

Supplement. Risk-Standardized Mortality Modeling

Variation in Diagnostic Coding of Patients with Pneumonia and its Association with Hospital Risk-Standardized Mortality Rates

1. Inclusion/Exclusion Criteria

- a. Discharge from hospital participating in Perspective Database July 1, 2007 – June 30, 2010
- b. Age ≥ 18
- c. Principal Diagnosis of Pneumonia
- d. Principal Diagnosis of Sepsis with secondary diagnosis of pneumonia
- e. Principal Diagnosis of Respiratory Failure with secondary diagnosis of pneumonia
- f. Antibiotics and chest x-ray or CT scan within first 2 days
- g. Exclusions
 - i. Transfer from acute care hospital
 - ii. Transfer to acute care hospital:
 - iii. Length of stay < 2 days
 - iv. Hospitals with < 100 eligible patients

2. Risk-standardized Mortality Modeling

- a. Outcome: Patient discharge status 1=Expired, 0=discharged alive
- b. Patient-level covariates

Candidate Variables	
Age	Age in years
Sex	M, F
Comorbidities: derived from AHRQ Elixhauser algorithm	1. Congestive Heart Failure 2. Valvular disease 3. Pulmonary Circulation disorders 4. Peripheral vascular disease 5. Hypertension 6. Paralysis 7. Other neurological disorders 8. Chronic pulmonary disease 9. Diabetes without chronic complications 10. Diabetes with chronic complications 11. Hypothyroidism 12. Renal failure 13. Liver disease 14. Chronic Peptic ulcer disease 15. HIV and AIDS 16. Lymphoma 17. Metastatic cancer 18. Solid tumor without metastasis 19. Rheumatoid arthritis/collagen vascular diseases 20. Coagulation deficiency 21. Obesity 22. Weight loss 23. Fluid and electrolyte disorders 24. Blood loss anemia 25. Deficiency anemias 26. Alcohol abuse 27. Drug abuse 28. Psychoses 29. Depression

Modeling Mortality: 2 separate models,

1. Include patients with Principal sepsis or respiratory failure with secondary pneumonia as well as Principal pneumonia
2. Restricted to patients with Principal diagnosis of pneumonia

Modeling Strategy:

1. Include all candidate variables in initial models
2. Remove all variables with $p > 0.25$
3. Remove variables 1 at a time ; retain those with $p < 0.05$

Hierarchical Generalized Linear Model (HGLM) including a hospital-specific random intercept

$$\text{Model: } h(Y_{ij}) = \alpha_i + \beta Z_{ij}$$
$$\alpha_i = \mu + \omega_i, \omega_i \sim N(0, \tau^2)$$

where

Y_{ij} is the outcome (in-hospital mortality) at the i^{th} hospital for the j^{th} patient

Z_{ij} is set of patient-specific covariates identified in GLM modeling

$h()$ is the logit link

α_i is the hospital specific-intercept

ω_i is the random deviation of the i^{th} hospital

μ is the adjusted mean outcome over hospitals $\hat{\mu} = \sum_{i=1}^{n_h} \alpha_i$

τ^2 is the between hospital variance

Model accounts for hospital level clustering, and separates between and within hospital variation in mortality -- estimates hospital-specific intercept.

Computation of risk-standardized mortality rate (RSMR) for each model

(1. All PN cases 2. Principal PN cases)

- ii. Compute **Expected** mortality rate **for each hospital** based upon patient mix and average hospital intercept $\hat{\mu}$ -- that is, use HGLM estimates of μ and β , with observed patient characteristics Z_{ij} :

$$\hat{e}_i(Z) = \sum_{j=1}^{n_i} \hat{e}_{ij}(Z) = \sum_{j=1}^{n_i} h^{-1}(\hat{\mu} + \hat{\beta}Z_{ij})$$

- iii. Compute **Predicted** mortality rate **for each hospital** based upon patient mix and hospital – specific intercept α_i – that is, use HGLM estimates of hospital-specific rates:

$$\hat{y}_i(Z) = \sum_{j=1}^{n_i} \hat{y}_{ij}(Z) = \sum_{j=1}^{n_i} h^{-1}(\hat{\alpha}_i + \hat{\beta}Z_{ij})$$

iv. Compute overall unadjusted mortality rate across all patients:

$$\bar{y} = \frac{1}{n_{total}} \sum_{i=1}^h \sum_{j=1}^{n_i} y_{ij}$$

v. Compute standardized mortality rate for each hospital as ratio of predicted to expected rate times the overall unadjusted mortality rate:

$$RSMR_i = \hat{s}_i(\mathbf{Z}) = \left(\frac{\hat{y}_i(\mathbf{Z})}{\hat{e}_i(\mathbf{Z})} \right) (\bar{y})$$

vi. Use bootstrapping to derive probability estimate and CI estimates on RSMR for each hospital.

A. Start with file containing data for all hospitals, all cases, and repeat b=1 to 500 times:

1. Sampling **with replacement**, take sample of size $n = 329$ hospitals
2. For each selected sample fit 2 HGLM models (using covariates retained in final models from initial data):
 - i. Using all cases in selected hospitals
 - ii. Using Principal PN cases in selected hospitals

When a hospital is selected more than once, treat each repetition as a distinct hospital.

Output from each model:

- a) regression coefficients for risk factors, $\hat{\beta}^{(b)}$
 - b) hospital-specific random effects, $\hat{a}_i^{(b)}$ and standard errors, $\widehat{se}(\hat{a}_i^{(b)})$
 - c) mean hospital effect, $\hat{\mu}^{(b)}$ and associated variance, τ^2
3. Generate a random effect for each hospital (include replicates for hospitals selected more than once), assuming a Normal distribution for the random effects, as $\hat{a}_i^{(b^*)} \sim N(\hat{a}_i^{(b)}, \widehat{var}(\hat{a}_i^{(b)}))$.
 4. For each hospital sampled, and each eligible case j for the appropriate model (1. all cases, 2. Principal PN cases), compute predicted mortality $\hat{y}_{ij}^{(b)}$ and expected mortality $\hat{e}_{ij}^{(b)}$ and $\hat{s}_i(\mathbf{Z})$ for each case, using overall mortality $\hat{y}_{ij}^{(b)}$, $\hat{\mu}^{(b)}$ and $\hat{\beta}^{(b)}$ from step 2, and $\hat{a}_i^{(b^*)}$ from step 3.

- B.** For each model (1. all cases, 2. Principal PN cases) compute probability estimate for each hospital as the mean of the bootstrapped estimates, and the 95% confidence interval estimate for each hospital as the (2.5th, 97.5th) percentiles of the distribution of estimates.
- C.** For each model, identify hospitals as at average if the 95% CI includes the overall mean mortality, better than average if the upper limit is less than the mean mortality, and worse than average if the lower limit is greater than the mean mortality.

SAS Code for Bootstrap

```

%macro hboot;

** select sample with replacement **;
** repeat B times to get desired total B replicates **;

%do i=1 %to &B; /* B replicate samples */

*create seed dataset*;
data s_seed;
  _seed_=&sseed+&i;
run;

PROC SURVEYSELECT
  data = &indsn /* input file */
  out=bootsample&i
  seed = s_seed /* starting seed dataset */
  method = urs /* sample with replacement */
  sampsize = &nhspp /* number of hospitals */
  ;
  cluster prov_id; /* keep all pats within hospitals */
run;
proc sort data=bootsample&i;
  by prov_id pat_key;
run;

** add new replicate ID variable (sampid) so that hospitals with **;
** repeats in a sample will have distinct IDs for glimmix **;
data boot&i(drop=j);
  set bootsample&i;
** adds rows per patient for repeat selection **;
  do j=1 to numberhits;
    sampid=(prov_id*100)+j; ** unique sample id for each hit **;
    replicate=&i; ** assign counter for replicate **;
    output; ** output obs with new ID for each hit **;
  end;
run;

/*RUN GLIMMIX AND OUTPUT PARAMETER ESTIMATES AND RANDOM INTERCEPTS */
/* first for all cases then repeat for principle dx PN only */

ods listing close;
** first for all cases **;
PROC GLIMMIX data = boot&i method = quad(qpoints=5) ;
  /* use model derived for full data */

```

```

class sampid male CHF_rev VALVE_rev PULMCIRC_rev PERIVASC_rev HTN_C_rev
  PARA_rev cpulm_rev DIABETES_rev HYPOTHY_rev LIVER_rev AIDS_rev
  LYMPH_rev METS_rev TUMOR_rev ARTH_rev OBESE_rev WGTLOSS_rev
  ANEMDEF_rev DRUG_rev PSYCH_rev DEPRESS_rev renlfail_rev;
model mort_stat(Ref = '0:Alive') = age male CHF_rev VALVE_rev
  PULMCIRC_rev PERIVASC_rev HTN_C_rev PARA_rev cpulm_rev
  DIABETES_rev HYPOTHY_rev LIVER_rev AIDS_rev LYMPH_rev
  METS_rev TUMOR_rev ARTH_rev OBESE_rev WGTLOSS_rev ANEMDEF_rev
  DRUG_rev PSYCH_rev DEPRESS_rev renlfail_rev
  / dist = binary link = logit s;
RANDOM intercept / subject = sampid s;
ods output
ParameterEstimates = ParameterEst&i(keep = effect estimate stderr)
SolutionR = RandomInt&i;
run;

** 2nd for pdx PN cases only **;
data pnboot&i;
  set boot&i;
  if pdx_v1=1;
run;
PROC GLIMMIX data = pnboot&i method = quad(qpoints=5) ;
  /* use model derived for PN cases */
  class sampid male CHF_rev VALVE_rev PULMCIRC_rev HTN_C_rev PARA_rev
  cpulm_rev DIABETES_rev HYPOTHY_rev LIVER_rev LYMPH_rev METS_rev
  TUMOR_rev OBESE_rev WGTLOSS_rev ANEMDEF_rev DRUG_rev PSYCH_rev
  DEPRESS_rev renlfail_rev;
  model mort_stat(Ref = '0:Alive') = age male CHF_rev VALVE_rev
  PULMCIRC_rev HTN_C_rev PARA_rev cpulm_rev DIABETES_rev
  HYPOTHY_rev LIVER_rev LYMPH_rev METS_rev TUMOR_rev OBESE_rev
  WGTLOSS_rev ANEMDEF_rev DRUG_rev PSYCH_rev DEPRESS_rev
  renlfail_rev
  / dist = binary link = logit s;
  RANDOM intercept / subject = sampid s;
  ods output
  ParameterEstimates = pnParameterEst&i(keep = effect estimate stderr )
  SolutionR = pnRandomInt&i;
run;
ods listing;

/*Generate ALPHAS FOR EACH HOSPITAL */
data ranint&i(keep = replicate subject estimate StdErrPred gen_est);
  set randomint&i;
  ** generate RE **;
  aseed=&aseed+&i;
  gen_est=estimate+(stderrpred*rannor(aseed));
  replicate=&i; ** assign unique replicate ID **;
run;
data pnrantint&i(keep = replicate subject estimate StdErrPred gen_est);
  set pnrandomint&i;
  ** generate RE **;
  aseed2=&aseed2+&i;
  gen_est=estimate+(stderrpred*rannor(aseed2));
  replicate=&i; ** assign unique replicate ID **;
run;

/*OUTPUT ONLY OVERALL INTERCEPT */
data intercept&i(keep = replicate int_est );

```

```

        set parameterest&i;
        rename estimate = int_est;
        replicate = &i;
        if effect = 'Intercept' then output;
run;

data pntercept&i(keep = replicate int_est );
    set pnparameterest&i;
    rename estimate = int_est;
    replicate = &i;
    if effect = 'Intercept' then output;
run;
/* clean up PARAMETER ESTIMATES DATA*/
/* delete rows for reference levels */
data parameterest&i;
    set parameterest&i;
    replicate = &i;
    if stderr = . then delete;
run;
** repeat for PN **;
data PNparameterest&i;
    set PNparameterest&i;
    replicate = &i;
    if stderr = . then delete;
run;

/* MERGE random effects with intcpt to compute hosp random intcpt (alphas)*/
data alpha&i;
    merge ranint&i intercept&i;
    by replicate;
    ** compute hospital random intercept **;
    alpha = int_est+gen_est;

    ** retrieve hospital ID from sample id **;
    sampid = 1*(substr(subject,8,6));
    prov_id = int(sampid/100);
run;

    ** repeat for pn cases **;
data pnalpha&i;
    merge pnrant&i pntercept&i;
    by replicate;
    ** compute hospital random intercept **;
    alpha = int_est+gen_est;

    ** retrieve hospital ID from sample id **;
    sampid = 1*(substr(subject,8,6));
    prov_id = int(sampid/100);
run;

/*TRANSPOSE model betas INTO ONE ROW */
proc transpose data = parameterest&i out = parmsout&i(drop = _NAME_)
    prefix = beta_;
    by replicate;
    id effect;
    var estimate;
run;

```

```

proc transpose data = pnparameterest&i out = pnparmsout&i(drop = _NAME_)
  prefix = beta_;
  by replicate;
  id effect;
  var estimate;
run;

/*PUT TOGETHER hosp random effects (alphas) & PARAMETER ESTIMATES (betas) */
data allparms&i;
  merge alpha&i parmsout&i;
  by replicate;
run;
** repeat for pn cases **;
data PNparms&i;
  merge pnalphabet&i pnparmsout&i;
  by replicate;
run;

/*SORT AND MERGE ALL FILES*/
proc sort data = allparms&i; by sampid; run;
proc sort data = pnparms&i; by sampid; run;
proc sort data = boot&i; by sampid; run;
proc sort data = pnboot&i; by sampid; run;

** compute RSMR for each set of parameter estimates **;
** concatenate into one file **;
data pred_exp&i;
  merge boot&i allparms&i(in=in_i);
  by sampid;
  if in_i; /* if hosp in ith replicate */
  ** compute pred and exp for each subject **;
pred_lp = alpha + age*beta_age + female*beta_male + CHF*beta_chf_rev +
  VALVE*beta_valve_rev + PULMCIRC*beta_pulmcirc_rev +
  PERIVASC*beta_perivasc_rev + HTN_C*beta_htn_c_rev +
  PARA*beta_para_rev + cpulm*beta_cpulm_rev +
  DIABETES*beta_diabetes_rev + HYPOTHY*beta_hypothy_rev +
  LIVER*beta_liver_rev + AIDS*beta_aids_rev + LYMPH*beta_lymph_rev +
  METS*beta_mets_rev + TUMOR*beta_tumor_rev + ARTH*beta_arth_rev +
  OBESE*beta_obese_rev + WGHTLOSS*beta_wghtloss_rev +
  ANEMDEF*beta_anemdef_rev + DRUG*beta_drug_rev +
  PSYCH*beta_psych_rev + DEPRESS*beta_depress_rev +
  renlfail*beta_renlfail_rev;

exp_lp = beta_intercept + age*beta_age + female*beta_male + CHF*beta_chf_rev
+ VALVE*beta_valve_rev + PULMCIRC*beta_pulmcirc_rev +
PERIVASC*beta_perivasc_rev + HTN_C*beta_htn_c_rev +
PARA*beta_para_rev + cpulm*beta_cpulm_rev +
DIABETES*beta_diabetes_rev + HYPOTHY*beta_hypothy_rev +
LIVER*beta_liver_rev + AIDS*beta_aids_rev + LYMPH*beta_lymph_rev +
METS*beta_mets_rev + TUMOR*beta_tumor_rev + ARTH*beta_arth_rev +
OBESE*beta_obese_rev + WGHTLOSS*beta_wghtloss_rev +
ANEMDEF*beta_anemdef_rev + DRUG*beta_drug_rev + PSYCH*beta_psych_rev
+ DEPRESS*beta_depress_rev + renlfail*beta_renlfail_rev;

pred_pp = exp(pred_lp)/(1 + exp(pred_lp));
exp_pp = exp(exp_lp)/(1 + exp(exp_lp));

```

```

** sum subj within hospital - compute RSMR and output one obs per hosp**;
RETAIN NPATALL N_PRED N_EXP 0;

IF FIRST.sampid THEN DO;
  NPATALL = 0;
  N_PRED = 0;
  N_EXP = 0;
END;
NPATALL+1;
N_PRED+PRED_PP;
N_EXP+EXP_PP;
IF LAST.sampid THEN do;
  RSMR_ALL = (N_PRED/N_EXP)* &allmort;
  OUTPUT;
END;
RUN;

/* APPEND RESULTS INTO ONE FILE */
proc append base=hsp_rsmr data=pred_exp&i force;
run;

** repeat for pn cases only;
** merge with param estimates and compute pred/exp prob **;
data PNpred_exp&i;
  merge pnboot&i pnparms&i(in=in_i);
  by sampid;
  if in_i;

pred_lp = alpha + age*beta_age + female*beta_male + CHF*beta_chf_rev +
  VALVE*beta_valve_rev + PULMCIRC*beta_pulmcirc_rev +
  HTN_C*beta_htn_c_rev + PARA*beta_para_rev + cpulm*beta_cpulm_rev +
  DIABETES*beta_diabetes_rev + HYPOTHY*beta_hypothy_rev +
  LIVER*beta_liver_rev + LYMPH*beta_lymph_rev + METS*beta_mets_rev +
  TUMOR*beta_tumor_rev + OBESE*beta_obese_rev +
  WGHTLOSS*beta_wghtloss_rev + ANEMDEF*beta_anemdef_rev +
  DRUG*beta_drug_rev + PSYCH*beta_psych_rev +
  DEPRESS*beta_depress_rev + renlfail*beta_renlfail_rev;

exp_lp = beta_intercept + age*beta_age + female*beta_male + CHF*beta_chf_rev
+ VALVE*beta_valve_rev + PULMCIRC*beta_pulmcirc_rev +
HTN_C*beta_htn_c_rev + PARA*beta_para_rev + cpulm*beta_cpulm_rev +
DIABETES*beta_diabetes_rev + HYPOTHY*beta_hypothy_rev +
LIVER*beta_liver_rev + LYMPH*beta_lymph_rev + METS*beta_mets_rev +
TUMOR*beta_tumor_rev + OBESE*beta_obese_rev +
WGHTLOSS*beta_wghtloss_rev + ANEMDEF*beta_anemdef_rev +
DRUG*beta_drug_rev + PSYCH*beta_psych_rev + DEPRESS*beta_depress_rev
+ renlfail*beta_renlfail_rev;

pred_pp = exp(pred_lp)/(1 + exp(pred_lp));
exp_pp = exp(exp_lp)/(1 + exp(exp_lp));

** sum subj within hospital compute RSMR and output one obs per hosp**;
RETAIN NPATALL N_PRED N_EXP 0;

IF FIRST.sampid THEN DO;
  NPATALL = 0;
  N_PRED = 0;

```



```

    N_EXP = 0;
END;
    NPATALL+1;
    N_PRED+PRED_PP;
    N_EXP+EXP_PP;
IF LAST.sampid THEN do;
RSMR_PN = (N_PRED/N_EXP)* &pnmort;
OUTPUT;
END;

RUN;

/* append PN results into single file */
proc append base=pn_rsmr data=PNpred_exp&i force;
run;

%end; ** end loop through replicate samples **;

/*COMPUTE 95% BOOTSTRAP CI PER HOSPITAL - USE ORIGINAL HOSPITAL ID*/
PROC SORT DATA = HSP_RSMR; BY PROV_ID; RUN;
PROC UNIVARIATE NOPRINT DATA = HSP_RSMR;
    BY PROV_ID;
    VAR RSMR_ALL;
    OUTPUT PCTLPRE = P_ PCTLPTS = 2.5 97.5 mean = mean
        out = rsmr_ci;
RUN;

PROC MEANS DATA = HSP_RSMR n min mean std max clm;
VAR RSMR_ALL;
RUN;

/*DEFINE HOSPITAL PERFORMANCE AS PER CMS DEFINITION*/
DATA RSMR_CI_PERFORM;
    SET WORK.RSMR_CI;
    IF P_2_5 GT &ALLMORT THEN HSP_PERFORMANCE = 2; /* WORSE */
ELSE IF P_97_5 LT &ALLMORT THEN HSP_PERFORMANCE = 0; /* BETTER */
    ELSE HSP_PERFORMANCE = 1; /* AT AVERAGE */

LENGTH HSP_PERFORMANCE 3;
FORMAT HSP_PERFORMANCE HSP_PERFORM.;
RUN;

PROC FREQ DATA = RSMR_CI_PERFORM;
TABLE HSP_PERFORMANCE;
RUN;

/*COMPUTE 95% BOOTSTRAP CI PER HOSPITAL FOR PDX PN*/
PROC SORT DATA = PN_RSMR;
BY PROV_ID;
RUN;

PROC UNIVARIATE NOPRINT DATA = PN_RSMR;
BY PROV_ID;
VAR RSMR_pn;
OUTPUT PCTLPRE = pnP_ PCTLPTS = 2.5 97.5 mean = mean out = PNrsmr_ci;
RUN;

PROC MEANS DATA = PN_RSMR n min mean std max clm;
VAR RSMR_PN;

```

```

RUN;

/*DEFINE HOSPITAL PERFORMANCE AS PER CMS DEFINITION*/
DATA PNRSMR_CI_PERFORM;SET PNRSMR_CI;
    IF pnP_2_5 GT &PNMORT THEN PNHSP_PERFORMANCE = 2; /* WORSE */
ELSE IF pnP_97_5 LT &PNMORT THEN PNHSP_PERFORMANCE = 0; /* BETTER */
    ELSE PNHSP_PERFORMANCE = 1; /* AT AVERAGE */

LENGTH PNHSP_PERFORMANCE 3;
FORMAT PNHSP_PERFORMANCE HSP_PERFORM.;
RUN;

PROC FREQ DATA = PNRSMR_CI_PERFORM;
TABLE PNHSP_PERFORMANCE;
RUN;

** merge all and pdx PN results **;
data &outdsn;
    merge RSMR_CI_PERFORM
          PNRSMR_CI_PERFORM;
    by prov_id;
run;

proc freq data=&outdsn;
    table hsp_performance * pnhsp_performance ;
run;

%mend;

/* DEFINE FORMAT FOR PERFORMANCE RATING */
PROC FORMAT;
VALUE HSP PERFORM
    0 = '0: BETTER THAN'
    1 = '1: NO DIFFERENT'
    2 = '2: WORSE THAN';
run;

/*FIND MEAN MORTALITY AT PATIENT LEVEL */
proc means data = hsp_pn mean;
var mort_stat;
run;
proc means data = hsp_pn mean;
    where pdx_v1=1; /* only pdx PN */
var mort_stat;
run;

*****;
** define macro variables **;
** indsn: input patient level dataset **;
** sseed: seed for surveysample selection **;
** nhsp: # of hospitals = # selected in each replicate sample **;
** B: # rounds of repeat sampling **;
** aseed: seed for generating alphas, all cases **;
** aseed2: seed for generating alphas, pdx PN only **;
** allmort: mean mortality for all cases **;
** pnmort: mean mortality for pdx PN cases **;
** outdsn: output data with performance ratings **;
*****;

```

```
%let indsn=hsp_pn;  
%let sseed=5836;  
%let nhsp=329;  
%let B=500;  
%let aseed=7543;  
%let aseed2=8791;  
%let allmort= 0.07215;  
%let pnmort= 0.03547;  
%let outdsn=boot.bothRSMR;  
  
** run macro **;  
%hboot;
```