30-day mortality

University of Waterloo Health Data Science Lab

March 2018

Data preparation

```
set.seed(4375) # "HDSL"

admissions_df <- read.csv("../../../MIMIC/sentiment_data/patient_df.csv")
sapsii_df <- read_csv("../../../MIMIC/sentiment_data/sapsii_df.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
notes_df <- read_csv("../../../MIMIC/sentiment_data/notes_df_sntmnt.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   row_id = col_double(),
##   subject_id = col_double(),
##   hadm_id = col_double(),
##   chartdate = col_datetime(format = ""),
##   charttime = col_datetime(format = ""),
##   storetime = col_datetime(format = ""),
##   category = col_character(),
##   description = col_character(),
##   cgid = col_double(),
##   iserror = col_logical(),
##   text = col_character(),
##   polarity = col_double(),
##   subjectivity = col_double()
## )
```

```
## Warning: 244 parsing failures.
##    row     col           expected actual                                                file
##  59631 iserror 1/0/T/F/TRUE/FALSE    1.0 '../../../MIMIC/sentiment_data/notes_df_sntmnt.csv'
## 210653 iserror 1/0/T/F/TRUE/FALSE    1.0 '../../../MIMIC/sentiment_data/notes_df_sntmnt.csv'
## 271374 iserror 1/0/T/F/TRUE/FALSE    1.0 '../../../MIMIC/sentiment_data/notes_df_sntmnt.csv'
## 271376 iserror 1/0/T/F/TRUE/FALSE    1.0 '../../../MIMIC/sentiment_data/notes_df_sntmnt.csv'
## 271377 iserror 1/0/T/F/TRUE/FALSE    1.0 '../../../MIMIC/sentiment_data/notes_df_sntmnt.csv'
## ...... ....... .................. ...... ............................................
## See problems(...) for more details.
```

```
mimicii_patients_df <- read.csv("../../../MIMIC/sentiment_data/mimic2_patients_df.csv")
mimicii_admissions_df <- read.csv("../../../MIMIC/sentiment_data/mimic2_admissions_df.csv")
```

```
notes_df$text <- NULL

notes_df <- notes_df %>% left_join(admissions_df, by="hadm_id")

# Get the difference between the time of death and the time that the note was taken.
# We don't want to look at notes close to death (~12hrs)

deathtime.POSIX <- as.POSIXlt(strptime(notes_df$dod, "%Y-%m-%d %H:%M:%S"))
charttime.POSIX <- as.POSIXlt(strptime(notes_df$charttime, "%Y-%m-%d %H:%M:%S"))
notes_df$death_chart_diffhrs <- difftime(deathtime.POSIX,
                                         charttime.POSIX,
                                         units="hours")

setDT(notes_df)

notes_df <- notes_df[is.na(death_chart_diffhrs) |
                       death_chart_diffhrs > 12]

notes_df <- notes_df[is.na(iserror)]

analysis_df <- admissions_df

setDT(analysis_df)

analysis_df[dod != "",
            mortality_days := difftime(as.Date(dod),
                                       as.Date(admittime), units="days")]

# Remove patients with data entry error: death time recorded before admission.
analysis_df <- analysis_df[is.na(mortality_days) | mortality_days >= 0]

analysis_df[!is.na(mortality_days) & mortality_days < 30, mortality_30d := 1]
analysis_df[is.na(mortality_30d), mortality_30d := 0]
analysis_df$mortality_30d <- factor(analysis_df$mortality_30d)
```

```
setDT(analysis_df)
analysis_df$first_careunit <- factor(analysis_df$first_careunit)

### For counting number of notes per patient

mytbl <- notes_df
nrow(mytbl)
```

```
## [1] 471207
```

```
length(unique(mytbl$hadm_id))
```

```
## [1] 27477
```

```
count.tbl <- count(mytbl, hadm_id)
count.tbl
```

```
## # A tibble: 27,477 x 2
##    hadm_id     n
##      <dbl> <int>
##  1  100003    12
```

```
##  2  100006    14
##  3  100007     7
##  4  100017     5
##  5  100018    25
##  6  100021    10
##  7  100030     7
##  8  100031    14
##  9  100033     4
## 10  100036     5
## # ... with 27,467 more rows
```

```
analysis_df_counted <- left_join(analysis_df, count.tbl, by="hadm_id")
setDT(analysis_df_counted)
analysis_df_counted[is.na(n), n := 0]


mean(analysis_df_counted$n)
```

```
## [1] 12.35661
```

```
sd(analysis_df_counted$n)
```

```
## [1] 25.32777
```

```
median(analysis_df_counted$n)
```

```
## [1] 5
```

```
summary(analysis_df_counted$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    5.00   12.36   12.00  913.00
```

```
a_df_counted_nonzeronotes <- analysis_df_counted[n > 0]
nrow(a_df_counted_nonzeronotes)
```

```
## [1] 27477
```

```
mean(a_df_counted_nonzeronotes$n)
```

```
## [1] 17.14914
```

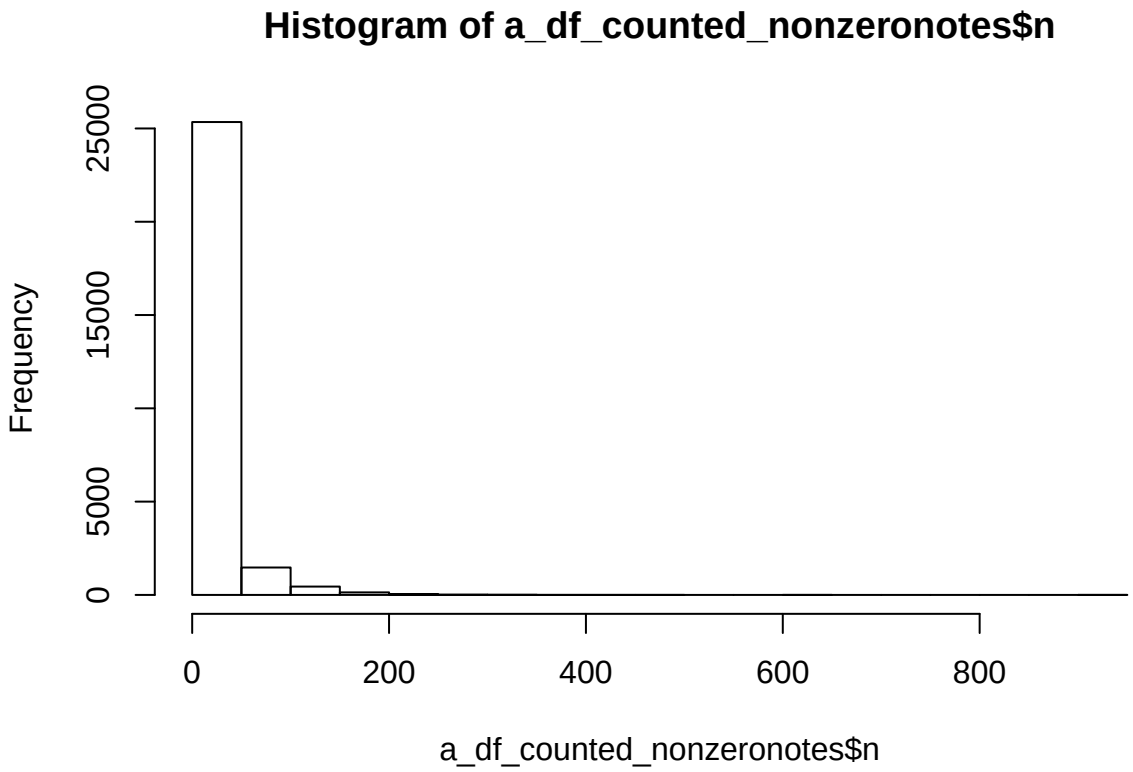
```
sd(a_df_counted_nonzeronotes$n)
```

```
## [1] 28.42744
```

```
summary(a_df_counted_nonzeronotes$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    4.00    8.00   17.15   17.00  913.00
```

```
hist(a_df_counted_nonzeronotes$n)
```




```
# Break down by 30-day mortality group
# Survived
a_df_counted_nonzeronotes_surv <- analysis_df_counted[n > 0 & mortality_30d == 0]
nrow(a_df_counted_nonzeronotes_surv)
```

```
## [1] 24448
```

```
mean(a_df_counted_nonzeronotes_surv$n)
```

```
## [1] 16.68967
```

```
sd(a_df_counted_nonzeronotes_surv$n)
```

```
## [1] 28.92469
```

```
summary(a_df_counted_nonzeronotes_surv$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    4.00    7.00   16.69   16.00  913.00
```

```
# Expired
a_df_counted_nonzeronotes_exp <- analysis_df_counted[n > 0 & mortality_30d == 1]
nrow(a_df_counted_nonzeronotes_exp)
```

```
## [1] 3029
```

```
mean(a_df_counted_nonzeronotes_exp$n)
```

```
## [1] 20.85771
```

```
sd(a_df_counted_nonzeronotes_exp$n)
```

```
## [1] 23.72081
```

```r
summary(a_df_counted_nonzeronotes_exp$n)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    5.00   12.00   20.86   28.00  250.00
```

```r
# Perform unequal variance, unpaired, two-sided t-test for sapsii
# between 30-day mortality groups
t.test(a_df_counted_nonzeronotes_exp$n, a_df_counted_nonzeronotes_surv$n,
       alternative = "two.sided", paired = FALSE, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  a_df_counted_nonzeronotes_exp$n and a_df_counted_nonzeronotes_surv$n
## t = 8.8866, df = 4228.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  3.248506 5.087576
## sample estimates:
## mean of x mean of y
##  20.85771  16.68967
```

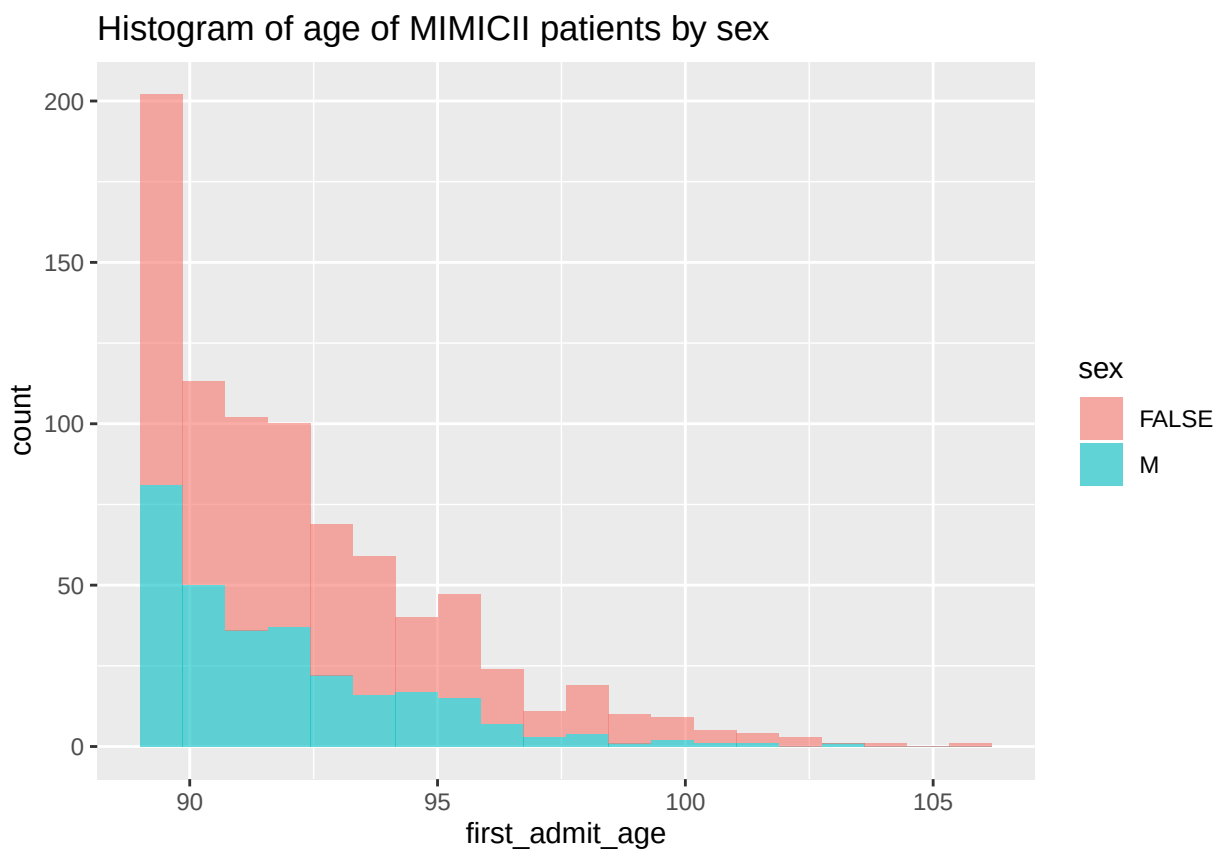
```r
### End block

notes_df <- notes_df %>%
  group_by(hadm_id) %>%
  #summarize(mean_polarity = mean(polarity),
  #          mean_subjectivity = mean(subjectivity))
  mutate(mean_polarity = mean(polarity),
            mean_subjectivity = mean(subjectivity))
# joining notes by admission id.
analysis_df <- analysis_df %>% left_join(notes_df, by="hadm_id", copy = TRUE)

# Do we have to rename here?
analysis_df <- analysis_df %>%
  rename(icustay_id = icustay_id.x)

# joining SAPS II and SOFA scores by ICU stay ID.
analysis_df <- analysis_df %>% left_join(sapsii_df, by="icustay_id", copy=TRUE)


### new add ( for the next chunk)
analysis_df <- analysis_df %>%
  rename(gender = gender.x, first_admit_age = first_admit_age.x, first_careunit = first_careunit.y)
```

Notice there are some data where patients are recorded to be very old (~300 yrs old). This was done by MIMIC to anonymize patients over the age of 89.

However, in MIMIC-II, the ages of patients were recorded after death. Therefore, we will randomly sample ages from the >89 group from MIMIC-II and assign their ages to the >89 group from MIMIC-III. We will do so by gender/sex since as we will see, their age distributions are slightly different.

```r
# rename mimicii columns
mimicii_admissions_df <- mimicii_admissions_df %>%
  rename(subject_id = SUBJECT_ID, hadm_id = HADM_ID,
         admit_dt = ADMIT_DT, disch_dt = DISCH_DT)
```

```
mimicii_patients_df <- mimicii_patients_df %>%
  rename(subject_id = SUBJECT_ID, sex = SEX,
         dob = DOB, dod = DOD,
         hospital_expire_flag = HOSPITAL_EXPIRE_FLG)
mimicii_age_dataset <- mimicii_admissions_df %>%
  group_by(subject_id) %>%
  arrange(admit_dt) %>%
  filter(row_number()==1)

mimicii_age_dataset <- mimicii_age_dataset %>%
  left_join(mimicii_patients_df, by="subject_id")

mimicii_age_dataset$first_admit_age <- as.numeric(as.Date(mimicii_age_dataset$admit_dt) -
                                        as.Date(mimicii_age_dataset$dob)) / 365.242

setDT(mimicii_age_dataset)
setDT(analysis_df)

ggplot(mimicii_age_dataset[first_admit_age > 89 & first_admit_age < 120]) +
  geom_histogram(aes(first_admit_age, fill=sex), bins=20, alpha=0.6) +
  ggtitle("Histogram of age of MIMICII patients by sex")
```




```
analysis_df[gender == "M" & first_admit_age > 89]$first_admit_age <-
  sample(mimicii_age_dataset[sex == "M" & first_admit_age > 89 & first_admit_age < 120]$first_admit_age
         size = nrow(analysis_df[gender == "M" & first_admit_age > 89]),
         replace = TRUE)
```

```
analysis_df[gender == "M" & first_admit_age > 89]$first_admit_age <-
  sample(mimicii_age_dataset[sex == "M" & first_admit_age > 89 & first_admit_age < 120]$first_admit_age
         size = nrow(analysis_df[gender == "M" & first_admit_age > 89]),
         replace = TRUE)


analysis_df_w_pol <- analysis_df %>%
  filter(!is.na(mean_polarity))
```

Code for producting summary statistics in table 2

```
adf.survived <- analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 0,]
adf.expired <- analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 1,]

sum(adf.survived$first_careunit == "CCU") / nrow(adf.survived)
```

```
## [1] 0.1406983
```

```
sum(adf.expired$first_careunit == "CCU") / nrow(adf.expired)
```

```
## [1] 0.1561145
```

```
sum(adf.survived$first_careunit == "CSRU") / nrow(adf.survived)
```

```
## [1] 0.1903615
```

```
sum(adf.expired$first_careunit == "CSRU") / nrow(adf.expired)
```

```
## [1] 0.07852417
```

```
sum(adf.survived$first_careunit == "MICU") / nrow(adf.survived)
```

```
## [1] 0.3465612
```

```
sum(adf.expired$first_careunit == "MICU") / nrow(adf.expired)
```

```
## [1] 0.5091329
```

```
sum(adf.survived$first_careunit == "SICU") / nrow(adf.survived)
```

```
## [1] 0.1632163
```

```
sum(adf.expired$first_careunit == "SICU") / nrow(adf.expired)
```

```
## [1] 0.1541518
```

```
sum(adf.survived$first_careunit == "TSICU") / nrow(adf.survived)
```

```
## [1] 0.1591627
```

```
sum(adf.expired$first_careunit == "TSICU") / nrow(adf.expired)
```

```
## [1] 0.1020767
```

```
# Perform asymptotic chi-square test of independence between the
# variables "first_careunit" and "mortality_30d". We use asymptotic test because
# this problem is too large for Fisher's exact test.
freq.table <- table(analysis_df_w_pol$mortality_30d, analysis_df_w_pol$first_careunit.x)
freq.table
```

```
##
##        CCU   CSRU   MICU   SICU  TSICU
```

```
##   0  57409  77673 141407  66597  64943
##   1   9863   4961  32166   9739   6449
```

```
chisq.test(freq.table)
```

```
##
##  Pearson's Chi-squared test
##
## data:  freq.table
## X-squared = 9122.6, df = 4, p-value < 2.2e-16
```

```
mean(adf.survived$sapsii)
```

```
## [1] 37.7858
```

```
sd(adf.survived$sapsii)
```

```
## [1] 14.23368
```

```
mean(adf.expired$sapsii)
```

```
## [1] 47.53128
```

```
sd(adf.expired$sapsii)
```

```
## [1] 14.29524
```

```
# Perform unequal variance, unpaired, two-sided t-test for sapsii
# between 30-day mortality groups
t.test(adf.survived$sapsii, adf.expired$sapsii,
       alternative = "two.sided", paired = FALSE, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  adf.survived$sapsii and adf.expired$sapsii
## t = -159.55, df = 83756, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -9.865201 -9.625758
## sample estimates:
## mean of x mean of y
##  37.78580  47.53128
```

```
mean(adf.survived$first_admit_age)
```

```
## [1] 66.80608
```

```
sd(adf.survived$first_admit_age)
```

```
## [1] 36.927
```

```
mean(adf.expired$first_admit_age)
```

```
## [1] 79.52286
```

```
sd(adf.expired$first_admit_age)
```

```
## [1] 48.79807
```

```
# Perform unequal variance, unpaired, two-sided t-test for first_admit_age
# between 30-day mortality groups
```

```r
t.test(adf.survived$first_admit_age, adf.expired$first_admit_age,
       alternative = "two.sided", paired = FALSE, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  adf.survived$first_admit_age and adf.expired$first_admit_age
## t = -62.778, df = 74786, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13.11380 -12.31974
## sample estimates:
## mean of x mean of y
##  66.80608  79.52286
```

```r
sum(adf.survived$gender == "F") / nrow(adf.survived)
```

```
## [1] 0.4269917
```

```r
sum(adf.expired$gender == "F") / nrow(adf.expired)
```

```
## [1] 0.453259
```

```r
# Perform asymptotic chi-square test of independence
# between the variables "gender" and "mortality_30d".
# We use asymptotic test to remain consistent with the previous
# test for first_careunit.
freq.table <- table(analysis_df_w_pol$mortality_30d, analysis_df_w_pol$gender)
freq.table
```

```
##
##          F      M
##   0 174225 233804
##   1  28636  34542
```

```r
chisq.test(freq.table, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  freq.table
## X-squared = 153.96, df = 1, p-value < 2.2e-16
```

```r
mean(adf.survived$mean_polarity)
```

```
## [1] 0.05339996
```

```r
mean(adf.expired$mean_polarity)
```

```
## [1] 0.03504258
```

```r
# Perform unequal variance, unpaired, two-sided t-test for first_admit_age
# between 30-day mortality groups
t.test(adf.expired$mean_polarity, adf.survived$mean_polarity,
       alternative = "two.sided", paired = FALSE, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  adf.expired$mean_polarity and adf.survived$mean_polarity
## t = -118.59, df = 91006, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01866078 -0.01805397
## sample estimates:
##  mean of x  mean of y
## 0.03504258 0.05339996
```

```
mean(adf.survived$mean_subjectivity)
```

```
## [1] 0.3670352
```

```
mean(adf.expired$mean_subjectivity)
```

```
## [1] 0.3686932
```

```
# Perform unequal variance, unpaired, two-sided t-test for first_admit_age
# between 30-day mortality groups
t.test(adf.expired$mean_subjectivity, adf.survived$mean_subjectivity,
       alternative = "two.sided", paired = FALSE, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  adf.expired$mean_subjectivity and adf.survived$mean_subjectivity
## t = 10.061, df = 87669, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.001334975 0.001980925
## sample estimates:
## mean of x mean of y
## 0.3686932 0.3670352
```

```
# Calculate Spearman rank rho correlation between mean_polarity and sapsii
spearman.test(analysis_df_w_pol$mean_polarity, analysis_df_w_pol$sapsii)
```

```
##      Rsquare            F          df1          df2       pvalue
## 6.835003e-02 3.456972e+04 1.000000e+00 4.712050e+05 0.000000e+00
##            n
## 4.712070e+05
```

```
spearman.test(analysis_df_w_pol$mean_subjectivity, analysis_df_w_pol$sapsii)
```

```
##      Rsquare            F          df1          df2       pvalue
## 1.485681e-04 7.001645e+01 1.000000e+00 4.712050e+05 0.000000e+00
##            n
## 4.712070e+05
```

Fit logistic regression model and plot results

```
analysis_df_w_pol$mean_polarity <- 10 * analysis_df_w_pol$mean_polarity
analysis_df_w_pol$mean_subjectivity <- 10 * analysis_df_w_pol$mean_subjectivity

gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
```

```
}

model_mortality_30d = glm(mortality_30d ~ mean_polarity +
                            + mean_subjectivity + sapsii +
                            first_careunit.x + gender,
                          family=binomial(link='logit'),
                          data=analysis_df_w_pol)

summary(model_mortality_30d)
```

```
##
## Call:
## glm(formula = mortality_30d ~ mean_polarity + +mean_subjectivity +
##     sapsii + first_careunit.x + gender, family = binomial(link = "logit"),
##     data = analysis_df_w_pol)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0338  -0.5746  -0.4217  -0.2871   3.1050
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -3.6412238  0.0478050 -76.168  < 2e-16 ***
## mean_polarity         -0.9617665  0.0132214 -72.743  < 2e-16 ***
## mean_subjectivity      0.2056180  0.0119135  17.259  < 2e-16 ***
## sapsii                 0.0368385  0.0003002 122.718  < 2e-16 ***
## first_careunit.xCSRU  -0.9256076  0.0187302 -49.418  < 2e-16 ***
## first_careunit.xMICU   0.1647768  0.0130717  12.606  < 2e-16 ***
## first_careunit.xSICU  -0.0656062  0.0159582  -4.111 3.94e-05 ***
## first_careunit.xTSICU -0.3936485  0.0175668 -22.409  < 2e-16 ***
## genderM               -0.0269666  0.0090025  -2.995  0.00274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 371371  on 471206  degrees of freedom
## Residual deviance: 335375  on 471198  degrees of freedom
## AIC: 335393
##
## Number of Fisher Scoring iterations: 5
```

```
model.odds.ratio <- odds.ratio(model_mortality_30d)
```

```
## Waiting for profiling to be done...
```

```
model.odds.ratio
```

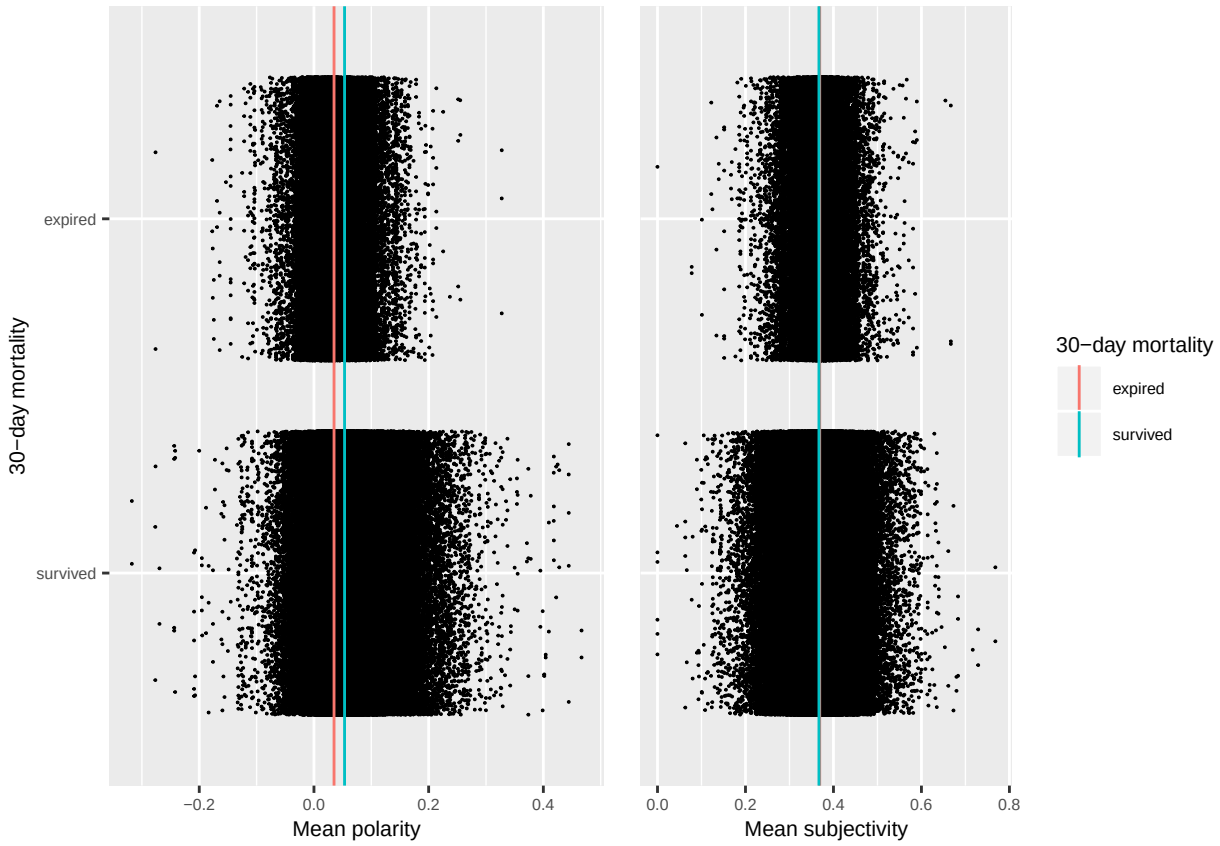
```
##                            OR    2.5 % 97.5 %         p
## (Intercept)           0.026220 0.023873 0.0288 < 2.2e-16 ***
## mean_polarity         0.382217 0.372431 0.3922 < 2.2e-16 ***
## mean_subjectivity     1.228284 1.199947 1.2573 < 2.2e-16 ***
## sapsii                1.037525 1.036915 1.0381 < 2.2e-16 ***
## first_careunit.xCSRU  0.396291 0.381984 0.4111 < 2.2e-16 ***
## first_careunit.xMICU  1.179130 1.149343 1.2098 < 2.2e-16 ***
```

```
## first_careunit.xSICU  0.936500 0.907662 0.9663 3.938e-05 ***
## first_careunit.xTSICU 0.674591 0.651742 0.6982 < 2.2e-16 ***
## genderM               0.973394 0.956372 0.9907   0.00274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrm_30d <- lrm(model_mortality_30d$formula, data = analysis_df_w_pol)

# Calculate Nagelkerke pseudo-R^2
lrm_30d$stats["R2"]
```

```
##        R2
## 0.1348736
```

```
output.folder = "../../../MIMIC/"

analysis_df_w_pol$mean_polarity <- 0.1 * analysis_df_w_pol$mean_polarity
analysis_df_w_pol$mean_subjectivity <- 0.1 * analysis_df_w_pol$mean_subjectivity

pol.vs.mort.plot <- ggplot(analysis_df_w_pol, aes(mean_polarity, mortality_30d))
setDF(analysis_df_w_pol)
mean_mean_polarity_df <-
  data.frame(mortality_30d = c("expired", "survived"),
             mean_mean_polarity =
               c(mean(analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 1, "mean_polarity"]),
                 mean(analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 0, "mean_polarity"])))

jitter.plot.mp <- pol.vs.mort.plot + geom_jitter(size = 0.005, alpha=1) +
  xlab("Mean polarity") +
  scale_y_discrete("30-day mortality", labels = c("0" = "survived", "1" = "expired")) +
  geom_vline(aes(xintercept = mean_mean_polarity, colour = mortality_30d), data=mean_mean_polarity_df) +
  scale_colour_discrete(name = "30-day mortality") + theme(legend.position = "none", text=element_text(s

sub.vs.mort.plot <- ggplot(analysis_df_w_pol, aes(mean_subjectivity, mortality_30d))
mean_mean_subjectivity_df <-
  data.frame(mortality_30d = c("expired", "survived"),
             mean_mean_subjectivity =
               c(mean(analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 1, "mean_subjectivity"]),
                 mean(analysis_df_w_pol[analysis_df_w_pol$mortality_30d == 0, "mean_subjectivity"])))

jitter.plot.ms <- sub.vs.mort.plot + geom_jitter(size = 0.005, alpha = 1) +
  xlab("Mean subjectivity") +
  scale_y_discrete("30-day mortality", labels = c("0" = "survived", "1" = "expired")) +
  geom_vline(aes(xintercept = mean_mean_subjectivity, colour = mortality_30d), data=mean_mean_subjectivi
  scale_colour_discrete(name = "30-day mortality") +
  theme(text=element_text(size=8), axis.title.y = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank())

jitter.plot <- grid.arrange(jitter.plot.mp, jitter.plot.ms, ncol = 2)
```

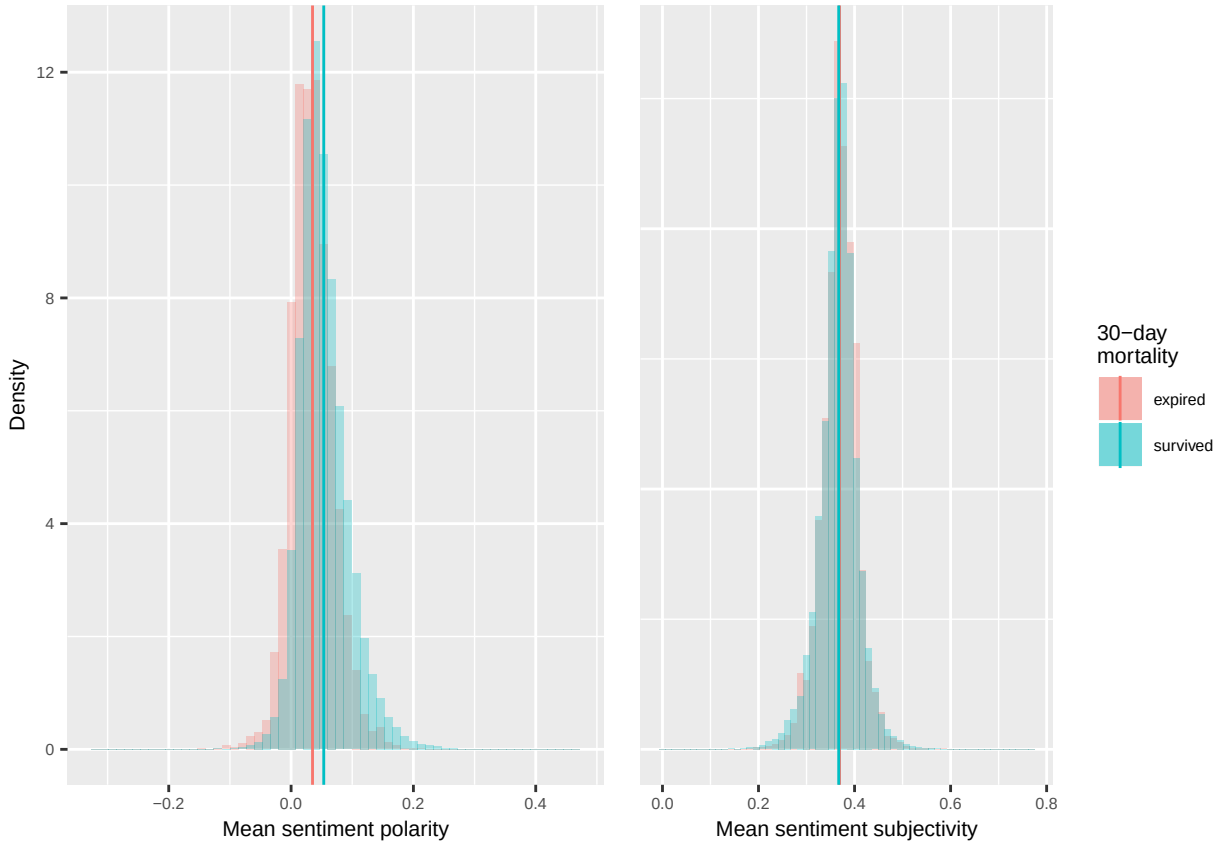



```
jitter.plot
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

```
ggsave(file.path(output.folder, "jitter_plot.tiff"),
       plot = jitter.plot, width=150, height=84, units="mm", dpi=300)

# Mixture histograms

setDT(analysis_df_w_pol)
levels(analysis_df_w_pol$mortality_30d) <- c("survived", "expired")
pol.mixture.histogram <- ggplot() +
  geom_histogram(aes(y = ..density.., x = mean_polarity, fill=mortality_30d),
                 alpha = 0.3, bins = 60, data=analysis_df_w_pol[mortality_30d == "expired"]) +
  geom_histogram(aes(y = ..density.. , x = mean_polarity, fill=mortality_30d),
                 alpha = 0.3, bins = 60, data=analysis_df_w_pol[mortality_30d == "survived"]) +
  ylab("Density") + xlab("Mean sentiment polarity") +
  geom_vline(aes(xintercept = mean_mean_polarity, colour = mortality_30d), data=mean_mean_polarity_df) +
  scale_fill_manual(values = gg_color_hue(2),
                    name="30-day\nmortality") +
  scale_colour_manual(values = gg_color_hue(2), name="30-day\nmortality") +
  theme(text = element_text(size = 8), legend.position = "none")

sub.mixture.histogram <- ggplot() +
  geom_histogram(aes(y = ..density.., x = mean_subjectivity, fill=mortality_30d),
```

```
                alpha = 0.3, bins = 60, data=analysis_df_w_pol[mortality_30d == "expired"]) +
  geom_histogram(aes(y = ..density.. , x = mean_subjectivity, fill=mortality_30d),
                alpha = 0.3, bins = 60, data=analysis_df_w_pol[mortality_30d == "survived"]) +
  ylab("Density") + xlab("Mean sentiment subjectivity") +
  geom_vline(aes(xintercept = mean_mean_subjectivity, colour = mortality_30d), data=mean_mean_subjectivi
  scale_fill_manual(values = gg_color_hue(2),
                    name="30-day\nmortality") +
  scale_colour_manual(values = gg_color_hue(2), name="30-day\nmortality") +
  theme(text=element_text(size=8), axis.title.y = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank())

mixture.histogram <- grid.arrange(pol.mixture.histogram, sub.mixture.histogram, ncol = 2)
```




```
mixture.histogram
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

```
ggsave(file.path(output.folder,"mixture_histogram.tiff"), plot = mixture.histogram,
       width=150, height=84, units="mm", dpi=300)
```

Cross-validation and ROC

```r
# Reset data to where it was before modifying for plots
analysis_df_w_pol <- analysis_df %>%
  filter(!is.na(mean_polarity))

set.seed(4375)

# Change order of factors (required for using prSummary)
analysis_df_w_pol$mortality_30d <- factor(analysis_df_w_pol$mortality_30d, levels = c("1", "0"))
levels(analysis_df_w_pol$mortality_30d) <- c("expired", "survived")

train_control <- trainControl(method="repeatedcv", number=10, repeats=50,
                              summaryFunction = twoClassSummary, classProbs = TRUE, savePredictions = F

roc.model1 <- train(mortality_30d ~
                      mean_polarity + mean_subjectivity + sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

```r
roc.model2 <- train(mortality_30d ~
                      sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

```r
roc.model1$results
```

```
##   parameter       ROC       Sens      Spec      ROCSD      SensSD
## 1      none 0.7263407 0.02447877 0.9960569 0.00312658 0.001811878
##        SpecSD
## 1 0.000389735
```

```r
roc.model2$results
```

```
##   parameter       ROC       Sens     Spec       ROCSD      SensSD
## 1      none 0.7155225 0.01965525 0.995199 0.003046881 0.001690089
##         SpecSD
## 1 0.0003750374
```

```r
### Precision recall ###

train_control <- trainControl(method="repeatedcv", number=10, repeats=50,
                              summaryFunction = prSummary, classProbs = TRUE, savePredictions = FALSE)

roc.model1 <- train(mortality_30d ~
                      mean_polarity + mean_subjectivity + sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. AUC will be used instead.
```

```
roc.model2 <- train(mortality_30d ~
                      sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. AUC will be used instead.
```

```
roc.model1$results
```

```
##   parameter       AUC Precision     Recall          F       AUCSD
## 1      none 0.2785503 0.4912058 0.02449871 0.04666123 0.004332146
##   PrecisionSD    RecallSD         FSD
## 1  0.02838169 0.001902701 0.003526504
```

```
roc.model2$results
```

```
##   parameter       AUC Precision     Recall          F       AUCSD
## 1      none 0.2561555 0.3877138 0.01965684 0.03741096 0.003498693
##   PrecisionSD    RecallSD         FSD
## 1  0.02759292 0.001680017 0.003139239
```

Bootstrap CI's

```
### Paired method ###

# Reset data to where it was before modifying for plots.
# This is technically reduntant if we run the AUC's first, but
# doesn't hurt, and will allow users to jump over the previous AUC section.
analysis_df_w_pol <- analysis_df %>%
  filter(!is.na(mean_polarity))

analysis_df_w_pol$mortality_30d <- factor(analysis_df_w_pol$mortality_30d, levels = c("1", "0"))
levels(analysis_df_w_pol$mortality_30d) <- c("expired", "survived")

train_control <- trainControl(method="repeatedcv", number=10, repeats=1,
                              summaryFunction = twoClassSummary, classProbs = TRUE, savePredictions = T
seed <- 4375
set.seed(seed)
roc.model1 <- train(mortality_30d ~
                      mean_polarity + mean_subjectivity + sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

```
set.seed(seed)
roc.model2 <- train(mortality_30d ~
                      sapsii + first_careunit.x + gender,
                    data=analysis_df_w_pol,
                    method="glm", family=binomial(), trControl=train_control)
```

```
## Warning in train.default(x, y, weights = w, ...): The metric "Accuracy" was
## not in the result set. ROC will be used instead.
```

```r
library(precrec)
```

```
##
## Attaching package: 'precrec'
```

```
## The following object is masked from 'package:pROC':
##
##     auc
```

```r
n.boot <- 2000 # Use 2000 for paper

reduced.data <- analysis_df_w_pol[c("mean_polarity",
                                    "sapsii", "first_careunit",
                                    "gender", "mortality_30d")]

auc.df <- roc.model1$pred
auc.df$expired1 <- roc.model1$pred$expired
auc.df$expired2 <- roc.model2$pred$expired

auc.df <- auc.df[c("obs", "expired1", "expired2")]

for (j in c(1,2))
{
  auc.list <- lapply(1:n.boot, function(i){

    data.sample <- sample_n(auc.df, nrow(auc.df), replace = TRUE)
    auc.1 <- evalmod(labels = (data.sample$obs == "expired"), scores = data.sample$expired1)
    auc.2 <- evalmod(labels = (data.sample$obs == "expired"), scores = data.sample$expired2)

    auc.1 <- attr(auc.1, "auc")[j,4] # use [1,4] for ROC, and [2,4] for PRC
    auc.2 <- attr(auc.2, "auc")[j,4]

    auc.diff <- auc.1 - auc.2

    return(auc.diff)
  })

  auc.results <- do.call(rbind, auc.list)

  if(j == 1){
    print("AUROC difference bootstrap 95% CI:")
  }
  else {
    print("AUPRC difference bootstrap 95% CI:")
  }

  print(quantile(auc.results, c(0.025, 0.975)))
}
```

```
## [1] "AUROC difference bootstrap 95% CI:"
##        2.5%       97.5%
## 0.009882574 0.011731376
## [1] "AUPRC difference bootstrap 95% CI:"
##       2.5%      97.5%
## 0.02127058 0.02375802
```