Supplement: R Code

S1. R Code for Hosmer-Lemeshow goodness of fit test (n=49)

```
require(ggplot2)
require(binom)
require(pROC)
require(ResourceSelection)
require(grid)
replace.unknown.with.na <- function(raw.values) {
  as.numeric(ifelse(raw.values == 'Unknown', NA,
raw.values))
}
read.and.prepare.ncr.data <- function(filename) {
  data <- read.csv(filename, as.is=TRUE)
  colnames(data) <- c('integer.score',
'predicted.mortality.rate', 'died')
  data <- within(data, {
    integer.score <-
replace.unknown.with.na(integer.score)
    predicted.mortality.rate <-
replace.unknown.with.na(predicted.mortality.rate)
    died <- (died == 'Yes')
  })
  # We'll exclude the rows with missing predictions
  return(subset(data, !
is.na(predicted.mortality.rate)))
}
construct.demirjian.data <- function() {
  # Reproduced data from (Demirjian 2011)
  count.data <- data.frame(
    risk.score=seq(6, 44, by=2),
    num.subjects=c(1, 1, 1, 5, 17, 43, 60, 104, 127,
137, 181, 151, 122, 80, 49, 21, 14, 6, 1, 1),
    num.died=c(0, 0, 0, 0, 2, 3, 8, 9, 47, 52, 105,
103, 96, 69, 48, 17, 12, 6, 1, 1)
  )
  per.subject.data <- with(
    count.data,
    data.frame(
      integer.score=c(
        rep(risk.score, num.died),
        rep(risk.score, num.subjects - num.died)
), died=c(

rep( c(1, 0),

          times=c(sum(num.died), sum(num.subjects -
num.died)) )

) )

)
```

```
  return(per.subject.data)
}
plot.hosmer.lemeshow.results <- function(data,
hl.results) {
  plot.data <- data.frame(
    predicted.mortality.rate=prop.table(hl.results
$expected, 1)[,2],
    observed.mortality.rate=prop.table(hl.results
$observed, 1)[,2]
  )
  confint.df <- binom.confint(
    hl.results$observed[,2],
    margin.table(hl.results$observed, 1),
    methods='wilson'
)

  plot.data$observed.rate.upper <- confint.df
$upper.Freq
  plot.data$observed.rate.lower <- confint.df
$lower.Freq
  return(
    ggplot()
    + geom_pointrange(
        aes(
          x=predicted.mortality.rate,
          y=observed.mortality.rate,
          ymin=observed.rate.lower,
          ymax=observed.rate.upper
        ),
        plot.data,
        size=0.4
      )
    + geom_dotplot(
        aes(x=predicted.mortality.rate,
fill=mortality.status),
        within(data, mortality.status <-
factor(ifelse(died, 'Died', 'Survived'))),
        y=-0.165,
        stackgroups=TRUE,
        binwidth=0.025,
        dotsize=0.9,
        method='histodot'
      )
    + geom_hline(yintercept=0)
    + geom_abline(intercept=0, slope=1,
linetype='dashed', size=0.4)
    + coord_cartesian(xlim=c(0, 1), ylim=c(-0.17, 1))
    + labs(x='Predicted mortality rate', y='Observed
mortality rate', fill=NULL)
    + scale_fill_grey(start=0, end=1)
    + theme(legend.key.size=unit(0.7, 'lines'))
) }

check.varying.subgroup.sizes <- function(data,
```

```
subgroup.sizes=3:10) {
  p.values <- sapply(
    subgroup.sizes,
    function(num.subgroups)
    hoslem.test(data$died, data
$predicted.mortality.rate, num.subgroups)$p.value
  )
  names(p.values) <- subgroup.sizes
  return(p.values)
}
compute.rocs.with.confidence.intervals <-
function(ncr.data, demirjian.data, roc.ci.step=0.01) {
  ncr.roc.object <- with(ncr.data, roc(died,
predicted.mortality.rate, ci=TRUE))
  demirjian.roc.object <- with(demirjian.data,
roc(died, integer.score))
  roc.ci.specificities <- seq(0, 1, roc.ci.step)
  ncr.ci.object <- ci.se(ncr.roc.object,
specificities=roc.ci.specificities)
  demirjian.ci.object <- ci.se(demirjian.roc.object,
specificities=roc.ci.specificities)
  return(list(
    ncr.roc=ncr.roc.object,
    ncr.ci=ncr.ci.object,
    demirjian.roc=demirjian.roc.object,
    demirjian.ci=demirjian.ci.object
)) }

plot.roc.comparison <- function(roc.list) {
  make.roc.plot.data <- function(roc.object) {
    return(data.frame(
      specificity=roc.object$specificities,
      sensitivity=roc.object$sensitivities
)) }

  make.roc.ci.plot.data <- function(roc.ci.object) {
    return(data.frame(
      specificity=as.numeric(rownames(roc.ci.object)),
      sensitivity.lower=roc.ci.object[,'2.5%'],
      sensitivity.upper=roc.ci.object[,'97.5%']
    ))
}

  combine.cohorts <- function(ncr.data, demirjian.data)
{
    ncr.data$cohort <- rep('NCR cohort',
nrow(ncr.data))
    demirjian.data$cohort <- rep('Model cohort',
nrow(demirjian.data))
    combined.data <- rbind(ncr.data, demirjian.data)
    combined.data$cohort <-
relevel(factor(combined.data$cohort), 'NCR cohort')
    return(combined.data)
  }
```

```
  roc.plot.data <- combine.cohorts(
    make.roc.plot.data(roc.list$ncr.roc),
    make.roc.plot.data(roc.list$demirjian.roc)
  )
  roc.ci.plot.data <- combine.cohorts(
    make.roc.ci.plot.data(roc.list$ncr.ci),
    make.roc.ci.plot.data(roc.list$demirjian.ci)
  )
  (ggplot()
   # draw ribbons below grid lines...
   + geom_ribbon(
       aes(specificity, ymin=sensitivity.lower,
ymax=sensitivity.upper, fill=cohort),
       roc.ci.plot.data
     )
   # ...manually draw grid lines...
   + geom_hline(yintercept=seq(0, 1, 0.25), size=0.2,
color='gray60', linetype='dotted')
   + geom_vline(xintercept=seq(0, 1, 0.25), size=0.2,
color='gray60', linetype='dotted')
   + geom_abline(intercept=1, slope=1, size=0.25,
col='gray50')
   # ...and draw paths on top
   + geom_path(aes(specificity, sensitivity,
linetype=cohort), roc.plot.data, size=0.4)
+ xlim(1, 0)

   + scale_linetype_manual(values=c('NCR cohort'=1,
'Model cohort'=2))
   + scale_fill_manual(values=c('NCR cohort'='gray85',
'Model cohort'='gray70'))
   + labs(x='Specificity', y='Sensitivity')
   + theme(
       legend.key.size=unit(1, 'lines'),
       panel.grid=element_blank()
     )
) }

theme_set(
  theme_bw() + theme(
    text=element_text(size=8.75),
    legend.title=element_blank(),
    legend.margin=unit(-1, 'lines'),
    legend.key=element_blank(),
    legend.position='bottom',
    plot.margin=unit(c(0.5, 1, 0, 0), 'lines')
) )

data <-
read.and.prepare.ncr.data('InHospMortalityAKIIntegerSco
re.csv')
demirjian.data <- construct.demirjian.data()
hl.results <- hoslem.test(data$died, data
$predicted.mortality.rate, 5)
```

```
plot.hosmer.lemeshow.results(data, hl.results)
ggsave('Fig2 Hosmer Lemeshow.eps', width=2.9,
height=3.3)
hl.results
check.varying.subgroup.sizes(data)
roc.list <-
compute.rocs.with.confidence.intervals(data,
demirjian.data)
plot.roc.comparison(roc.list)
ggsave('Fig3 ROC comparisons.eps', width=2.9,
height=3.1)
roc.list$ncr.roc
roc.test(roc.list$ncr.roc, roc.list$demirjian.roc)
```

S2. R Code for Sensitivity Analysis

```
p.values.if.n.patients.died <- function(data,
num.to.change) {
  rows.to.change <- do.call(expand.grid,
rep(list(1:nrow(data)), num.to.change))
  rows.to.change$p.value <-
sapply(1:nrow(rows.to.change), function(index) {
    indices.to.change <-
as.numeric(rows.to.change[index,])
    new.died <- data$died
    new.died[indices.to.change] <- TRUE
    return(hoslem.test(new.died, data
$predicted.mortality.rate, 5)$p.value)
  })
  return(rows.to.change)
}
p.values.if.more.deaths.included <- function(data,
num.to.add) {
  predicted.mortality.rate.to.add <-
do.call(expand.grid, rep(list((0:100) / 100),
num.to.add))
  predicted.mortality.rate.to.add$p.value <- sapply(
    1:nrow(predicted.mortality.rate.to.add),
    function(index) {
      new.predicted.mortality.rates <-
as.numeric(predicted.mortality.rate.to.add[index,])
      data.to.add <- data.frame(
        integer.score=rep(NA, num.to.add),
predicted.mortality.rate=new.predicted.mortality.rates,
        died=TRUE
      )
      new.data <- rbind(data, data.to.add)
      return(hoslem.test(new.data$died, new.data
$predicted.mortality.rate, 5)$p.value)
    }
  )
  return(predicted.mortality.rate.to.add)
}

max(p.values.if.n.patients.died(data, 1)$p.value)
max(p.values.if.n.patients.died(data, 2)$p.value)
max(p.values.if.more.deaths.included(data, 1)$p.value)
max(p.values.if.more.deaths.included(data, 2)$p.value)
```