, additional exploratory analysis was performed by pulling ICD9 procedure codes

for stress and angio. It was determined to not be worth adding in the ICD-9 codes for these as they contributed a

<1% difference to the final cohort after merging in the part B procedures and performing exclusions. The majority

of procedures will be captured more thoroughly in the Part B files and didn't merit pulling additional ICD9 codes

from Medpar\*/

/\*2011\*/

**data** ptb11lnits ;

set vingtPB.ptb11lnits (keep= bene\_id line\_num clm\_id provzip9 prvstate LINE\_ICD\_DGNS\_CD HCPCS\_CD EXPNSDT1 EXPNSDT2 PRF\_NPI PRV\_TYPE

rename = (LINE\_ICD\_DGNS\_CD = LINEDGNS EXPNSDT1=sexpndt1 EXPNSDT2=sexpndt2)

where = (hcpcs\_cd in (

/\*cor. Angio\*/ '93454', '93456', '93458', '93460',

/\*PCI\*/ '92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996',

/\*FFR CFR \*/ '93571', '93572',

/\*cabg\*/ '33510', '33511', '33512', '33513',

'33514', '33516', '33517', '33518', '33519', '33521', '33522', '33523', '33533',

'33534', '33535', '33536', '33572', '35600', '4110F',

/\*stress\*/ '93350', '93351', '78466', '78468','78469', '78472', '78473',

'78481', '78494', '78496', '78460', '78461', '78464','78465', '78478', '78480', '78483',

'78451', '78452', '78453', '78454', '78491', '78492', '93015', '93016', '93017', '93018',

'J2785', 'J0150', 'J0151', 'J0152', 'J0153', 'J1245', 'J1250','J0395', 'G8965'

)

/\*ICD 9 DX and SG \*/

or linedgns in (

/\*AMI\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090',

/\*stress dx\*/ '8941', '8942', '8943', '8944',

/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671'

)

));

**run**;

/\*2012\*/

**data** ptb12lnits ;

set vingtPB.ptb12lnits (keep= bene\_id line\_num clm\_id LINE\_ICD\_DGNS\_CD PRVDR\_ZIP PRVDR\_STATE\_CD HCPCS\_CD LINE\_1ST\_EXPNS\_DT LINE\_LAST\_EXPNS\_DT PRF\_PHYSN\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD

rename = (PRVDR\_ZIP = provzip9 PRVDR\_STATE\_CD = PRVSTATE LINE\_ICD\_DGNS\_CD = LINEDGNS LINE\_1ST\_EXPNS\_DT = sexpndt1 LINE\_LAST\_EXPNS\_DT = sexpndt2 PRF\_PHYSN\_NPI=PRF\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD=PRV\_TYPE)

where = (hcpcs\_cd in (

/\*cor. Angio\*/ '93454', '93456', '93458', '93460',

/\*PCI\*/ '92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996',

/\*FFR CFR \*/ '93571', '93572',

/\*cabg\*/ '33510', '33511', '33512', '33513',

'33514', '33516', '33517', '33518', '33519', '33521', '33522', '33523', '33533',

'33534', '33535', '33536', '33572', '35600', '4110F',

/\*stress\*/ '93350', '93351', '78466', '78468','78469', '78472', '78473',

'78481', '78494', '78496', '78460', '78461', '78464','78465', '78478', '78480', '78483',

'78451', '78452', '78453', '78454', '78491', '78492', '93015', '93016', '93017', '93018',

'J2785', 'J0150', 'J0151', 'J0152', 'J0153', 'J1245', 'J1250','J0395', 'G8965'

)

/\*ICD 9 DX and SG \*/

or linedgns in (

/\*AMI\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090',

/\*stress dx\*/ '8941', '8942', '8943', '8944',

/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671'

)

));

**run**;

/\*2013\*/

**data** ptb13lnits ;

set vingtPB.ptb13lnits (keep= bene\_id line\_num clm\_id LINE\_ICD\_DGNS\_CD PRVDR\_ZIP PRVDR\_STATE\_CD HCPCS\_CD LINE\_1ST\_EXPNS\_DT LINE\_LAST\_EXPNS\_DT PRF\_PHYSN\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD

rename = (PRVDR\_ZIP = provzip9 PRVDR\_STATE\_CD = PRVSTATE LINE\_ICD\_DGNS\_CD = LINEDGNS LINE\_1ST\_EXPNS\_DT = sexpndt1 LINE\_LAST\_EXPNS\_DT = sexpndt2 PRF\_PHYSN\_NPI=PRF\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD=PRV\_TYPE)

where = (hcpcs\_cd in (

/\*cor. Angio\*/ '93454', '93456', '93458', '93460',

/\*PCI\*/ '92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996',

/\*FFR CFR \*/ '93571', '93572',

/\*cabg\*/ '33510', '33511', '33512', '33513',

'33514', '33516', '33517', '33518', '33519', '33521', '33522', '33523', '33533',

'33534', '33535', '33536', '33572', '35600', '4110F',

/\*stress\*/ '93350', '93351', '78466', '78468','78469', '78472', '78473',

'78481', '78494', '78496', '78460', '78461', '78464','78465', '78478', '78480', '78483',

'78451', '78452', '78453', '78454', '78491', '78492', '93015', '93016', '93017', '93018',

'J2785', 'J0150', 'J0151', 'J0152', 'J0153', 'J1245', 'J1250','J0395', 'G8965'

)

/\*ICD 9 DX and SG \*/

or linedgns in (

/\*AMI\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090',

/\*stress dx\*/ '8941', '8942', '8943', '8944',

/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671'

)

));

**run**;

/\*2014\*/

**data** ptb14lnits ;

set vingtPB.ptb14lnits (keep= bene\_id line\_num clm\_id LINE\_ICD\_DGNS\_CD PRVDR\_ZIP PRVDR\_STATE\_CD HCPCS\_CD LINE\_1ST\_EXPNS\_DT LINE\_LAST\_EXPNS\_DT PRF\_PHYSN\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD

rename = (PRVDR\_ZIP = provzip9 PRVDR\_STATE\_CD = PRVSTATE LINE\_ICD\_DGNS\_CD = LINEDGNS LINE\_1ST\_EXPNS\_DT = sexpndt1 LINE\_LAST\_EXPNS\_DT = sexpndt2 PRF\_PHYSN\_NPI=PRF\_NPI CARR\_LINE\_PRVDR\_TYPE\_CD=PRV\_TYPE)

where = (hcpcs\_cd in (

/\*cor. Angio\*/ '93454', '93456', '93458', '93460',

/\*PCI\*/ '92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996',

/\*FFR CFR \*/ '93571', '93572',

/\*cabg\*/ '33510', '33511', '33512', '33513',

'33514', '33516', '33517', '33518', '33519', '33521', '33522', '33523', '33533',

'33534', '33535', '33536', '33572', '35600', '4110F',

/\*stress\*/ '93350', '93351', '78466', '78468','78469', '78472', '78473',

'78481', '78494', '78496', '78460', '78461', '78464','78465', '78478', '78480', '78483',

'78451', '78452', '78453', '78454', '78491', '78492', '93015', '93016', '93017', '93018',

'J2785', 'J0150', 'J0151', 'J0152', 'J0153', 'J1245', 'J1250','J0395', 'G8965'

)

/\*ICD 9 DX and SG \*/

or linedgns in (

/\*AMI\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090',

/\*stress dx\*/ '8941', '8942', '8943', '8944',

/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671'

)

));

**run**;

/\*Save Pt B procedures as files in Temporary folder\*/

**data** kaytemp.ptb11lnits;

set ptb11lnits;

**run**;

**data** kaytemp.ptb12lnits;

set ptb12lnits;

**run**;

**data** kaytemp.ptb13lnits;

set ptb13lnits;

**run**;

**data** kaytemp.ptb14lnits;

set ptb14lnits;

**run**;

/\*Read in saved Part B codes from Temporary folder\*/

**data** ptb11lnits;

set kaytemp.ptb11lnits;

**run**;

**data** ptb12lnits;

set kaytemp.ptb12lnits;

**run**;

**data** ptb13lnits;

set kaytemp.ptb13lnits;

**run**;

**data** ptb14lnits;

set kaytemp.ptb14lnits;

**run**;

/\*b. Create variables for procedures \*/

**%macro** angio(yr);

data angio&yr.;

set ptb&yr.lnits;

if hcpcs\_cd in (/\*cor. Angio + PCI\*/ '93454', '93456', '93458', '93460',/\* + \*/

'92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996') then corangio=**1**; else corangio=**0**;

if hcpcs\_cd in (/\*PCI\*/ '92920', '92921', '92924', '92925', '92928', '92929', '92933', '92934',

'92943', '92944', '92973', '92974', '92980', '92981',

'92982', '92984', '92995', '92996') then pci=**1**; else pci=**0**;

if hcpcs\_cd in (/\*cabg\*/ '33510', '33511', '33512', '33513',

'33514', '33516', '33517', '33518', '33519', '33521', '33522', '33523', '33533',

'33534', '33535', '33536', '33572', '35600', '4110F' ) then cabg=**1**; else cabg=**0**;

if hcpcs\_cd in (/\*FFR CFR \*/ '93571', '93572') then ffr=**1**; else ffr=**0**;

if hcpcs\_cd in (/\*stress\*/ '93350', '93351', '78466', '78468','78469', '78472', '78473',

'78481', '78494', '78496', '78460', '78461', '78464','78465', '78478', '78480', '78483',

'78451', '78452', '78453', '78454', '78491', '78492', '93015', '93016', '93017', '93018',

'J2785', 'J0150', 'J0151', 'J0152', 'J0153', 'J1245', 'J1250','J0395', 'G8965') then stress=**1**; else stress=**0**;

if linedgns in (/\*AMI\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090') then pbami=**1**; else pbami=**0**;

if linedgns in (/\*stress dx\*/ '8941', '8942', '8943', '8944') then pbstress=**1**; else pbstress=**0**;

if linedgns in (/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671') then pbvalve=**1**; else pbvalve=**0**;

sexpmonth = month(sexpndt1);

refyr= year(sexpndt1);

run;

/\*c. remove non-unique duplicate procedures - keep unique duplicates for now

As of now unit of analysis is still at the line level\*/

/\*\*corangio\*\*\*/

data corangio&yr.;

set angio&yr.;

sexpmonth = month(sexpndt1);

run;

proc sort data = corangio&yr. ;

by bene\_id corangio sexpndt1 ;

run;

data corangio&yr.1;

set corangio&yr.;

count+**1**;

by bene\_id corangio;

if first.corangio then count=**1**;

if corangio^=**0**;

run;

DATA corangio&yr.2;

SET corangio&yr.1;

BY bene\_id count;

LAG\_date = LAG(sexpndt1);

DIF\_date = DIF(sexpndt1);

IF FIRST.bene\_id THEN DO;

LAG\_date = **.**;

DIF\_date = **.**;

END;

RUN;

DATA corangio&yr.3;

SET corangio&yr.2;

if dif\_date=**0** then delete;

run;

**%mend**;

%***angio***(**11**);

%***angio***(**12**);

%***angio***(**13**);

%***angio***(**14**);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* 2. Construct FFS cohort steps \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*a. Read in denominator files 2011-2014 \*/

**data** denom11;

set vingt.den11p20 (rename = (bene\_death\_dt=sdod));

**run**;

/\*2012 had different variable names from 2011 and 2013-2014 - make match\*/

**data** denom12;

set vingt.den12p20 (rename = (ENTLMT\_RSN\_CURR = BENE\_ENTLMT\_RSN\_CURR AGE\_AT\_END\_REF\_YR = BENE\_AGE\_AT\_END\_REF\_YR RACE\_CD=bene\_race\_cd SEX\_IDENT\_CD=BENE\_SEX\_IDENT\_CD zip\_cd=bene\_zip\_cd death\_dt=sdod

BUYIN\_IND\_01 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_01

BUYIN\_IND\_02 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_02

BUYIN\_IND\_03 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_03

BUYIN\_IND\_04 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_04

BUYIN\_IND\_05 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_05

BUYIN\_IND\_06 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_06

BUYIN\_IND\_07 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_07

BUYIN\_IND\_08 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_08

BUYIN\_IND\_09 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_09

BUYIN\_IND\_10 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_10

BUYIN\_IND\_11 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_11

BUYIN\_IND\_12 = BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_12

HMO\_IND\_01 = BENE\_HMO\_IND\_01

HMO\_IND\_02 = BENE\_HMO\_IND\_02

HMO\_IND\_03 = BENE\_HMO\_IND\_03

HMO\_IND\_04 = BENE\_HMO\_IND\_04

HMO\_IND\_05 = BENE\_HMO\_IND\_05

HMO\_IND\_06 = BENE\_HMO\_IND\_06

HMO\_IND\_07 = BENE\_HMO\_IND\_07

HMO\_IND\_08 = BENE\_HMO\_IND\_08

HMO\_IND\_09 = BENE\_HMO\_IND\_09

HMO\_IND\_10 = BENE\_HMO\_IND\_10

HMO\_IND\_11 = BENE\_HMO\_IND\_11

HMO\_IND\_12 = BENE\_HMO\_IND\_12

));

**run**;

**data** denom13;

set vingt.den13p20 (rename = (bene\_death\_dt=sdod));

**run**;

**data** denom14;

set vingt.den14p20 (rename = (bene\_death\_dt=sdod));

**run**;

/\* b. ONLY OASI\*/

**%macro** ffs(yr);

data oasi&yr.;

set denom&yr. ;

if BENE\_ENTLMT\_RSN\_CURR^ne **0**;

run;

/\*c. ONLY ELDERLY\*/

data oasi65&yr.;

set oasi&yr.;

if BENE\_AGE\_AT\_END\_REF\_YR^< **65**;

run;

/\*d. ONLY FFS IN GIVEN MONTH -

ENTLMT\_BUYIN

3 - Part A and Part B

C - Part A and Part B state buy-in

HMO\_IND

0 - Not a member of HMO

4 - Fee-for-service participant in case or disease management demonstration project

\*/

data ffs&yr.;

set oasi65&yr. ;

array MemberMos\_AB (**12**) BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_01 - BENE\_MDCR\_ENTLMT\_BUYIN\_IND\_12;

array MemberMos\_noHMO (**12**) BENE\_HMO\_IND\_01 - BENE\_HMO\_IND\_12;

array Member\_FFSMos (**12**) Member\_FFSMos&yr.01 - Member\_FFSMos&yr.12;

if BENE\_ENROLLMT\_REF\_YR=**20**&yr.;

do i= **1** to **12**;

if MemberMos\_AB(i) in ('3','C') & MemberMos\_noHMO(i) in ('0','4') then Member\_FFSMos(i)=**1**;

else Member\_FFSMos(i)=**0**;

end;

run;

/\* e. DROP EXCLUSIONS - Create new descriptive variables\*/

data ffs&yr.;

set ffs&yr.;

if bene\_race\_cd=**2** then black=**1**; else black=**0**;

if bene\_race\_cd=**0** then black=**.**;

if BENE\_AGE\_AT\_END\_REF\_YR^<**65**;

if BENE\_AGE\_AT\_END\_REF\_YR^>**110**;

if BENE\_SEX\_IDENT\_CD^=**0**;

if **65**=<BENE\_AGE\_AT\_END\_REF\_YR<**70** then agegrp=**1**;

if **70**=<BENE\_AGE\_AT\_END\_REF\_YR<**75** then agegrp=**2**;

if **75**=<BENE\_AGE\_AT\_END\_REF\_YR<**80** then agegrp=**3**;

if **80**=<BENE\_AGE\_AT\_END\_REF\_YR<**85** then agegrp=**4**;

if **85**=<BENE\_AGE\_AT\_END\_REF\_YR<**90** then agegrp=**5**;

if **90**=<BENE\_AGE\_AT\_END\_REF\_YR<**95** then agegrp=**6**;

if **95**=<BENE\_AGE\_AT\_END\_REF\_YR then agegrp=**7**;

if state\_code^=**65**;

if state\_code^=**40**;

if state\_code^=**48**;

if state\_code^=**00**;

bene\_zip\_cd = substr(bene\_zip\_cd,**1**,**5**);

run;

/\* f. CONVERT CHAR ZIP TO NUM\*/

data ffs&yr.;

set ffs&yr.;

zip=bene\_zip\_cd\***1**;

run;

data ffs\_mos&yr.;

set ffs&yr. ;

keep bene\_id sdod Member\_FFSMos&yr.01 Member\_FFSMos&yr.02 Member\_FFSMos&yr.03 Member\_FFSMos&yr.04 Member\_FFSMos&yr.05 Member\_FFSMos&yr.06 Member\_FFSMos&yr.07 Member\_FFSMos&yr.08 Member\_FFSMos&yr.09 Member\_FFSMos&yr.10 Member\_FFSMos&yr.11 Member\_FFSMos&yr.12 ;

run;

proc sort data = ffs\_mos&yr.;

by bene\_id;

run;

**%mend**;

%***ffs***(**11**);

%***ffs***(**12**);

%***ffs***(**13**);

%***ffs***(**14**);

/\* g. Create FFS dataset for all years combined containing only necessary variables for determining continuous eligibility later \*/

**data** ffs\_mos08thru14;

merge ffs\_mos08 ffs\_mos09 ffs\_mos10 ffs\_mos11 ffs\_mos12 ffs\_mos13 ffs\_mos14;

by bene\_id;

**run**;

/\* h. Save/read in combined year FFS dataset as ffs.ffs08thru14\*/

**data** ffs08thru14;

set ffs.ffs\_mos08thru14;

**run**;

/\* i. SAVE to FFS folder as ffs.ffs(yr)\*/

**data** ffs11;

set ffs.ffs11;

**run**;

**data** ffs12;

set ffs.ffs12;

**run**;

**data** ffs13;

set ffs.ffs13;

**run**;

**data** ffs14;

set ffs.ffs14;

**run**;

/\*FFS read in\*/

**data** ffs11;

set ffs.ffs11;

**run**;

**data** ffs12;

set ffs.ffs12;

**run**;

**data** ffs13;

set ffs.ffs13;

**run**;

**data** ffs14;

set ffs.ffs14;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*3. Merge Coronary Angio and FFS cohorts\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*a. keep only FFS eligible coronary angiographies - merge by month and then stack

Merge is an inner join - sexpndt1 comes from corangio (Pt B) dataset, i is an indicator added to FFS dataset

such that only corangios where patient had FFS coverage that month will be retained and patients who had FFS coverage but no

corangio will not be added in\*/

**%macro** ffsangio(yr);

proc sort data = ffs&yr.;

by bene\_id;

run;

data janffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**1**))

ffs&yr. (where=( member\_ffsmos&yr.01=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data febffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**2**))

ffs&yr. (where=( member\_ffsmos&yr.02=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data marffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**3**))

ffs&yr. (where=( member\_ffsmos&yr.03=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data aprffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**4**))

ffs&yr. (where=( member\_ffsmos&yr.04=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data mayffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**5**))

ffs&yr. (where=( member\_ffsmos&yr.05=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data junffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**6**))

ffs&yr. (where=( member\_ffsmos&yr.06=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data julffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**7**))

ffs&yr. (where=( member\_ffsmos&yr.07=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data augffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**8**))

ffs&yr. (where=( member\_ffsmos&yr.08=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data sepffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**9**))

ffs&yr. (where=( member\_ffsmos&yr.09=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data octffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**10**))

ffs&yr. (where=( member\_ffsmos&yr.10=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data novffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**11**))

ffs&yr. (where=( member\_ffsmos&yr.11=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data decffscath&yr.;

merge corangio&yr.3 (where=( sexpmonth=**12**))

ffs&yr. (where=( member\_ffsmos&yr.12=**1**));

by bene\_id ;

if sexpndt1;

if i;

run;

data eligangio&yr.;

set janffscath&yr. febffscath&yr. marffscath&yr. aprffscath&yr. mayffscath&yr. junffscath&yr. julffscath&yr. augffscath&yr. sepffscath&yr. octffscath&yr. novffscath&yr. decffscath&yr. ;

run;

**%mend**;

%***ffsangio***(**12**);

%***ffsangio***(**13**);

%***ffsangio***(**14**);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* 4. Merge in CCW with eligangio dataset and exclude benes who had CCW AMI in the year prior to index coronary angiography \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* a. Read in CCW files and bene xwalk for 2011 to 2013 \*/

**data** benexwalk;

set venk.bene\_bene\_xwalk\_2008\_2012 (rename=(bene\_id\_14049=bene\_id));

**run**;

**data** cc11;

set venk.mbsf\_cc2011;

**run**;

**data** cc12;

set venk.mbsf\_cc2012;

**run**;

**data** cc13;

set venk.mbsf\_cc2013;

**run**;

/\* b. Note - Bene\_id\_14049 variable links stein 20% Medicare data, bene\_id\_21074 variable links CCW data\*/

/\* c. Merge eligangio(yr) with benexwalk by bene\_id\_14049 \*/

**proc** **sort** data= benexwalk;

by bene\_id;

**run**;

/\*2012\*/

**proc** **sort** data= eligangio12;

by bene\_id;

**run**;

/\* merge in CCW link variable by bene\_id, i comes from angiography cohort, thus only merge in crosswalk variables for benes in cohort\*/

**data** eligangiox12;

merge eligangio12 benexwalk;

by bene\_id;

if i;

**run**;

/\* d. Separate by year, merge eligangio(yr) with cc(yr) by bene\_id\_21074, name cceligangio(yr) \*/

**proc** **sort** data = eligangiox12;

by bene\_id\_21074;

**run**;

**data** cc11;

set cc11 (rename=(bene\_id=bene\_id\_21074));

**run**;

**proc** **sort** data = cc11;

by bene\_id\_21074;

**run**;

/\*link CCW dataset to angio cohort by xwalk linking variable, i in angio cohort, thus only merge in CCW data to benes in cohort\*/

**data** cceligangio12;

merge eligangiox12 cc11;

by bene\_id\_21074;

if i; /\*if observation exists in cohort\*/

**run**;

/\* e. Define chronic conditions on whether claims were met (1,3)

https://www.resdac.org/cms-data/variables/acute-myocardial-infarction-end-year-indicator

Code Code value

0 Beneficiary did not meet claims criteria or have sufficient fee-for-service (FFS) coverage

1 Beneficiary met claims criteria but did not have sufficient FFS coverage

2 Beneficiary did not meet claims criteria but had sufficient FFS coverage

3 Beneficiary met claims criteria and had sufficient FFS coverage\*/

**data** cceligangio12;

set cceligangio12;

if ami =**1** or ami=**3** then xami=**1**; else xami=**0**;

**run**;

/\*2013\*/

**proc** **sort** data= eligangio13;

by bene\_id;

**run**;

**data** eligangiox13;

merge eligangio13 benexwalk;

by bene\_id;

if i;

**run**;

/\* d. Separate by year, merge eligangio(yr) with cc(yr) by bene\_id\_21074, name cceligangio(yr) \*/

**proc** **sort** data = eligangiox13;

by bene\_id\_21074;

**run**;

**data** cc12;

set cc12 (rename=(bene\_id=bene\_id\_21074));

**run**;

**proc** **sort** data = cc12;

by bene\_id\_21074;

**run**;

**data** cceligangio13;

merge eligangiox13 cc12;

by bene\_id\_21074;

if i;

**run**;

/\* e. Define chronic conditions on whether claims were met (1,3)

https://www.resdac.org/cms-data/variables/acute-myocardial-infarction-end-year-indicator

Code Code value

0 Beneficiary did not meet claims criteria or have sufficient fee-for-service (FFS) coverage

1 Beneficiary met claims criteria but did not have sufficient FFS coverage

2 Beneficiary did not meet claims criteria but had sufficient FFS coverage

3 Beneficiary met claims criteria and had sufficient FFS coverage\*/

**data** cceligangio13;

set cceligangio13;

if ami =**1** or ami=**3** then xami=**1**; else xami=**0**;

**run**;

/\*2014\*/

**proc** **sort** data= eligangio14;

by bene\_id;

**run**;

**data** eligangiox14;

merge eligangio14 benexwalk;

by bene\_id;

if i;

**run**;

/\* d. Separate by year, merge eligangio(yr) with cc(yr) by bene\_id\_21074, name cceligangio(yr) \*/

**proc** **sort** data = eligangiox14;

by bene\_id\_21074;

**run**;

**data** cc13;

set cc13 (rename=(bene\_id=bene\_id\_21074));

**run**;

**proc** **sort** data = cc13;

by bene\_id\_21074;

**run**;

**data** cceligangio14;

merge eligangiox14 cc13;

by bene\_id\_21074;

if i;

**run**;

/\* e. Define chronic conditions on whether claims were met (1,3)

https://www.resdac.org/cms-data/variables/acute-myocardial-infarction-end-year-indicator

Code Code value

0 Beneficiary did not meet claims criteria or have sufficient fee-for-service (FFS) coverage

1 Beneficiary met claims criteria but did not have sufficient FFS coverage

2 Beneficiary did not meet claims criteria but had sufficient FFS coverage

3 Beneficiary met claims criteria and had sufficient FFS coverage

\*/

**data** cceligangio14;

set cceligangio14;

if ami =**1** or ami=**3** then xami=**1**; else xami=**0**;

**run**;

/\* f. drop AMIs which met claims conditions\*/

**data** cceligangio12;

set cceligangio12;

if xami^=**1**;

**run**;

**data** cceligangio13;

set cceligangio13;

if xami^=**1**;

**run**;

**data** cceligangio14;

set cceligangio14;

if xami^=**1**;

**run**;

/\* g. stack cceligangio(yr)'s \*/

**data** eligangio;

set cceligangio12 cceligangio13 cceligangio14;

**run**;

**proc** **sort** data = eligangio;

by bene\_id;

**run**;

/\* h. save final file eligangio to temp folder\*/

**data** kaytemp.eligangio;

set eligangio;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 5. Keep only benes with continuous ffs eligibility 3 months prior to and 1 month after index coronary angio \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**data** ffs\_mos08thru14;

set ffs.ffs\_mos08thru14;

**run**;

/\* a. Merge eligangio dataset with combined year FFS dataset, drop if bene is not FFS eligible 3 months prior to and 1 month after index coronary angio or if they died within 100 days after coronary angio\*/

**data** conteligangio;

merge eligangio ffs\_mos08thru14;

by bene\_id;

if sexpndt1;

if i;

servmonth = put(sexpndt1,yymmn4.);

if servmonth=**1201** then do;

Member\_Mos = sum(of Member\_FFSMos1110 Member\_FFSMos1111 Member\_FFSMos1112 Member\_FFSMos1201 Member\_FFSMos1202 );

end;

if servmonth=**1202** then do;

Member\_Mos = sum(of Member\_FFSMos1111 Member\_FFSMos1112 Member\_FFSMos1201 Member\_FFSMos1202 Member\_FFSMos1203 );

end;

if servmonth=**1203** then do;

Member\_Mos = sum(of Member\_FFSMos1112 Member\_FFSMos1201 Member\_FFSMos1202 Member\_FFSMos1203 Member\_FFSMos1204 );

end;

if servmonth=**1204** then do;

Member\_Mos = sum(of Member\_FFSMos1201 Member\_FFSMos1202 Member\_FFSMos1203 Member\_FFSMos1204 Member\_FFSMos1205 );

end;

if servmonth=**1205** then do;

Member\_Mos = sum(of Member\_FFSMos1202 Member\_FFSMos1203 Member\_FFSMos1204 Member\_FFSMos1205 Member\_FFSMos1206 );

end;

if servmonth=**1206** then do;

Member\_Mos = sum(of Member\_FFSMos1203 Member\_FFSMos1204 Member\_FFSMos1205 Member\_FFSMos1206 Member\_FFSMos1207 );

end;

if servmonth=**1207** then do;

Member\_Mos = sum(of Member\_FFSMos1204 Member\_FFSMos1205 Member\_FFSMos1206 Member\_FFSMos1207 Member\_FFSMos1208 );

end;

if servmonth=**1208** then do;

Member\_Mos = sum(of Member\_FFSMos1205 Member\_FFSMos1206 Member\_FFSMos1207 Member\_FFSMos1208 Member\_FFSMos1209 );

end;

if servmonth=**1209** then do;

Member\_Mos = sum(of Member\_FFSMos1206 Member\_FFSMos1207 Member\_FFSMos1208 Member\_FFSMos1209 Member\_FFSMos1210 );

end;

if servmonth=**1210** then do;

Member\_Mos = sum(of Member\_FFSMos1207 Member\_FFSMos1208 Member\_FFSMos1209 Member\_FFSMos1210 Member\_FFSMos1211 );

end;

if servmonth=**1211** then do;

Member\_Mos = sum(of Member\_FFSMos1208 Member\_FFSMos1209 Member\_FFSMos1210 Member\_FFSMos1211 Member\_FFSMos1212);

end;

if servmonth=**1212** then do;

Member\_Mos = sum(of Member\_FFSMos1209 Member\_FFSMos1210 Member\_FFSMos1211 Member\_FFSMos1212 Member\_FFSMos1301);

end;

if servmonth=**1301** then do;

Member\_Mos = sum(of Member\_FFSMos1210 Member\_FFSMos1211 Member\_FFSMos1212 Member\_FFSMos1301 Member\_FFSMos1302 );

end;

if servmonth=**1302** then do;

Member\_Mos = sum(of Member\_FFSMos1211 Member\_FFSMos1212 Member\_FFSMos1301 Member\_FFSMos1302 Member\_FFSMos1303 );

end;

if servmonth=**1303** then do;

Member\_Mos = sum(of Member\_FFSMos1212 Member\_FFSMos1301 Member\_FFSMos1302 Member\_FFSMos1303 Member\_FFSMos1304 );

end;

if servmonth=**1304** then do;

Member\_Mos = sum(of Member\_FFSMos1301 Member\_FFSMos1302 Member\_FFSMos1303 Member\_FFSMos1304 Member\_FFSMos1305 );

end;

if servmonth=**1305** then do;

Member\_Mos = sum(of Member\_FFSMos1302 Member\_FFSMos1303 Member\_FFSMos1304 Member\_FFSMos1305 Member\_FFSMos1306 );

end;

if servmonth=**1306** then do;

Member\_Mos = sum(of Member\_FFSMos1303 Member\_FFSMos1304 Member\_FFSMos1305 Member\_FFSMos1306 Member\_FFSMos1307 );

end;

if servmonth=**1307** then do;

Member\_Mos = sum(of Member\_FFSMos1304 Member\_FFSMos1305 Member\_FFSMos1306 Member\_FFSMos1307 Member\_FFSMos1308 );

end;

if servmonth=**1308** then do;

Member\_Mos = sum(of Member\_FFSMos1305 Member\_FFSMos1306 Member\_FFSMos1307 Member\_FFSMos1308 Member\_FFSMos1309 );

end;

if servmonth=**1309** then do;

Member\_Mos = sum(of Member\_FFSMos1306 Member\_FFSMos1307 Member\_FFSMos1308 Member\_FFSMos1309 Member\_FFSMos1310 );

end;

if servmonth=**1310** then do;

Member\_Mos = sum(of Member\_FFSMos1307 Member\_FFSMos1308 Member\_FFSMos1309 Member\_FFSMos1310 Member\_FFSMos1311 );

end;

if servmonth=**1311** then do;

Member\_Mos = sum(of Member\_FFSMos1308 Member\_FFSMos1309 Member\_FFSMos1310 Member\_FFSMos1311 Member\_FFSMos1312);

end;

if servmonth=**1312** then do;

Member\_Mos = sum(of Member\_FFSMos1309 Member\_FFSMos1310 Member\_FFSMos1311 Member\_FFSMos1312 Member\_FFSMos1401);

end;

if servmonth=**1401** then do;

Member\_Mos = sum(of Member\_FFSMos1310 Member\_FFSMos1311 Member\_FFSMos1312 Member\_FFSMos1401 Member\_FFSMos1402 );

end;

if servmonth=**1402** then do;

Member\_Mos = sum(of Member\_FFSMos1311 Member\_FFSMos1312 Member\_FFSMos1401 Member\_FFSMos1402 Member\_FFSMos1403 );

end;

if servmonth=**1403** then do;

Member\_Mos = sum(of Member\_FFSMos1312 Member\_FFSMos1401 Member\_FFSMos1402 Member\_FFSMos1403 Member\_FFSMos1404 );

end;

if servmonth=**1404** then do;

Member\_Mos = sum(of Member\_FFSMos1401 Member\_FFSMos1402 Member\_FFSMos1403 Member\_FFSMos1404 Member\_FFSMos1405 );

end;

if servmonth=**1405** then do;

Member\_Mos = sum(of Member\_FFSMos1402 Member\_FFSMos1403 Member\_FFSMos1404 Member\_FFSMos1405 Member\_FFSMos1406 );

end;

if servmonth=**1406** then do;

Member\_Mos = sum(of Member\_FFSMos1403 Member\_FFSMos1404 Member\_FFSMos1405 Member\_FFSMos1406 Member\_FFSMos1407 );

end;

if servmonth=**1407** then do;

Member\_Mos = sum(of Member\_FFSMos1404 Member\_FFSMos1405 Member\_FFSMos1406 Member\_FFSMos1407 Member\_FFSMos1408 );

end;

if servmonth=**1408** then do;

Member\_Mos = sum(of Member\_FFSMos1405 Member\_FFSMos1406 Member\_FFSMos1407 Member\_FFSMos1408 Member\_FFSMos1409 );

end;

if servmonth=**1409** then do;

Member\_Mos = sum(of Member\_FFSMos1406 Member\_FFSMos1407 Member\_FFSMos1408 Member\_FFSMos1409 Member\_FFSMos1410 );

end;

if servmonth=**1410** then do;

Member\_Mos = sum(of Member\_FFSMos1407 Member\_FFSMos1408 Member\_FFSMos1409 Member\_FFSMos1410 Member\_FFSMos1411 );

end;

if servmonth=**1411** then do;

Member\_Mos = sum(of Member\_FFSMos1408 Member\_FFSMos1409 Member\_FFSMos1410 Member\_FFSMos1411 Member\_FFSMos1412);

end;

diedif = NDI\_DEATH\_DT-sexpndt1;

if Member\_Mos<**5** or (diedif ne **.** and diedif<**100**) /\*drop if not continuously eligible or if they died within 100 days after coronary angio\*/ then delete;

sexpmonth=month(sexpndt1);

**run**;

/\* b. keep only index procedures for all years\*/

**data** nodupeligangio;

set conteligangio;

**run**;

**proc** **sort** data = nodupeligangio ;

by bene\_id sexpndt1;

**run**;

**proc** **sort** data = nodupeligangio nodupkey;

by bene\_id ;

**run**;

/\* c. Save final file nodupeligangio in temp folder\*/

**data** kaytemp.nodupeligangio;

set nodupeligangio;

**run**;

/\*read in\*/

**data** nodupeligangio;

set kaytemp.nodupeligangio;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 6. Read in and create datasets for Medpar exclusion procedures (Same as in part B but from Medpar file)\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* a. read in medpar files, rename dgnscd vars so all years match\*/

**data** medp11;

set vingtMP.med11p20 (rename=(dgnscd1=dgns\_cd1 dgnscd2=dgns\_cd2 dgnscd3=dgns\_cd3 dgnscd4=dgns\_cd4

dgnscd5=dgns\_cd5 dgnscd6=dgns\_cd6 dgnscd7=dgns\_cd7 dgnscd8=dgns\_cd8 dgnscd9=dgns\_cd9 dgnscd10=dgns\_cd10

dgnscd11=dgns\_cd11 dgnscd12=dgns\_cd12 dgnscd13=dgns\_cd13 dgnscd14=dgns\_cd14 dgnscd15=dgns\_cd15 dgnscd16=dgns\_cd16

dgnscd17=dgns\_cd17 dgnscd18=dgns\_cd18 dgnscd19=dgns\_cd19 dgnscd20=dgns\_cd20 dgnscd21=dgns\_cd21 dgnscd22=dgns\_cd22

dgnscd23=dgns\_cd23 dgnscd24=dgns\_cd24 dgnscd25=dgns\_cd25 medparid = medpar\_id));

**run**;

**data** medp12;

set vingtMP.med12p20 (rename=(dgns\_cd\_cnt=dgnscnt));

**run**;

**data** medp13;

set vingtMP.med13p20 (rename=(dgns\_1\_cd=dgns\_cd1 dgns\_2\_cd=dgns\_cd2 dgns\_3\_cd=dgns\_cd3 dgns\_4\_cd=dgns\_cd4

dgns\_5\_cd=dgns\_cd5 dgns\_6\_cd=dgns\_cd6 dgns\_7\_cd=dgns\_cd7 dgns\_8\_cd=dgns\_cd8 dgns\_9\_cd=dgns\_cd9 dgns\_10\_cd=dgns\_cd10

dgns\_11\_cd=dgns\_cd11 dgns\_12\_cd=dgns\_cd12 dgns\_13\_cd=dgns\_cd13 dgns\_14\_cd=dgns\_cd14 dgns\_15\_cd=dgns\_cd15 dgns\_16\_cd=dgns\_cd16

dgns\_17\_cd=dgns\_cd17 dgns\_18\_cd=dgns\_cd18 dgns\_19\_cd=dgns\_cd19 dgns\_20\_cd=dgns\_cd20 dgns\_21\_cd=dgns\_cd21 dgns\_22\_cd=dgns\_cd22

dgns\_23\_cd=dgns\_cd23 dgns\_24\_cd=dgns\_cd24 dgns\_25\_cd=dgns\_cd25 dgns\_cd\_cnt=dgnscnt));

**run**;

**data** medp14;

set vingtMP.med14p20 (rename=(dgns\_1\_cd=dgns\_cd1 dgns\_2\_cd=dgns\_cd2 dgns\_3\_cd=dgns\_cd3 dgns\_4\_cd=dgns\_cd4

dgns\_5\_cd=dgns\_cd5 dgns\_6\_cd=dgns\_cd6 dgns\_7\_cd=dgns\_cd7 dgns\_8\_cd=dgns\_cd8 dgns\_9\_cd=dgns\_cd9 dgns\_10\_cd=dgns\_cd10

dgns\_11\_cd=dgns\_cd11 dgns\_12\_cd=dgns\_cd12 dgns\_13\_cd=dgns\_cd13 dgns\_14\_cd=dgns\_cd14 dgns\_15\_cd=dgns\_cd15 dgns\_16\_cd=dgns\_cd16

dgns\_17\_cd=dgns\_cd17 dgns\_18\_cd=dgns\_cd18 dgns\_19\_cd=dgns\_cd19 dgns\_20\_cd=dgns\_cd20 dgns\_21\_cd=dgns\_cd21 dgns\_22\_cd=dgns\_cd22

dgns\_23\_cd=dgns\_cd23 dgns\_24\_cd=dgns\_cd24 dgns\_25\_cd=dgns\_cd25 dgns\_cd\_cnt=dgnscnt));

**run**;

/\*save to temp folder\*/

**data** kaytemp.medp11;

set medp11;

**run**;

**data** kaytemp.medp12;

set medp12;

**run**;

**data** kaytemp.medp13;

set medp13;

**run**;

**data** kaytemp.medp14;

set medp14;

**run**;

/\*read in\*/

**data** medp11;

set kaytemp.medp11;

**run**;

**data** medp12;

set kaytemp.medp12;

**run**;

**data** medp13;

set kaytemp.medp13;

**run**;

**data** medp14;

set kaytemp.medp14;

**run**;

/\* b. Output separate datasets for AMI, valve procedures, and stress tests found in any of the 25 diagnosis codes\*/

/\*2011-2014\*/

**%macro** mp(yr);

/\*ami\*/

data mpami&yr.;

do until (eof);

set medp&yr. end=eof;

array DX (**25**) DGNS\_CD1 - DGNS\_CD25;

do i=**1** to dgnscnt;

if DX{i} in (/\*ami\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090')

then do ;

dgn=dx{i}; output;

end;

end;

end;

run;

/\*valve\*/

data mpvalve&yr.;

do until (eof);

set medp&yr. end=eof;

array DX (**25**) DGNS\_CD1 - DGNS\_CD25;

do i=**1** to dgnscnt;

if DX{i} in (/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671') then do ;

dgn=dx{i}; output;

end;

end;

end;

run;

/\*stress\*/

data mpstress&yr.;

do until (eof);

set medp&yr. end=eof;

array DX (**25**) DGNS\_CD1 - DGNS\_CD25;

do i=**1** to dgnscnt;

if DX{i} in (/\*stress dx\*/ '8941', '8942', '8943', '8944') then do ;

dgn=dx{i}; output;

end;

end;

end;

run;

/\* c. Assign variable names to output codes, keep only merging variables and newly assigned diagnosis variable names \*/

data mpami&yr. ;

set mpami&yr. ;

if dgn in (/\*ami\*/ '41001','41011','41021','41031','41041','41051','41061','41071','41081','41091',

'41000','41010','41020','41030','41040','41050','41060','41070','41080','41090')

then mpami=**1**; else mpami=**0**; /\*Note - every obs in these datasets should be 1 if done correctly\*/

run;

data mpami&yr. ;

set mpami&yr. ;

keep bene\_id medpar\_id mpami;

run;

data mpvalve&yr. ;

set mpvalve&yr. ;

if dgn in (/\*valve procedures\*/

'3500','3501','3502','3503','3504','3505','3506','3507','3508',

'3509','3510','3511','3512','3513','3514','3520','3521','3522','3523','3524','3525',

'3526','3527','3528','3596','3597','3599',

/\*valve disorders\*/

/\*(single)\*/ '4240', '4241', '4242', '4243', '42490', '42491', '42499',

/\*(combined mitral/aortic)\*/ '3960', '3961', '3962', '3963', '3969',

/\*(tricuspid valve and pulmonary)\*/ '3970', '3971',

/\*(rheumatic aortic valve)\*/ '3950', '3951', '3952', '3959',

/\*valve prosthesis\*/ '99602', '99671') then mpvalve=**1**; else mpvalve=**0**;

run;

data mpvalve&yr. ;

set mpvalve&yr. ;

keep bene\_id medpar\_id mpvalve;

run;

data mpstress&yr.;

set mpstress&yr.;

if dgn in (/\*stress dx\*/ '8941', '8942', '8943', '8944') then mpstress=**1**; else mpstress=**0**;

run;

data mpstress&yr.;

set mpstress&yr.;

keep bene\_id medpar\_id mpstress;

run;

proc sort data = medp&yr.;

by bene\_id medpar\_id;

run;

proc sort data = mpami&yr. nodupkey;

by bene\_id medpar\_id mpami;

run;

proc sort data = mpami&yr. ;

by bene\_id medpar\_id;

run;

proc sort data = mpvalve&yr. nodupkey;

by bene\_id medpar\_id mpvalve;

run;

proc sort data = mpvalve&yr.;

by bene\_id medpar\_id;

run;

proc sort data = mpstress&yr. nodupkey;

by bene\_id medpar\_id mpstress;

run;

proc sort data = mpstress&yr.;

by bene\_id medpar\_id;

run;

/\* d. Merge ami, valve, stress datasets back to original medpar dataset for each year to add in new indicator diagnosis variables\*/

data mp&yr.;

merge medp&yr. mpami&yr. mpvalve&yr. mpstress&yr.;

by bene\_id medpar\_id;

run;

**%mend**;

%***mp***(**11**);

%***mp***(**12**);

%***mp***(**13**);

%***mp***(**14**);

/\*save to temp folder\*/

**data** kaytemp.mpdx11;

set mp11;

**run**;

**data** kaytemp.mpdx12;

set mp12;

**run**;

**data** kaytemp.mpdx13;

set mp13;

**run**;

**data** kaytemp.mpdx14;

set mp14;

**run**;

/\*Read in Medpar DX \*/

**data** mp11;

set kaytemp.mpdx11 ;

**run**;

**data** mp12;

set kaytemp.mpdx12;

**run**;

**data** mp13;

set kaytemp.mpdx13;

**run**;

**data** mp14;

set kaytemp.mpdx14;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 7. ER Exclusions cohort construction - want to drop patients admitted or transferred from an ED, interested in elective coronary angios only, not emergency ones \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* a. read in outpatient files \*/

**data** op12revs;

set vingtop.op12revs;

**run**;

**data** op13revs;

set vingtop.op13rev;

**run**;

**data** op14revs;

set vingtop.op14revs;

**run**;

/\*2012 - 2014\*/

/\* b. identify ED outpatient visits\*/

**%macro** er(yr);

data optrans&yr. (keep=bene\_id clm\_id REV\_CNTR /\*2011\*/ rev\_dt);

set op&yr.revs (rename=(/\*2012\*/ REV\_CNTR\_DT=rev\_dt /\*2010 srev\_dt=rev\_dt\*/)) ;

if REV\_CNTR in ('0450','0451','0452','0453','0454','0455','0456','0457','0458','0459', '0981')

then output optrans&yr.;

run;

/\* c. identify inpatient transfers from medpar file \*/

data mptrans&yr. (keep = bene\_id MEDPAR\_ID DSTNTNCD /\*2011\*/ ADMSNDT DSCHRGDT trans );

set vingtmp.med&yr.p20 (rename=(/\*2010 and 2011 medparid=medpar\_id \*/ /\*2010 SADMSNDT=admsndt SDSCHRGDT=dschrgdt \*/ /\*2012\*/ admsn\_dt=admsndt dschrg\_dt=dschrgdt DSCHRG\_DSTNTN\_CD=dstntncd ));

if DSTNTNCD in ('02','05','09') then trans=**1**; else trans=**0**;

run;

/\* d. identify inpatient transfers from outpatient ED \*/

proc sql;

create table mpoptrans&yr. as

select p.admsndt, p.bene\_id, p.trans, p.medpar\_id, w.clm\_id, w.bene\_id, w.rev\_dt

from optrans&yr. as w, mptrans&yr. as p

where (p.trans=**1**) and (w.bene\_id = p.bene\_id) and (**0**<=p.admsndt - w.rev\_dt<=**1**)

;

quit;

/\* e. merge OP ED transfers into latest full medpar file\*/

data mpoptrans&yr.;

set mpoptrans&yr. (drop=admsndt);

optrans=**1**;

run;

proc sort data = mpoptrans&yr.;

by bene\_id medpar\_id;

run;

data med&yr.p20;

set mp&yr. (rename=(/\*2010 and 2011 medparid=medpar\_id\*/ /\*2010 SADMSNDT=admsndt SDSCHRGDT=dschrgdt \*/ /\*2012\*/ admsn\_dt=admsndt dschrg\_dt=dschrgdt DSCHRG\_DSTNTN\_CD=dstntncd ER\_CHRG\_AMT=er\_amt));

run;

proc sort data = med&yr.p20;

by bene\_id medpar\_id;

run;

data mpop&yr.;

merge mpoptrans&yr. med&yr.p20;

by bene\_id medpar\_id;

run;

proc sort data = mpop&yr. nodupkey;

by bene\_id medpar\_id;

run;

/\* f. create dataset containing only merging variables and new variables for inpatient ER or outpatient ER transfers\*/

data mpop&yr.;

set mpop&yr.;

if optrans=**.** then optrans=**0**;

if ER\_AMT=**0** then ER=**0**; else ER=**1**;

if er=**1** or optrans=**1** then ers=**1**; else ers=**0**;

run;

proc freq data = mpop&yr.;

tables optrans er ers;

run;

data mpop&yr.;

set mpop&yr.;

keep bene\_id ADMSNDT MEDPAR\_ID optrans rev\_dt er optrans ers ;

run;

**%mend**;

%***er***(**12**);

%***er***(**13**);

%***er***(**14**);

/\* g. keep only ER exclusions and stack separate year datasets into one to save to iac folder\*/

**data** mpop;

set mpop12 mpop13 mpop14;

if ers^=**0**;

**run**;

**data** mpop;

set mpop;

admsnyear=year(admsndt);

if admsnyear^=**2009**;

if admsnyear^=**2010**;

if admsnyear^=**2011**;

**run**;

**data** iac.ffr\_er\_exclsns12\_13\_14;

set mpop;

**run**;

/\*read in ER exclusions dataset\*/

**data** mpop;

set iac.ffr\_er\_exclsns12\_13\_14;

**run**;

/\* h. remove duplicate observations with the same admission date break stacked dataset back up by year of admission\*/

**proc** **sort** data = mpop nodupkey;

by bene\_id admsndt;

**run**;

**data** mpop12;

set mpop;

if admsnyear^ne **2012**;

**run**;

**data** mpop13;

set mpop;

if admsnyear^ne **2013**;

**run**;

**data** mpop14;

set mpop;

if admsnyear^ne **2014**;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 8. Assemble all Part B, Medpar, ER exclusions, outcomes (revasc and FFR), and covariates (prior stress) to be merged into eligible bene cohort, drop exclusions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* a. Make separate datasets for each Downstream/Concurrent/Past Procedures from coronary angio cohort (from Part B file) - PCI, CABG, FFR, ami, stress, and valve procedures

Because coronary angio cohort is assembled by line (angio procedure), and because we need to determine other procs within a certain time frame of that procedure, SQL was used to output

datasets where the expense data of a procedure of interest occurs within the original expense data of the coronary angio. Once indicators for the given procedure of interest are

created, still attached to the bene\_id and CA expense date, there will be merged back in to the original cohort, thus allowing us to drop or flag those indicators in a wide format\*/

**%macro** dx(yr);

/\*pci\*/

data pci&yr.;

set angio&yr. (rename=(sexpndt1=pci\_dt));

if pci^=**0**;

keep bene\_id pci\_dt pci;

run;

proc sort data = pci&yr. nodupkey;

by bene\_id pci\_dt;

run;

/\*ffr\*/

data ffr&yr.;

set angio&yr. (rename=(sexpndt1=ffr\_dt));

if ffr^=**0**;

keep bene\_id ffr\_dt ffr;

run;

proc sort data = ffr&yr. nodupkey;

by bene\_id ffr\_dt;

run;

/\*cabg - will be used to exclude patient with cabg 1 yr prior to angio and identify revascularization 30 days after\*/

data cabg&yr.;

set angio&yr. (rename=(sexpndt1=cabg\_dt));

if cabg^=**0**;

keep bene\_id cabg\_dt cabg;

run;

proc sort data = cabg&yr. nodupkey;

by bene\_id cabg\_dt;

run;

/\*bstress\*/

data bstress&yr.;

set angio&yr. (rename=(sexpndt1=stress\_dt));

if pbstress =**1** or stress =**1** then bstress=**1**; else bstress=**0**;

if bstress^=**0**;

keep bene\_id stress\_dt bstress;

run;

proc sort data = bstress&yr. nodupkey;

by bene\_id stress\_dt;

run;

/\*bami\*/

data bami&yr.;

set angio&yr. (rename=(sexpndt1=ami\_dt));

if pbami^=**0**;

keep bene\_id ami\_dt pbami;

run;

proc sort data = bami&yr. nodupkey;

by bene\_id ami\_dt;

run;

/\*bvalve\*/

data bvalve&yr.;

set angio&yr. (rename=(sexpndt1=valve\_dt));

if pbvalve^=**0**;

keep bene\_id valve\_dt pbvalve;

run;

proc sort data = bvalve&yr. nodupkey;

by bene\_id valve\_dt;

run;

/\* b. Make separate datasets for each Downstream/Concurrent/Past procedure/diagnosis from medpar exclusion cohort - ami, stress, valve \*/

/\*mpami\*/

data mp11;

set mp11 (rename=(admsndt=admsn\_dt));

run;

data mpami&yr.;

set mp&yr. (rename=(admsn\_dt=ami\_dt));

if mpami^ne **1**;

keep bene\_id ami\_dt mpami;

run;

proc sort data = mpami&yr. nodupkey;

by bene\_id ami\_dt;

run;

/\*mpstress\*/

data mpstress&yr.;

set mp&yr. (rename=(admsn\_dt=stress\_dt));

if mpstress^ne **1**;

keep bene\_id stress\_dt mpstress;

run;

proc sort data = mpstress&yr. nodupkey;

by bene\_id stress\_dt;

run;

/\*mpvalve\*/

data mpvalve&yr.;

set mp&yr. (rename=(admsn\_dt=valve\_dt));

if mpvalve^ne **1**;

keep bene\_id valve\_dt mpvalve;

run;

proc sort data = mpvalve&yr. nodupkey;

by bene\_id valve\_dt;

run;

/\* c. Combine like indicators from Medpar and Part B - stress, ami, valve, remove duplicates \*/

/\*stress\*/

data stress&yr.;

set bstress&yr. mpstress&yr.;

run;

proc sort data = stress&yr. nodupkey;

by bene\_id stress\_dt;

run;

/\*ami\*/

data ami&yr.;

set bami&yr. mpami&yr.;

run;

proc sort data = ami&yr. nodupkey;

by bene\_id ami\_dt;

run;

/\*valve\*/

data valve&yr.;

set bvalve&yr. mpvalve&yr.;

run;

proc sort data = valve&yr. nodupkey;

by bene\_id valve\_dt;

run;

**%mend**;

%***dx***(**11**);

%***dx***(**12**);

%***dx***(**13**);

%***dx***(**14**);

/\* d. output dataset of ER Exclusions from latest eligible angio cohort (nodupeligangio) if ER visit happened 3 days before or 1 day after angio, will be associated with the expense date of coronary angio \*/

**proc** **sql**;

create table noers as

select p.bene\_id, p.admsndt, p.ers, w.bene\_id, w.sexpndt1

from nodupeligangio as w, mpop as p

where (w.bene\_id = p.bene\_id) and

(-**1**<w.sexpndt1 - p.admsndt<=**3**)

;

**quit**;

**data** noers;

set noers;

diffdt= sexpndt1 - admsndt;

diffdtabs= abs(admsndt - sexpndt1);

**run**;

**proc** **sort** data = noers out=noers\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** noers2;

set noers\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id admsndt ers sexpndt1 diffdt diffdtabs ;

**run**;

/\*double checked - there are no duplicates to drop\*/

**proc** **sort** data = noers2 nodupkey;

by bene\_id sexpndt1;

**run**;

**proc** **sort** data = nodupeligangio;

by bene\_id sexpndt1;

**run**;

/\* e. merge identified ER exclusions into original eligible cohort and drop ER exclusions, identified ER exclusions (indicated by 'ers') will be merged back by the bene\_id and coronary angio they were associated with and subsequently dropped\*/

**data** noers3;

merge noers2 nodupeligangio;

by bene\_id sexpndt1;

**run**;

**data** noers4;

set noers3;

if ers^=**1**;

**run**;

/\* f. Repeat steps d. and e. for AMI 1 year prior, CABG 1 year prior, and valve procedures 1 year prior \*/

/\*AMIs are from part B and Medpar 1\*/

**data** ami11thru14;

set ami11 ami12 ami13 ami14;

**run**;

/\*output amis from either part B or Medpar that occured within a year prior to angio\*/

**proc** **sql**;

create table noamier as

select p.ami\_dt, p.bene\_id, p.mpami, p.pbami, w.bene\_id, w.sexpndt1

from noers4 as w, ami11thru14 as p

where (w.bene\_id = p.bene\_id) and

(**0**<=w.sexpndt1 -p.ami\_dt <=**365**)

;

**quit**;

**data** noamier;

set noamier;

diffdt= sexpndt1 - ami\_dt ;

diffdtabs= abs(ami\_dt - sexpndt1);

**run**;

**proc** **sort** data = noamier out=noamier\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

/\*if multiple amis a year remove duplicates\*/

**data** noamier2;

set noamier\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 mpami pbami ami\_dt;

if mpami ne **1** then mpami=**0**;

**run**;

/\*drop output indicators from original dataset to merge\*/

**data** noers4;

set noers4;

drop pbami mpami;

**run**;

/\* merge identified AMI exclusions into last eligible cohort and drop AMI exclusions\*/

**data** noamier3;

merge noamier2 noers4;

by bene\_id sexpndt1;

if pbami ne **1** then pbami=**0**;

if mpami ne **1** then mpami=**0**;

**run**;

**data** noamier4;

set noamier3;

if pbami^=**1**;

if mpami^=**1**;

**run**;

/\* save to temp folder as noamier4\*/

**data** kaytemp.noamier4;

set noamier4;

**run**;

/\*CABG 1 year prior\*/

**data** cabg11thru14;

set cabg11 cabg12 cabg13 cabg14;

**run**;

**proc** **sql**;

create table nocabg as

select p.cabg\_dt, p.bene\_id, p.cabg, w.bene\_id, w.sexpndt1

from noamier4 as w, cabg11thru14 as p

where (w.bene\_id = p.bene\_id) and

(**0**<=w.sexpndt1 -p.cabg\_dt <=**365**)

;

**quit**;

**data** nocabg;

set nocabg;

diffdt= sexpndt1 - cabg\_dt ;

diffdtabs= abs(cabg\_dt - sexpndt1);

**run**;

**proc** **sort** data = nocabg out=nocabg\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** nocabg2;

set nocabg\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 cabg cabg\_dt;

**run**;

**data** noamier4;

set noamier4;

drop cabg;

**run**;

**proc** **sort** data = nocabg2;

by bene\_id sexpndt1;

**run**;

**data** nocabg3;

merge nocabg2 noamier4;

by bene\_id sexpndt1;

**run**;

**data** nocabg4;

set nocabg3;

if cabg^=**1**;

**run**;

/\*Valve 1 year prior\*/

**data** valve11thru14;

set valve11 valve12 valve13 valve14;

**run**;

**proc** **sql**;

create table novalve as

select p.valve\_dt, p.bene\_id, p.pbvalve, p.mpvalve, w.bene\_id, w.sexpndt1

from nocabg4 as w, valve11thru14 as p

where (w.bene\_id = p.bene\_id) and

(**0**<=w.sexpndt1 -p.valve\_dt <=**365**)

;

**quit**;

**data** novalve;

set novalve;

diffdt= sexpndt1 - valve\_dt ;

diffdtabs= abs(valve\_dt - sexpndt1);

**run**;

**proc** **sort** data = novalve out=novalve\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** novalve2;

set novalve\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 pbvalve mpvalve valve\_dt;

**run**;

**data** nocabg4;

set nocabg4;

drop pbvalve mpvalve;

**run**;

**proc** **sort** data = novalve2;

by bene\_id sexpndt1;

**run**;

**data** novalve3;

merge novalve2 nocabg4;

by bene\_id sexpndt1;

if pbvalve ne **1** then pbvalve=**0**;

if mpvalve ne **1** then mpvalve=**0**;

**run**;

**data** novalve4;

set novalve3;

if mpvalve^=**1**;

if pbvalve^=**1**;

**run**;

/\* g. Create dataset of variable for revasc if bene had PCI or CABG 30 days after angio\*/

/\*missing 2015 is not a problem for revasc occurring after 2014, revasc within 30 days, already dropped last 3 mos of 2014 when determining continuous FFS eligibility\*/

/\* remove duplicate indicators from last version of eligible cohort that the new variable dataset containing those indicators will be merged with\*/

**data** novalve4;

set novalve4;

drop cabg pci ffr pbstress;

**run**;

/\* stack datasets for PCI and CABG for all years to make a combined-year revasc dataset\*/

**data** revasc12thru14;

set pci12 pci13 pci14 cabg12 cabg13 cabg14;

**run**;

**data** revasc12thru14;

set revasc12thru14;

if pci\_dt ne **.** then revasc\_dt = pci\_dt;

if cabg\_dt ne **.** then revasc\_dt = cabg\_dt;

revasc=**1**;

if cabg=**.** then cabg=**0**;

if pci=**.** then pci=**0**;

format revasc\_dt date9.;

**run**;

/\*remove duplicates on same day\*/

**proc** **sort** data = revasc12thru14 nodupkey;

by bene\_id revasc\_dt;

**run**;

/\* h. output dataset of downstream revasc from latest eligible angio cohort (novalve4) if revasc happened within 30 days after angio \*/

**proc** **sql**;

create table revasc as

select p.revasc\_dt, p.bene\_id, p.revasc, p.cabg, p.pci, w.bene\_id, w.sexpndt1

from novalve4 as w, revasc12thru14 as p

where (w.bene\_id = p.bene\_id) and

(**0**<=p.revasc\_dt - w.sexpndt1 <=**30**)

;

**quit**;

**data** revasc;

set revasc;

diffdt= revasc\_dt - sexpndt1 ;

diffdtabs= abs(revasc\_dt - sexpndt1);

**run**;

**proc** **sort** data = revasc out=revasc\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** revasc2;

set revasc\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 revasc pci cabg revasc\_dt;

**run**;

**proc** **sort** data = revasc2;

by bene\_id sexpndt1;

**run**;

/\* i. merge identified revasc into last eligible cohort\*/

**data** revasc3;

merge revasc2 novalve4;

by bene\_id sexpndt1;

if cabg=**.** then cabg=**0**;

if pci=**.** then pci=**0**;

if revasc=**.** then revasc=**0**;

**run**;

/\* j. Repeat steps g. through i. for creating a variable for FFR - if FFR occurred 3 days before or after angio, and for stress 90 days prior to angio \*/

**data** ffr11thru14;

set ffr11 ffr12 ffr13 ffr14 ;

**run**;

**proc** **sql**;

create table ffr as

select p.ffr\_dt, p.bene\_id, p.ffr, w.bene\_id, w.sexpndt1

from revasc3 as w, ffr11thru14 as p

where (w.bene\_id = p.bene\_id) and

(-**3**<=p.ffr\_dt - w.sexpndt1 <=**3**)

;

**quit**;

**data** ffr;

set ffr;

diffdt= ffr\_dt - sexpndt1 ;

diffdtabs= abs(ffr\_dt - sexpndt1);

**run**;

**proc** **sort** data = ffr out=ffr\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** ffr2;

set ffr\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 ffr ffr\_dt;

**run**;

**proc** **sort** data = ffr2;

by bene\_id sexpndt1;

**run**;

**data** ffr3;

merge ffr2 revasc3;

by bene\_id sexpndt1;

if ffr=**.** then ffr=**0**;

**run**;

**data** kaytemp.ffr3;

set ffr3;

**run**;

**data** ffr3;

set kaytemp.ffr3;

**run**;

/\*STRESS 90 DAYS PRIOR TO CA\*/

**data** stress11thru14;

set stress11 stress12 stress13 stress14;

**run**;

**proc** **sql**;

create table stress90 as

select p.stress\_dt, p.bene\_id, p.bstress, p.mpstress, w.bene\_id, w.sexpndt1

from ffr3 as w, stress11thru14 as p

where (w.bene\_id = p.bene\_id) and

(**0**<=w.sexpndt1 -p.stress\_dt <=**90**)

;

**quit**;

**data** stress90;

set stress90;

diffdt= sexpndt1 - stress\_dt ;

diffdtabs= abs(stress\_dt - sexpndt1);

**run**;

**proc** **sort** data = stress90 out=stress90\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

**run**;

**data** stress902;

set stress90\_s;

by bene\_id sexpndt1 diffdtabs diffdt;

if first.sexpndt1;

keep bene\_id sexpndt1 bstress mpstress stress\_dt;

**run**;

**proc** **sort** data = stress902;

by bene\_id sexpndt1;

**run**;

**data** stress903;

merge stress902 ffr3;

by bene\_id sexpndt1;

if bstress ne **1** then bstress=**0**;

if mpstress ne **1** then mpstress=**0**;

if mpstress =**1** or bstress =**1** then stress=**1**; else stress=**0**;

**run**;

/\* k. save in final folder as finalcohort\*/

**data** kayfinal.finalcohort;

set stress903;

**run**;

/\*read in final dataset\*/

**data** finalcohort;

set kayfinal.finalcohort;

**run**;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 9. Create variable for geographic regions, tweak covariates, perform analysis \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* a. define regions \*/

**data** finalregion;

set finalcohort;

if prvstate in ('07','20','22','30','41','47') then region=**1**;

if prvstate in ('31','33','39') then region=**2**;

if prvstate in ('15','14','23','36','52') then region=**3**;

if prvstate in ('16','17','24','26','28','35','43') then region=**4**;

if prvstate in ('08','09','10','11','21','34','42','49','51') then region=**5**;

if prvstate in ('01','18','25','44') then region=**6**;

if prvstate in ('04','19','37','45') then region=**7**;

if prvstate in ('03','06','13','32','27','46','29','53') then region=**8**;

if prvstate in ('02','05','12','38','50') then region=**9**;

if prvstate^=**65**;

if prvstate^=**40**;

if prvstate^=**48**;

if prvstate^=**00**;

**run**;

/\* b. Combine stress or ffr into one variable \*/

**data** finalregion;

set finalregion;

if stress=**1** or ffr=**1** then stressorffr=**1**; else stressorffr=**0**;

**run**;

/\* c. Exploratory Descriptive Analysis\*/

**proc** **freq** data = finalregion;

tables ffr\*black ffr\*rti\_race\_cd ffr\*bene\_sex\_ident\_cd ffr\*agegrp ffr\*stress;

**run**;

**proc** **freq** data = finalregion;

tables refyr\*corangio;

**run**;

**proc** **freq** data = finalregion;

tables revasc\*ffr revasc\*stressorffr;

**run**;

**proc** **sort** data = finalregion;

by region;

**run**;

**proc** **means** data = finalregion nway noprint;

class region;

var corangio pci cabg revasc stressorffr stress ffr ;

output out = ffrregion sum = corangio pci cabg revasc stressorffr stress ffr;

**run**;

**proc** **freq** data = finalregion;

tables BENE\_SEX\_IDENT\_CD RTI\_RACE\_CD;

**run**;

**proc** **means** data = finalregion;

var BENE\_AGE\_AT\_END\_REF\_YR;

**run**;

/\*save to final folder\*/

**data** kayfinal.finalregion;

set finalregion;

**run**;

/\*read in\*/

**data** finalregion;

set kayfinal.finalregion;

**run**;

/\* d. Newest final model with sex ref = male instrad and agegrp = 65-75, 75-85, >85 instead\*/

**data** finalregion2;

set finalregion;

if **65**=< BENE\_AGE\_AT\_END\_REF\_YR <**75** then agegrp2=**1**;

if **75**=< BENE\_AGE\_AT\_END\_REF\_YR <**85** then agegrp2=**2**;

if **85**=< BENE\_AGE\_AT\_END\_REF\_YR then agegrp2=**3**;

**run**;

/\* e. Final Descriptive Analysis \*/

**proc** **freq** data = finalregion2;

tables ffr\*agegrp ffr\*agegrp2 ffr\*bene\_sex\_ident\_cd;

**run**;

**proc** **freq** data = finalregion2;

tables ffr\*region ffr\*bene\_sex\_ident\_cd ffr\*black;

**run**;

**proc** **freq** data = finalregion2;

tables revasc revasc\*pci revasc\*cabg revasc\*ffr;

**run**;

**proc** **means** data = finalregion2;

var ffr;

class region;

output out=meanregion mean=ffravg;

**run**;

**proc** **means** data = meanregion;

var ffravg;

**run**;

**proc** **freq** data = finalregion2;

tables BENE\_SEX\_IDENT\_CD black ffr revasc pci cabg ffr\*revasc

revasc\*stress\*ffr agegrp2\*ffr BENE\_SEX\_IDENT\_CD\*ffr black\*ffr stress\*ffr region\*ffr;

**run**;

/\* f. Exploratory Analytical Models \*/

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stress (ref='0') agegrp (ref='1') region (ref='1') revasc (ref='0')/ param=ref;

model ffr (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stress revasc /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stressorffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model revasc (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stressorffr /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stress (ref='0') ffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model revasc (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stress ffr /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stressorffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model revasc (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stressorffr stressorffr\*region /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stressorffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model cabg (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stressorffr /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') stressorffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model pci (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black stressorffr /expb;

**run**;

**proc** **logistic** data = finalregion ;

class BENE\_SEX\_IDENT\_CD (ref='2') black (ref='0') ffr (ref='0') agegrp (ref='1') region (ref='1')/ param=ref;

model pci (event='1') = BENE\_SEX\_IDENT\_CD region agegrp black ffr /expb;

**run**;

/\* g. Final Analytical Model \*/

**proc** **logistic** data = finalregion2 ;

class BENE\_SEX\_IDENT\_CD (ref='1') black (ref='0') stress (ref='0') agegrp2 (ref='1') region (ref='1') revasc (ref='1')/ param=ref;

model ffr (event='1') = BENE\_SEX\_IDENT\_CD region agegrp2 black stress revasc /expb;

**run**;