# Data Challenge: Data Cleaning

Aldo Iturrios

3/28/2021

```
# Data
clean_data <- read.csv("../Data/patient_data/patient_dataframe.csv")
train_outcome <- read.csv("../Data/outcomes/train_outcome.csv")
test_outcome <- read.csv("../Data/outcomes/test_nolabel.csv")
```

In this RMD file, we'll be constructing new variables to summarize our time series data. Specifically, for each measure (~37 hospital measures), we'll be calculating: * The Mean (across time / columns) * The Median * The Standard Deviation

## Isolate Variable Names

```
# Removing variables that are currently named without a time stamp (e.g. ALT, ALP, etc.)
# This will help when doing the regex below
clean_data_2 <- clean_data[, -((ncol(clean_data)-36):ncol(clean_data))]
var_names <- unique(names(clean_data_2[ ,6:ncol(clean_data_2)]))
timestamps <- c(unique(str_extract(var_names, "[0-9]{2}.[0-9]{2}")))
measures <- unique(str_extract(var_names, "(?<=_)[a-zA-Z]+[0-9]*"))
```

### Rename Regresison Coeff variables to add *_reg_coeff suffix

```
# Rename reg_coeff variables
for (i in 1:length(measures)){
  names(clean_data)[names(clean_data) == measures[i]] <- paste(measures[i], "reg_coeff", sep="_")
}
names(clean_data[grep("reg_coeff", colnames(clean_data))])
```

```
##  [1] "ALP_reg_coeff"         "ALT_reg_coeff"         "AST_reg_coeff"
##  [4] "Albumin_reg_coeff"     "BUN_reg_coeff"         "Bilirubin_reg_coeff"
##  [7] "Cholesterol_reg_coeff" "Creatinine_reg_coeff"  "DiasABP_reg_coeff"
## [10] "FiO2_reg_coeff"        "GCS_reg_coeff"         "Glucose_reg_coeff"
## [13] "HCO3_reg_coeff"        "HCT_reg_coeff"         "HR_reg_coeff"
## [16] "K_reg_coeff"           "Lactate_reg_coeff"     "MAP_reg_coeff"
## [19] "MechVent_reg_coeff"    "Mg_reg_coeff"          "NIDiasABP_reg_coeff"
## [22] "NIMAP_reg_coeff"       "NISysABP_reg_coeff"    "Na_reg_coeff"
## [25] "PaCO2_reg_coeff"       "PaO2_reg_coeff"        "Platelets_reg_coeff"
## [28] "RespRate_reg_coeff"    "SAPS_reg_coeff"        "SaO2_reg_coeff"
```

```
## [31] "SysABP_reg_coeff"      "Temp_reg_coeff"        "TroponinI_reg_coeff"
## [34] "TroponinT_reg_coeff"   "Urine_reg_coeff"       "WBC_reg_coeff"
## [37] "pH_reg_coeff"
```

```
length(names(clean_data[grep("reg_coeff", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

### Create Mean, Median, SD, Min, and Max Variables for each Measure

```
for (i in 1:length(measures)){

  var_subset <- grep(paste("[0-9]{2}.[0-9]{2}_(", measures[i], ")", sep=""), colnames(clean_data))
  if (length(var_subset) > length(timestamps)) {
    print("Error in Subset")
  }
  data_subset <- clean_data[,var_subset]

  if (measures[i] != "SAPS"){
    clean_data[paste(measures[i], "mean", sep="_")] <- rowMeans(data_subset, na.rm = TRUE)
    clean_data[paste(measures[i], "median", sep="_")] <- apply(data_subset, 1, median, na.rm = TRUE)
    clean_data[paste(measures[i], "sd", sep="_")] <- apply(data_subset, 1, sd, na.rm = TRUE)
    clean_data[paste(measures[i], "min", sep="_")] <- apply(data_subset, 1, min, na.rm = TRUE)
    clean_data[paste(measures[i], "max", sep="_")] <- apply(data_subset, 1, max, na.rm = TRUE)
  }
  else{
    clean_data[paste(measures[i], "mean", sep="_")] <-clean_data[,var_subset]
    clean_data[paste(measures[i], "median", sep="_")] <- clean_data[,var_subset]
    clean_data[paste(measures[i], "sd", sep="_")] <- ifelse(is.na(clean_data[,var_subset]), 0, NA)
    clean_data[paste(measures[i], "min", sep="_")] <- clean_data[,var_subset]
    clean_data[paste(measures[i], "max", sep="_")] <- clean_data[,var_subset]
  }
}

# Verify: Amount of Summary Variables created is the same as No. of Measures
length(names(clean_data[grep("mean", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

```
length(names(clean_data[grep("median", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

```
length(names(clean_data[grep("_sd", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

```
length(names(clean_data[grep("_min", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

```
length(names(clean_data[grep("_max", colnames(clean_data))])) == length(measures)
```

```
## [1] TRUE
```

### Make the SAPS variable just one single variable

```
clean_data$SAPS <- clean_data$SAPS_median
var_remove <- grep("SAPS_", colnames(clean_data))
clean_data <- clean_data[, -var_remove]
```

### Fix the NA's in Min and Max

```
min_vars <- grep("_min", colnames(clean_data))
max_vars <- grep("_max", colnames(clean_data))

# Min: Replace Inf with NA's
for (i in min_vars){
  clean_data[, i][clean_data[, i] == Inf] <- NA
}

# Max: Replace -Inf with NA's
for (i in max_vars){
  clean_data[, i][clean_data[, i] == -Inf] <- NA
}
```

## Split Data into Train Data and Test Data

```
# Train Data
train_data <- clean_data[1:nrow(train_outcome), ]
train_data <- merge(train_data, train_outcome, by="id")

# Test Data
test_data <- clean_data[(nrow(train_outcome) + 1):nrow(clean_data), ]
test_data <- merge(test_data, test_outcome, by="id")

# Verify: Number of rows consistent
nrow(clean_data) == (nrow(train_data) + nrow(test_data))
```

```
## [1] TRUE
```

```r
# Verify No NA's for Outcome in Train Data
sum(is.na(train_data$outcome)) == 0
```

```
## [1] TRUE
```

```r
# Verify All NA's for Outcome and Score variable in Train Data
sum(!is.na(test_data$outcome)) == 0
```

```
## [1] TRUE
```

```r
sum(!is.na(train_data$score)) == 0
```

```
## [1] TRUE
```

### Save Files with all original variables (with NA's intact)

```r
write.csv(clean_data,"../Data/clean_data/clean_data.csv", row.names = FALSE)
write.csv(train_data,"../Data/clean_data/train_data.csv", row.names = FALSE)
write.csv(test_data,"../Data/clean_data/test_data.csv", row.names = FALSE)
```

### Save Files with ONLY Summary Variables (with NA's intact)

```r
time_var_subset <- grep(paste("[0-9]{2}.[0-9]{2}_.", sep=""), colnames(clean_data))
train_data_sumvars <- train_data[,-time_var_subset]
test_data_sumvars <- test_data[,-time_var_subset]
```

```r
write.csv(train_data_sumvars,"../Data/clean_data/train_data_sumvars.csv", row.names = FALSE)
write.csv(test_data_sumvars,"../Data/clean_data/test_data_sumvars.csv", row.names = FALSE)
```

### Save Files with ONLY Summary Variables (with Median Imputation)

```r
# Create copies data
train_data_med <- data.frame(train_data_sumvars)
test_data_med <- data.frame(test_data_sumvars)

# Median Imputation
for (i in 6:ncol(train_data_med)){
  train_data_med[, i][is.na(train_data_med[, i])] <- median(train_data_med[, i], na.rm = TRUE)
  test_data_med[, i][is.na(test_data_med[, i])] <- median(test_data_med[, i], na.rm = TRUE)
}

# Save Data files
write.csv(train_data_med,"../Data/clean_data/train_data_sumvars_med.csv", row.names = FALSE)
write.csv(test_data_med,"../Data/clean_data/test_data_sumvars_med.csv", row.names = FALSE)
```

### Save Files with ONLY Summary Variables (with Mean Imputation)

```
# Create copies data
train_data_mean <- data.frame(train_data_sumvars)
test_data_mean <- data.frame(test_data_sumvars)

# Mean Imputation
for (i in 6:ncol(train_data_mean)){
  train_data_mean[, i][is.na(train_data_mean[, i])] <- mean(train_data_mean[, i], na.rm = TRUE)
  test_data_mean[, i][is.na(test_data_mean[, i])] <- mean(test_data_mean[, i], na.rm = TRUE)
}

# Save Data files
write.csv(train_data_mean,"../Data/clean_data/train_data_sumvars_mean.csv", row.names = FALSE)
write.csv(test_data_mean,"../Data/clean_data/test_data_sumvars_mean.csv", row.names = FALSE)
```