Appendix: Code for the final project

0. Description

Outcome: length of stay (losdays2 or loshour)

1. Clean Data

1. read data

```
GHP_raw = read_excel("./GHProject_Dataset.xlsx") %>%
  clean_names()
```

We do not need to remove icu_flag

```
#histogram
GHP_raw %>%
  ggplot(aes(x = losdays2, color = icu_flag)) +
    geom_bar(data = GHP_raw %>% filter(icu_flag == 0), aes(x = losdays2)) +
    geom_bar(data = GHP_raw %>% filter(icu_flag == 1), aes(x = losdays2)) +
    theme_bw()

GHP_raw %>%
  ggplot(aes(y = losdays2, x = icu_flag, group = icu_flag)) +
    geom_boxplot()


GHP = GHP_raw %>%
  filter(ICU_Flag != 1)
```

2. change admitdtm into date formate, relevel cindex and mews based on project instruction, and keep the the first visit for each patient (70 observations were removed)

```
GHP = GHP_raw %>%
  separate(admitdtm, into = c("weekday", "date", "year"), sep = ",") %>%
  mutate(date = paste(date, " 2016"), date = as.Date(date, format = " %B %d %Y")) %>%
  dplyr::select(-year) %>%
  mutate(cindex = if_else(cindex == 1 & cindex == 2, 1, cindex),
         cindex = if_else(cindex == 3 & cindex == 4, 2, cindex),
         cindex = if_else(cindex >= 5, 3, cindex)) %>% # 0 = normal, 1 = mild, 2 = moderate, 3 = severe
  mutate(mews = if_else(mews == 1, 0, mews),
         mews = if_else((mews == 2) + (mews == 3) == 1, 1, mews),
         mews = if_else((mews == 4) + (mews == 5) == 1, 2, mews),
         mews = if_else(mews > 5, 3, mews)) %>% # 0 = normal, 1 = increase caution, 2 = further deterio
  group_by(patientid) %>%
  filter(date == min(date)) %>%
  ungroup()
```

3. We wanted to investigate the relationship between the length of stay and the health/socioeconomics status of patients, so we decided the variables of intrest to exclude "postalcode", "facilityzip", "facilityname", "date", "weekday", "patientid", and "visitid". Because we also noticed that "loshours" and "losday2" are in fact the same variable

with different unit, we chose to only use the "loshours". (why we choose vital signs over mews?)

```
GHP = GHP %>%
  dplyr::select(-postalcode, -facilityzip, -facilityname, -date, -weekday, -mews, -patientid, -visitid,
```

4. Missing values were replaced with the median of that variable for numeric data. We choose median over mean to address potential skewness in variables.

```
replace_na_w_median = function(vector){
  if (is.numeric(vector)) {
    vector[is.na(vector)] = median(vector, na.rm = TRUE)
  } else if (is.character(vector)) {
    vector[is.na(vector)] = "unknown"
  }
  return(vector)
}

GHP_missing_to_median = map_df(GHP, replace_na_w_median)
```

2. Exploratory analysis

1. We first examine the distribution of the outcome, "losdays2", with boxplot and histogram.

```
attach(GHP_missing_to_median)
boxplot(loshours)
histogram(loshours)
```

we can see the distribution of the outcome is highly right skewed, so we performed log transformation to it, and the transformed outcome looks symmetrical.

```
attach(GHP_missing_to_median)
boxplot(log(loshours))
histogram(log(loshours))
GHP_missing_to_median = GHP_missing_to_median %>%
  mutate(log_los = log(loshours)) %>%
  dplyr::select(-loshours)
```

2. We use "ggpairs" to investigate the distribution of each variable and check whether they have potential association. (this may goes to the appendix)

```
ggpairs(GHP_missing_to_median) + ggplot2::theme_bw()
```

3. model building

anova tests for categorical data: all categorical variable shows significant difference among their levels, except gender. However, I still include gender in the full model, because ref. . . ?.

```
slr_gender = lm(log_los ~ gender, data = GHP_missing_to_median)
slr_maritalstatus = lm(log_los ~ maritalstatus, data = GHP_missing_to_median)
slr_insurancetype = lm(log_los ~ insurancetype, data = GHP_missing_to_median)
slr_religion = lm(log_los ~ religion, data = GHP_missing_to_median)
```

```
anova(slr_gender)
anova(slr_maritalstatus)
anova(slr_insurancetype)
anova(slr_religion)
```

what is the boxplot of marital status

```
GHP_missing_to_median %>%
  ggplot(aes(x = maritalstatus, y = log_los)) +
    geom_boxplot()
```

We noticed that there is only one civial union, so we decided to combine this observation into married

```
GHP_missing_to_median = GHP_missing_to_median %>%
  mutate(maritalstatus = if_else(maritalstatus == "Civil Union", "Married", maritalstatus))
```

1. The full model is

```
# fit all predictor:
mult.reg = lm(log_los ~ bmi + o2sat + heartrate + bpsystolic + bpdiastolic + respirationrate + temperatu
  summary(mult.reg)
```

2. Automatic procedure to select varialbles

1. stepwise

```
stepwise <- step(mult.reg, direction = 'both')
summary(stepwise)
```

2. forward

```
fwd <- step(mult.reg, direction = 'forward')
summary(fwd)
```

3. backward

```
bwd <- step(mult.reg, direction = 'backward')
summary(bwd)
```

4. reduced model, without o2sat

```
auto_reduce3 <- update(stepwise, . ~ . -o2sat)
summary(auto_reduce3)
```

5. Anova test for nested models

```
anova(auto_reduce3, stepwise) # show this
```

So, it suggests that o2sat can be removed. Thus, the model from the auto method is :

```
auto_model = auto_reduce3
summary(auto_model)
```

3. Criterion based model selection

Calculate the criterion for each model size.

```
best <- function(model, ...)
{
  subsets <- regsubsets(formula(model), model.frame(model), nvmax = 19, ...)
  subsets <- with(summary(subsets),
                  cbind(p = as.numeric(rownames(which)), which, rss, rsq, adjr2, cp, bic))

  return(subsets)
}

best_subsets = best(mult.reg, nbest = 1)
pander(best_subsets)


par(mar=c(4,4,1,1))
par(mfrow=c(1,2))

plot(1:19, best_subsets[, "cp"], xlab="No. of predictor", ylab="Cp Statistic")
abline(0,1)

plot(1:19, best_subsets[, "adjr2"], xlab="No. of predictor", ylab="Adj R2")
```

1. The selected criterion-based model is

```
crit = lm(log_los ~ o2sat + heartrate + bpsystolic + bpdiastolic + respirationrate + temperature + is30
  summary(crit)
```

However, there are still several unsignificant predictors, so I tried to remove them from the model.

2. reduced model without o2sat

```
crit_reduce1 = update(crit, . ~ . -o2sat)
summary(crit_reduce1)
```

3. reduced model without religion

```
crit_reduce2 = update(crit, . ~ . -religion)
summary(crit_reduce2)
```

4. reduced model without maritalstatus

```
crit_reduce3 = update(crit, . ~ . -maritalstatus)
summary(crit_reduce3)
```

5. reduce model without o2sat and religion and maritalstatus and

```
crit_reduce4 = update(crit_reduce1, . ~ . -religion)
crit_reduce5 = update(crit_reduce4, . ~ . -maritalstatus)
summary(crit_reduce5)
```

6. Use anova to test these reduced models vs. the full model

```
anova(crit_reduce1, crit)
anova(crit_reduce2, crit)
anova(crit_reduce3, crit)
anova(crit_reduce5, crit) #show this
```

Thus, the full model is not superior to crit_reduce5, based on the principle of parsimony, we choose the crit_reduce5

```
criterion_based_model = crit_reduce5
summary(criterion_based_model)
```

4. THe final model

Since the criterion-based model and the automatic selected model are the same, we have the final model shown as below:

```
final_model = criterion_based_model
summary(final_model)
write_csv(broom::tidy(final_model), "model estimate.csv")
```

4. Model diagnostic and remedie

1. check collinearity using the VIF cut-off at 10

```
pander(vif(final_model))
```

2. diagnostic plots

```
par(mfrow = c(2, 2))
plot(final_model)
```

3. detect and remove outliers

```
#detect outliers
stu_res<-rstandard(final_model)
outliers_y<-stu_res[abs(stu_res)>2.5]
outliers_y

#remove outliers
rm_list_outlier = c(65, 75)
wholelist = 1:nrow(GHP_missing_to_median)
keep_list_outlier <- wholelist[!wholelist %in% rm_list_outlier]

GHP_outliers = GHP_missing_to_median[rm_list_outlier, ]
GHP_missing_to_median_clean = GHP_missing_to_median[keep_list_outlier, ]
```

4. detect and remove influential points and leverage points

```
inf = influence.measures(final_model)

#remove outliers
rm_list_influential = c(47, 54, 75, 55, 65, 51, 66, 46, 60, 53)
wholelist_influential = 1:nrow(GHP_missing_to_median_clean)
keep_list_influential <- wholelist_influential[!wholelist_influential %in% rm_list_influential]

GHP_influential=GHP_missing_to_median[rm_list_influential, ]
GHP_missing_to_median_clean = GHP_missing_to_median_clean[keep_list_influential, ]
```

5. Build the final model without outliers, influential points, and leverage points

Fit a full model without outliers, influential points, and leverage points

```
mult.reg_1 = lm(log_los ~ bmi + o2sat + heartrate + bpsystolic + bpdiastolic + respirationrate + tempera
  summary(mult.reg_1)
```

1. stepwise

```
stepwise_1 <- step(mult.reg_1, direction = 'both')
summary(stepwise_1)
```

2. criterion based selection

```
best <- function(model, ...)
{
  subsets <- regsubsets(formula(model), model.frame(model), nvmax = 18, ...)
  subsets <- with(summary(subsets),
                  cbind(p = as.numeric(rownames(which)), which, rss, rsq, adjr2, cp, bic))

  return(subsets)
}

best_subsets_1 = best(mult.reg_1, nbest = 1)
pander(best_subsets_1)


par(mar=c(4,4,1,1))
par(mfrow=c(1,2))

plot(1:18, best_subsets_1[, "cp"], xlab="No. of predictor", ylab="Cp Statistic")
abline(0,1)

plot(1:18, best_subsets_1[, "adjr2"], xlab="No. of predictor", ylab="Adj R2")
```

criterion based model with clean data set.

```
criterion_based_model_1 = lm(log_los ~ bmi + o2sat + heartrate + bpsystolic + bpdiastolic + respiration
  summary(criterion_based_model_1)
```

Anova test with partial F test to compare the two models

```
anova(stepwise_1, criterion_based_model_1)
```

This anova test suggest that the Crierion based model is not superior. So, with the "stepwise_1", I tried to remove its nonsignificant predictors,

```
stepwise_1_reduced = update(stepwise_1, . ~ . -maritalstatus)
stepwise_1_reduced = update(stepwise_1_reduced, . ~ . -o2sat)
summary(stepwise_1_reduced)
```

Still use Anova to compare the reduced model with the bigger model.

```
anova(stepwise_1_reduced, stepwise_1)
```

The result suggest that the bigger model is not superior, so we used the smaller model as our final model.

```
final_model_clean = stepwise_1_reduced
pander(broom::tidy(summary(final_model_clean)))
```

```r
write_csv(broom::tidy(summary(final_model_clean)), "estimates.csv")
```

The diagnostics of this new final model

```r
par(mfrow = c(2, 2))
plot(stepwise_1_reduced)
```

Check for collinearity

```r
pander(vif(final_model_clean))
```

6. Model validation

Use bootstrap to validate the model.

```r
# Bootstrap to assess the variability of model estimates#
set.seed(1)

# Our usual regression, no bootstrap yet
boot.fn <- function(data, index){
    return(coef(lm(log_los ~ heartrate + bpsystolic + bpdiastolic + respirationrate + temperature + is3
}

# Compute the estimates by sampling with replacement
# Sample chooses 3600 observations from 3600, with replacement
# Might have duplicates
set.seed(1)

# One draw
boot.fn(GHP_missing_to_median_clean,sample(3600,3600,replace=T))

# Use function boot() to repeat the sampling 1000 times.
boot = boot(GHP_missing_to_median_clean, boot.fn, 1000)
boot_data = data.frame((boot))
boot_data = t(boot_data)
boot_data = data.frame(boot_data)
write_csv(boot_data, "boot.csv")
```

The bootstrap results and the final model have similar coefficients and therefore justified the consistency of our model.