Derived from the Rand Tech Report part 3.2 (2005) in Appendix, to estimate total cost of the FIM measurements:

$$logcost$$

$$= \begin{cases} f_1(motor\ score) + f_2(cog\ score) + f_3(age) + \gamma\{tier\}, & if\ no\ missing \quad (1.1) \\ f_1(motor\ score_{-i}) + f_2(cog\ score) + f_3(age) + \gamma_1\{tier\} & \\ +\gamma_2\{missing_{-i}\} + \beta_1 \times motor\ score_i + \beta_2 \times missing_i, & o.w \quad (1.2) \end{cases}$$

in which

f_1, f_2, and f_3 are GAM fits (cubic splines, df = 3, knots = 4),

$motor\ score = weigt_1 \times var_1 + \cdots + weigt_n \times var_n$,

$\{tier\}$ denotes the three levels of comorbidity dummy variables (tier = 0,1,2,3),

$motor\ score_{-i}$ is motor score excluding the i^{th} component,

$\{missing_{-i}\}$ denotes missing-value dummy variables excluding the i^{th} component and

$$\{missing_{-i}\} = \begin{cases} 0, if\ there\ is\ no\ missing\ value\ for\ all\ other\ variables \\ 1, if\ there\ is\ at\ least\ one\ missing\ varibles\ besides\ i^{th}\ varible\ ' \end{cases}$$

$motor\ score_i$ is the i^{th} component (keeping 1.0 as the default), and

$missing_i$ indicates whether the i^{th} component is missing.

Thus, for our purpose, to estimate total cost of the GG measurements

$$logcost$$

$$= \begin{cases} f_1(GG\ var_1 + \cdots + var_n *) + f_2(age) + \gamma\{tier\}, & if\ no\ missing \quad (2.1) \\ f_1(GG\ var_1 + \cdots + var_{i-1} + 1 + var_{i+1} + \cdots + var_n) + f_2(age) + \gamma\{tier\}, if\ i\ is\ missing & (2.2) \end{cases}$$

$$logcost_i = \begin{cases} f_1\{GG\ var_1 + \ldots + var_{i-1} + var_{i+1} + \cdots + var_n\} + f_2(age) + \gamma_1\{tier\} & \\ +\gamma_2\{missing_{-i}\} + \beta_1 \times \mathbf{var_i} \sout{+ \beta_2 \times \mathbf{0}}, & if\ i\ is\ not\ missing \quad (2.3) \\ f_1\{GG\ var_1 + \ldots + var_{i-1} + var_{i+1} + \cdots + var_n\} + f_2(age) + \gamma_1\{tier\} & \\ +\gamma_2\{missing_{-i}\} + \beta_1 \times \mathbf{1} + \beta_2 \times \mathbf{1}, & if\ i\ is\ missing \quad (2.4) \end{cases}$$

in which

$\{GG\ var_1 + \ldots + var_{i-1} + var_{i+1} \ldots + var_n\}$ is the sum of each GG Selfcare and Mobility variables excluding the i^{th} component, and

var_i is the i^{th} component (keeping 1.0 as the default)

* CMS Final Rule 2019: we are finalizing based on public comments the use of an unweighted motor score to assign patients to CMGs beginning with FY 2020.

First, as every variable in Eq. 2.1 is known, we could get the estimated function f_1, f_2 and γ;

Second, for $i = 1,2, \dots, n$, we could estimate the i^{th} function γ_1, γ_2 and coefficient $\widehat{\beta_1}$ using f_1, f_2, f_3 when i is not missing;

Third, we could estimate coefficient $\widehat{\beta_2}$ by using the known functions and parameters when i is missing;

Finally, we could see that

$$logcost_i = \begin{cases} f_1\{GG\ var_1 + \dots + var_{i-1} + var_{i+1} + \cdots + var_n\} + f_2(age) + \gamma_1\{tier\} \\ \quad + \gamma_2\{missing_{-i}\} + \beta_1 \times \boldsymbol{var_i}, \quad if\ i\ is\ not\ missing \quad (2.3) \\ f_1\{GG\ var_1 + \dots + var_{i-1} + var_{i+1} + \cdots + var_n\} + f_2(age) + \gamma_1\{tier\} \\ \quad + \gamma_2\{missing_{-i}\} + \beta_1(\mathbf{1} + \frac{\boldsymbol{\beta_2}}{\boldsymbol{\beta_1}}), \quad if\ i\ is\ missing \quad (2.5) \end{cases}$$

Thus, $\widehat{var_i} = 1 + \frac{\beta_2}{\beta_1}$

R codes:

```
UDSPRO = read_sav("X:/Research/Yolanda/Data/UDSPRO V30Q1_V30Q4_2020.sav")

###################################
### choose CMS's 18 vars, filter impairment groups & medicare & missing cost
UDSPRO$dummycase = data.frame(1:nrow(UDSPRO))
UDSPRO$IMPGRPS = as.numeric(UDSPRO$IMPGRPS)
UDSPRO$AV14_ADM_H0350_BLADDER_NUM = as.numeric(UDSPRO$AV14_ADM_H0350_BLADDER)
UDSPRO$AV14_ADM_H0400_BOWEL_NUM = as.numeric(UDSPRO$AV14_ADM_H0400_BOWEL)
UDSPRO = UDSPRO %>% mutate(BLADDER = recode(AV14_ADM_H0350_BLADDER_NUM,`0`=6,`1`=5,`2`=3,`3`=2,`4`=1,`5`=0,`9`=0)) %>%
 mutate(BOWEL = recode(AV14_ADM_H0400_BOWEL_NUM,`0`=6,`1`=3,`2`=2,`3`=1,`9`=0))

DATAFT = UDSPRO %>%
 select(AV14_GG0130A_SC_EATING:AV14_GG0170K_MBL_WALK150FT,
      AV14_GG0170M_MBL_1STAIR, AV14_ADM_H0350_BLADDER, AV14_ADM_H0400_BOWEL,
      AV14_GG0130A_SC_EATING_NUM:AV14_GG0170K_MBL_WALK150FT_NUM,
```

```
    AV14_GG0170M_MBL_1STAIR_NUM, BLADDER, BOWEL, #AV14_ADM_H0350_BLADDER_NUM, AV14_ADM_H0400_BOWEL_NUM,
    dummycase, ADMITAGE, TIER, TOTALADJUSTEDFPP, IMPGRPS, PRIMPAY, SECPAY) %>%
 select(-AV14_GG0170A_MBL_ROLL, -AV14_GG0170G_MBL_CARTX, -AV14_GG0170H_MBL_PATIENTWALK,
    -AV14_GG0170A_MBL_ROLL_NUM, -AV14_GG0170G_MBL_CARTX_NUM)

IMP_select = data.frame(xtabs(~IMPGRPS, data = DATAFT))
IMP_select = IMP_select %>% filter(Freq > nrow(DATAFT) * .003) %>% select(IMPGRPS)
IMP_select = as.numeric(IMP_select[,1])

DATAFT = DATAFT %>%
 #filter(IMPGRPS %in% IMP_select) %>%
 filter(PRIMPAY %in% c("51","02")|SECPAY %in% c("51","02")) %>%
 filter(!is.na(TOTALADJUSTEDFPP))

DATA2 = DATAFT %>% select(-PRIMPAY, -SECPAY) %>% filter(IMPGRPS %in% IMP_select)

###################################
### GAM ##################
### modify GG Vars
n = 18
DATA2[,1:n] = as.data.frame(apply(DATA2[,1:n], 2, as.numeric))
DATA2 = mutate_at(DATA2, vars(1:n),list(~replace(., . %in% c(0,1,2,3,4,5,6), 0)))
DATA2[,1:n][DATA2[,1:n] != 0] = 1
DATA2[,1:n][is.na(DATA2[,1:n])] = 1
#temp = filter(DATA2, AV14_GG0170M_MBL_1STAIR %in% c(1,0))
DATA2[,1:n] = as.data.frame(apply(DATA2[,1:n], 2, as.factor))

DATA2[,(n+1):(n*2)] = as.data.frame(apply(DATA2[,(n+1):(n*2)], 2, as.numeric))
DATA2[,(n+1):(n*2)][is.na(DATA2[,(n+1):(n*2)])] = 1
DATA2$motorscore = rowSums(DATA2[,(n+1):(n*2)])
#names(DATA2)
#temp = filter(DATA2, is.na(motorscore))

DATA2$TIER = as.factor(DATA2$TIER)
DATA2$LogCost = log(DATA2$TOTALADJUSTEDFPP)

### fit model: logcost = f1(motor) + f2(age) + ??{tier}
gam1 = gam(LogCost ~ bs(motorscore,7) + bs(ADMITAGE,7) + TIER ,data = DATA2)
#summary(gam1)
#gam2 = gam(LogCost ~ ns(motorscore,4) + AGEGRPS + TIER ,data = DATA2)
#summary(gam2)
#gam3 = gam(LogCost ~ ns(motorscore,4) + TIER ,data = DATA2)
#summary(gam3)
#anova(gam3, gam1, gam2, test="F")
#plot(gam1, se=TRUE, col="green ")
#library(ggplot2)
#qplot(x = ADMITAGE, y = LogCost, data = DATA2)
```

```
### for i-th var ###
result = data.frame(x = 1, y = 1, z = 1, k = 1)
k=0
#summary(DATA2$IMPGRPS)

for (j in IMP_select) {
 DATAIMP = filter(DATA2, IMPGRPS == j)
 k = k+1
 temp1 = select(DATAIMP, 1:n, dummycase)
 temp2 = DATAIMP[(n+1):(n*2)]

 for (i in 1:n) {

  ### fit model: logcost = f1(motor-i) + f2(age) + ??1{tier} + ??2{missing-i} + ??1*motori + ??2*0
  ### logcost - f1(motor-i) - f2(age) - intercept = ??1{tier} + ??2{missing-i} + ??1*motori
  Gamma2 = temp1 %>% select(-i) %>% filter_all(all_vars(. != "1")) %>% select(dummycase) %>% mutate(MV=0)
  DATAIMP$motors_i = rowSums(temp2[,-i])
  DATA3 = DATAIMP %>% select(i, i+n, motors_i, dummycase, ADMITAGE, TIER, LogCost) %>%
   filter(DATAIMP[,all_of(i)] =='0') %>%
   left_join(Gamma2, by = "dummycase")
  names(DATA3)[2] = "VAR_NUM"
  DATA3$MV = as.factor(ifelse(is.na(DATA3$MV), 1, 0))

  #gam1$coefficients[1]
  #gam1$terms
  PCT_Motor = quantile(DATAIMP$motorscore, c(.2, .4, .6, .8))
  PCT_Age = quantile(DATAIMP$ADMITAGE, c(.2, .4, .6, .8))

  DATA3 =  mutate(DATA3, Y = LogCost - gam1$coefficients[1]
           - (gam1$coefficients[2]*motors_i + gam1$coefficients[3] *motors_i^2 + gam1$coefficients[4]*motors_i^3
             + gam1$coefficients[5]*(motors_i - PCT_Motor[1])^3 + gam1$coefficients[6]*(motors_i - PCT_Motor[2])^3
             + gam1$coefficients[7]*(motors_i - PCT_Motor[3])^3 + gam1$coefficients[8]*(motors_i - PCT_Motor[4])^3)
           - (gam1$coefficients[9]*ADMITAGE + gam1$coefficients[10] *ADMITAGE^2 + gam1$coefficients[11]*ADMITAGE^3
             + gam1$coefficients[12]*(ADMITAGE - PCT_Age[1])^3 + gam1$coefficients[13]*(ADMITAGE - PCT_Age[2])^3
             + gam1$coefficients[14]*(ADMITAGE - PCT_Age[3])^3 + gam1$coefficients[15]*(ADMITAGE - PCT_Age[4])^3))

  #TEMP = filter(DATA3, is.na(Y))
  gam2 = gam(Y ~ TIER + MV + VAR_NUM, data = DATA3)

  #summary(gam2)
  #gam2$coefficients

  #preds=predict(gam1, newdata = DATA3)
  #summary(preds)

  ### fit model: logcost = f1(motor-i) + f2(age) + ??1{tier} + ??2{missing-i} + ??1*motori + ??2*missingi
```

```
  ### logcost - (f1(motor-i) + f2(age) + ??1{tier} + ??2{missing-i} + ??1) = ??2
  DATA4 = DATAIMP %>% select(i, i+n, motors_i, dummycase, ADMITAGE, TIER, LogCost, IMPGRPS) %>%
   filter(DATAIMP[,i] =='1') %>%
   left_join(Gamma2, by = "dummycase")
  DATA4$MV = ifelse(is.na(DATA4$MV), 1, 0)
  DATA4$TIER1 = ifelse(DATA4$TIER == "1", 1,0)
  DATA4$TIER2 = ifelse(DATA4$TIER == "2", 1,0)
  DATA4$TIER3 = ifelse(DATA4$TIER == "3", 1,0)

  DATA4 =  mutate(DATA4, Y = LogCost - gam1$coefficients[1] - gam2$coefficients[1]
             - (gam1$coefficients[2]*motors_i + gam1$coefficients[3] *motors_i^2 + gam1$coefficients[4]*motors_i^3
               + gam1$coefficients[5]*(motors_i - PCT_Motor[1])^3 + gam1$coefficients[6]*(motors_i - PCT_Motor[2])^3
               + gam1$coefficients[7]*(motors_i - PCT_Motor[3])^3 + gam1$coefficients[8]*(motors_i - PCT_Motor[4])^3)
             - (gam1$coefficients[9]*ADMITAGE + gam1$coefficients[10] *ADMITAGE^2 + gam1$coefficients[11]*ADMITAGE^3
               + gam1$coefficients[12]*(ADMITAGE - PCT_Age[1])^3 + gam1$coefficients[13]*(ADMITAGE - PCT_Age[2])^3
               + gam1$coefficients[14]*(ADMITAGE - PCT_Age[3])^3 + gam1$coefficients[15]*(ADMITAGE - PCT_Age[4])^3)
             - (gam2$coefficients[2]*TIER1 + gam2$coefficients[3]*TIER2 + gam2$coefficients[4]*TIER3)
             - gam2$coefficients[5]*MV - gam2$coefficients[6])

  #hist(DATA4$Y)
  #boxplot(DATA4$Y)

  result[i+n*(k-1),] = data.frame(x = 1 + mean(DATA4$Y) /gam2$coefficients[6], y = names(DATA4)[1], z = mean(DATA4$IMPGRPS), k = nrow(DATAIMP))
 }}

result[,1][result[,1] > 6] = 6
result[,1][result[,1] < 1] = 1
names(result) = c("GG", "GG Items", "IMPGRP", "Counts")

result2 = result %>% group_by(`GG Items`) %>% summarise_at(vars(`GG`), median)

###################################
#### PLOT
DATAPLOT = read.csv("X:/Research/Yolanda/ACRM Conference/RAND_PLOT.csv")

DATAPLOT = filter(DATAPLOT, GG.Items %in% c('Eating', 'Toileting',
                            'BedChairTransfer', 'ToiletTransfer'))
names(DATAPLOT) = c('GG.Items','2017','2018','2019','2020','Total')
temp = gather(DATAPLOT, FiscalYear, GGScore, '2017':Total, factor_key=TRUE)

ggplot(temp, aes(x = GG.Items, y = GGScore, group = FiscalYear, color = FiscalYear)) +
 geom_col(aes(fill = FiscalYear), position = "dodge", linetype=0) +
 scale_fill_manual(values = c('steelblue2','dodgerblue1','skyblue1','deepskyblue1','dodgerblue3')) +
 theme(legend.position="bottom") +  #ylim(0, 6)
 scale_y_continuous(breaks=seq(0, 6, 1), limits = c(0, 6))
```

Appendix – Rand Tech Report part 3.2 (2005)

3.2. Models For Estimating Missing-Value Effects

The goal in this section is to develop equations that describe the relationship between costs and FIM™ scores. Right-hand-side variables to explain cost will include the standard measures: age, motor score, cognitive score, and comorbidities. In addition, we include dummy variables here to measure the missing-value effects. Our goal is to measure the (percentage) effect on cost of knowing that an item was actually missing versus scored as "maximally dependent."

In developing the FRGs, we had fit generalized additive models (GAM models) to the data. GAM is an exploratory tool that is particularly good at detecting nonlinear patterns in the data. GAM approximates a regression relationship as a sum of smooth (rather than linear) functions of the independent variables. As parameterized here, GAM divides the range of each variable into five equally spaced intervals and fits flexible curvers (cubic splines) within each interval. The method is described in Hastie et al., *The Elements of Statistical Learning,* New York, N.Y.: Springer-Verlag, 2001.

We continue to use GAM here to explore the relationships between candidate explanatory variables and cost. The basic form of our GAM model is

$$\text{logcost} = f_1(\text{motor}) + f_2(\text{cog}) + f_3(\text{age}) + \gamma\{\text{tier}\} \qquad (3.1)$$

where f_1, f_2, and f_3 are cubic splines fit by GAM and {tier} indicates comorbidity conditions. Here, as elsewhere in this document, we implement the tier effect as three dummy variables, tier-1 through tier-3, going from high to low levels of severity. The lists of tier diagnoses used here were developed from preliminary analysis of the same 2002 data. They are found in Carter and Totten (forthcoming) (although the definition of *ventilator condition* used there was not used here).

Because GAM approximates the relationship as a nonlinear function, a change in motor score from 20 to 21 might decrease predicted cost by a different percentage than might a change from 70 to 71. Because the relationship is assumed additive in the log scale, the decrease in predicted cost due to a change in motor score from 20 to 21 will be the same regardless of the values of the other independent variables. The flexibility that GAM affords can make its predictions more accurate than those of the linear model when the relationship between the dependent and independent variables is nonlinear.

We wish to examine the effect of the standard explainers of cost: motor score, cognitive score, and age. In addition, we want to control for all elements that might be missing, so that we can see whether the assignment of 1.0 introduced a bias. For each variable with percentage "missing" 1.0 percent or more (after rounding), we fit GAM models similar to what had been developed before, but with additional dummy variables to capture the effect of whether or not a given variable is missing:

$$\text{logcost} = f_1(\text{motor}_{-i}) + f_2(\text{cog}) + f_3(\text{age}) + \gamma_1\{\text{tier}\} + \gamma_2\{\text{missing}_{-i}\} + \beta_1^* \text{motor}_i + \beta_2^* \text{missing}_i \qquad (3.2)$$

So, for example, to investigate the effect of "missing" transfer to toilet, we ran a transfer-to-toilet model in which that variable was eliminated from the motor-score index, the standard variables were included, all "missing" indicator variables were included, and transfer to toilet was entered as a linear term. In Eq. (3.2), f_1, f_2, and f_3 are GAM fits, {tier} denotes the three levels of comorbidity dummy variables, motor_{-i} is motor score excluding the i^{th} component, $\{\text{missing}_{-i}\}$ denotes missing-value dummy variables excluding the i^{th} component, motor_i is the i^{th} component (keeping 1.0 as the default), and missing indicates whether the i^{th} component is missing. In each case, β_1 and β_2 contain the information about the effect of missing values. Because we kept the default assignment of 1.0 for motor_i, β_2 is the effect of being missing versus having a 1.0, $\beta_1 + \beta_2$ is the effect of being missing versus having a 2.0, etc.

These methods will tell us how much of a percentage shift in costs is explained by coding an activity as "missing" rather than "maximally dependent"; they also tell what score could be assigned to the missing activity other than the current default (1.0) to achieve the same effect as using the dummy variables.