STAT_E-109_Project

Erin Rebholz, Nadia Zafar, Imran Naskani, Max Yanover

2023-03-22

Intitial package installation and data loading

```
#Load in csv data for project

data <- read.csv('SDOH_Quality.csv')

count_initial <- nrow(data)

str(data)
```

```
## 'data.frame':    5317 obs. of  36 variables:
##  $ facility_ID                : chr  "10001" "10005" "10006" "10007" ...
##  $ facility_name              : chr  "SOUTHEAST HEALTH MEDICAL CENTER" "MARSHALL MEDICAL CENTERS" "
##  $ address                    : chr  "1108 ROSS CLARK CIRCLE" "2505 U S HIGHWAY 431 NORTH" "1701 VE
##  $ city                       : chr  "DOTHAN" "BOAZ" "FLORENCE" "OPP" ...
##  $ state                      : chr  "AL" "AL" "AL" "AL" ...
##  $ zip                        : int  36301 35957 35630 36467 36049 35235 35968 35007 35233 35660 ..
##  $ county                     : chr  "HOUSTON" "MARSHALL" "LAUDERDALE" "COVINGTON" ...
##  $ hospital_type              : chr  "Acute Care Hospitals" "Acute Care Hospitals" "Acute Care Hosp
##  $ hospital_ownership         : chr  "Government - Hospital District or Authority" "Government - Ho
##  $ hosp_overall_rating        : int  3 3 2 3 NA 2 3 4 NA 3 ...
##  $ census_region              : chr  "South" "South" "South" "South" ...
##  $ census_division            : chr  "East South Central" "East South Central" "East South Central"
##  $ median_age                 : chr  "39.1" "38.3" "33.8" "47.4" ...
##  $ per_white_non_hisp         : chr  "61.8" "81" "71.6" "83" ...
##  $ med_inc_15plus_12mo        : chr  "25326" "22419" "21362" "21490" ...
##  $ per_below_poverty          : chr  "20.7" "24.2" "22.9" "18.4" ...
##  $ per_college_grad_deg_25_plus: chr  "17.2" "13.7" "23.5" "12.1" ...
##  $ COMP_HIP_KNEE              : chr  "2.4" "1.8" "3.4" "" ...
##  $ MORT_30_AMI                : chr  "12.4" "12.6" "16.5" "" ...
##  $ MORT_30_CABG               : chr  "4.7" "" "3.5" "" ...
##  $ MORT_30_COPD               : chr  "8.5" "8.1" "7.8" "10.3" ...
##  $ MORT_30_HF                 : chr  "8.3" "16.9" "12.2" "13.9" ...
##  $ MORT_30_PN                 : chr  "15.9" "21.8" "17.8" "21.7" ...
##  $ MORT_30_STK                : chr  "16.4" "16.6" "18.9" "" ...
##  $ PSI_03                     : chr  "0.23" "0.86" "1.83" "0.32" ...
##  $ PSI_04                     : chr  "173.39" "142.88" "157.42" "" ...
##  $ PSI_06                     : chr  "0.17" "0.17" "0.26" "0.18" ...
##  $ PSI_08                     : chr  "0.1" "0.06" "0.05" "0.07" ...
##  $ PSI_09                     : chr  "2.33" "2.08" "3.46" "2.37" ...
```

```
##  $ PSI_10                      : chr  "0.61" "0.76" "0.65" "0.91" ...
##  $ PSI_11                      : chr  "8.92" "6.87" "3.89" "6.01" ...
##  $ PSI_12                      : chr  "3.33" "2.54" "2.8" "3.8" ...
##  $ PSI_13                      : chr  "5.98" "3.44" "3.72" "4.05" ...
##  $ PSI_14                      : chr  "0.65" "0.76" "0.68" "" ...
##  $ PSI_15                      : chr  "1.21" "0.87" "1.33" "1.02" ...
##  $ PSI_90                      : chr  "1.01" "0.91" "1.1" "0.99" ...
```

```
#Filter to only acute care and critical access facilities

data <- data %>% filter(hospital_type == 'Acute Care Hospitals' |  hospital_type == 'Critical Access Hos

(count_cah_acutre <- nrow(data))
```

```
## [1] 4585
```

```
# Drop facilities where zip code census demographic data is not available

data <- data %>% filter(median_age != "#N/A")

(count_cah_acutre_zipNA <- nrow(data))
```

```
## [1] 4407
```

```
datana <- data %>% filter(!is.na(hosp_overall_rating ))
str(datana)
```

```
## 'data.frame':    2970 obs. of  36 variables:
##  $ facility_ID                 : chr  "10001" "10005" "10006" "10007" ...
##  $ facility_name               : chr  "SOUTHEAST HEALTH MEDICAL CENTER" "MARSHALL MEDICAL CENTERS" "I
##  $ address                     : chr  "1108 ROSS CLARK CIRCLE" "2505 U S HIGHWAY 431 NORTH" "1701 VE
##  $ city                        : chr  "DOTHAN" "BOAZ" "FLORENCE" "OPP" ...
##  $ state                       : chr  "AL" "AL" "AL" "AL" ...
##  $ zip                         : int  36301 35957 35630 36467 35235 35968 35007 35660 36360 36116 ..
##  $ county                      : chr  "HOUSTON" "MARSHALL" "LAUDERDALE" "COVINGTON" ...
##  $ hospital_type               : chr  "Acute Care Hospitals" "Acute Care Hospitals" "Acute Care Hospi
##  $ hospital_ownership          : chr  "Government - Hospital District or Authority" "Government - Hos
##  $ hosp_overall_rating         : int  3 3 2 3 2 3 4 3 4 2 ...
##  $ census_region               : chr  "South" "South" "South" "South" ...
##  $ census_division             : chr  "East South Central" "East South Central" "East South Central"
##  $ median_age                  : chr  "39.1" "38.3" "33.8" "47.4" ...
##  $ per_white_non_hisp          : chr  "61.8" "81" "71.6" "83" ...
##  $ med_inc_15plus_12mo         : chr  "25326" "22419" "21362" "21490" ...
##  $ per_below_poverty           : chr  "20.7" "24.2" "22.9" "18.4" ...
##  $ per_college_grad_deg_25_plus: chr  "17.2" "13.7" "23.5" "12.1" ...
##  $ COMP_HIP_KNEE               : chr  "2.4" "1.8" "3.4" "" ...
##  $ MORT_30_AMI                 : chr  "12.4" "12.6" "16.5" "" ...
##  $ MORT_30_CABG                : chr  "4.7" "" "3.5" "" ...
##  $ MORT_30_COPD                : chr  "8.5" "8.1" "7.8" "10.3" ...
##  $ MORT_30_HF                  : chr  "8.3" "16.9" "12.2" "13.9" ...
##  $ MORT_30_PN                  : chr  "15.9" "21.8" "17.8" "21.7" ...
##  $ MORT_30_STK                 : chr  "16.4" "16.6" "18.9" "" ...
```

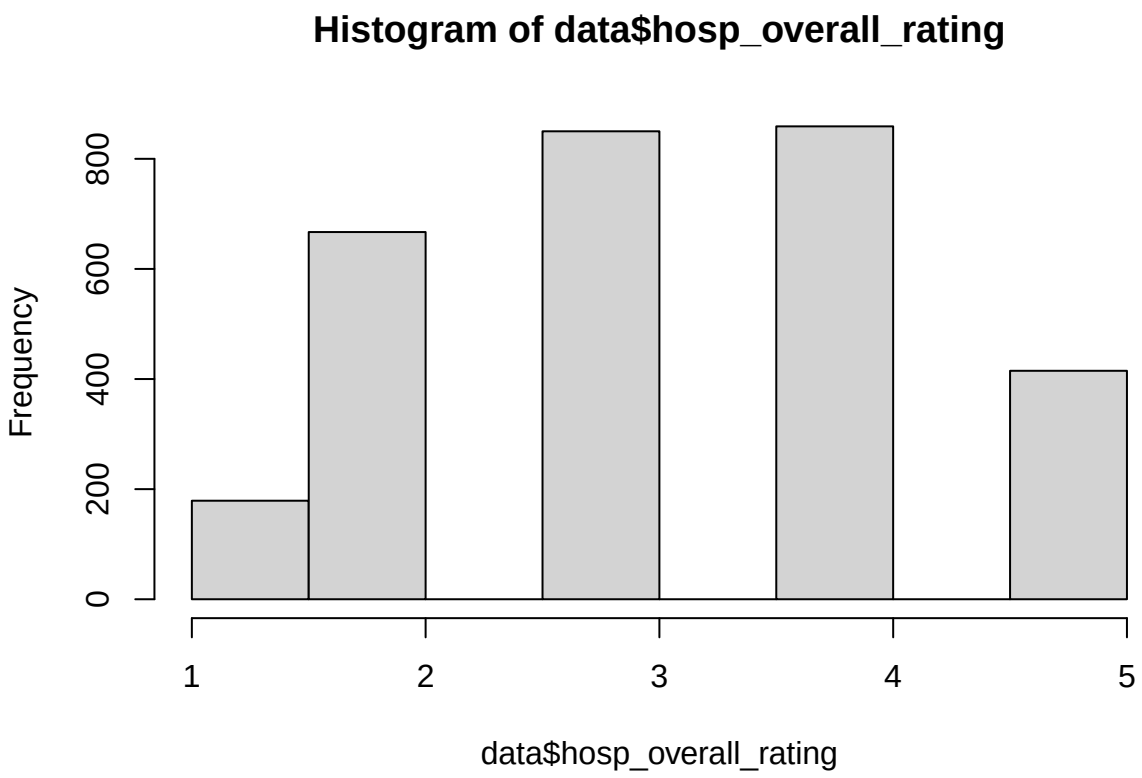
```
##  $ PSI_03                    : chr  "0.23" "0.86" "1.83" "0.32" ...
##  $ PSI_04                    : chr  "173.39" "142.88" "157.42" "" ...
##  $ PSI_06                    : chr  "0.17" "0.17" "0.26" "0.18" ...
##  $ PSI_08                    : chr  "0.1" "0.06" "0.05" "0.07" ...
##  $ PSI_09                    : chr  "2.33" "2.08" "3.46" "2.37" ...
##  $ PSI_10                    : chr  "0.61" "0.76" "0.65" "0.91" ...
##  $ PSI_11                    : chr  "8.92" "6.87" "3.89" "6.01" ...
##  $ PSI_12                    : chr  "3.33" "2.54" "2.8" "3.8" ...
##  $ PSI_13                    : chr  "5.98" "3.44" "3.72" "4.05" ...
##  $ PSI_14                    : chr  "0.65" "0.76" "0.68" "" ...
##  $ PSI_15                    : chr  "1.21" "0.87" "1.33" "1.02" ...
##  $ PSI_90                    : chr  "1.01" "0.91" "1.1" "0.99" ...
```

```r
#Switch to numeric variables fo censuse/SDOH measures

data$median_age <- as.numeric(data$median_age)
data$per_white_non_hisp <- as.numeric(data$per_white_non_hisp)
data$med_inc_15plus_12mo <- as.numeric(data$med_inc_15plus_12mo)
data$per_below_poverty <- as.numeric(data$per_below_poverty)
data$per_college_grad_deg_25_plus <- as.numeric(data$per_college_grad_deg_25_plus)

#Switch to numeric variables for quality measures
data$COMP_HIP_KNEE <- as.numeric(data$COMP_HIP_KNEE)
data$MORT_30_AMI <- as.numeric(data$MORT_30_AMI)
data$MORT_30_CABG <- as.numeric(data$MORT_30_CABG)
data$MORT_30_COPD <- as.numeric(data$MORT_30_COPD)
data$MORT_30_HF <- as.numeric(data$MORT_30_HF)
data$MORT_30_PN <- as.numeric(data$MORT_30_PN)
data$MORT_30_STK <- as.numeric(data$MORT_30_STK)
data$PSI_03 <- as.numeric(data$PSI_03)
data$PSI_04 <- as.numeric(data$PSI_04)
data$PSI_06 <- as.numeric(data$PSI_06)
data$PSI_08 <- as.numeric(data$PSI_08)
data$PSI_09 <- as.numeric(data$PSI_09)
data$PSI_10 <- as.numeric(data$PSI_10)
data$PSI_12 <- as.numeric(data$PSI_12)
data$PSI_11 <- as.numeric(data$PSI_11)
data$PSI_13 <- as.numeric(data$PSI_13)
data$PSI_11 <- as.numeric(data$PSI_11)
data$PSI_14 <- as.numeric(data$PSI_14)
data$PSI_15 <- as.numeric(data$PSI_15)
data$PSI_90 <- as.numeric(data$PSI_90)

#Switch chr variables to factor
data$city <- as.factor(data$city)
data$state <- as.factor(data$state)
data$hospital_type <- as.factor(data$hospital_type)
data$hospital_ownership <- as.factor(data$hospital_ownership)
data$census_region <- as.factor(data$census_region)
data$census_division <- as.factor(data$census_division)

#check to make sure all are now integers
str(data)
```

```
## 'data.frame':    4407 obs. of  36 variables:
##  $ facility_ID                 : chr  "10001" "10005" "10006" "10007" ...
##  $ facility_name               : chr  "SOUTHEAST HEALTH MEDICAL CENTER" "MARSHALL MEDICAL CENTERS" "
##  $ address                     : chr  "1108 ROSS CLARK CIRCLE" "2505 U S HIGHWAY 431 NORTH" "1701 VE
##  $ city                        : Factor w/ 2835 levels "ABBEVILLE","ABERDEEN",..: 666 251 835 1846 14
##  $ state                       : Factor w/ 51 levels "AK","AL","AR",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ zip                         : int  36301 35957 35630 36467 36049 35235 35968 35007 35233 35660 ..
##  $ county                      : chr  "HOUSTON" "MARSHALL" "LAUDERDALE" "COVINGTON" ...
##  $ hospital_type               : Factor w/ 2 levels "Acute Care Hospitals",..: 1 1 1 1 1 1 1 1 1 1 .
##  $ hospital_ownership          : Factor w/ 10 levels "Government - Federal",..: 2 2 6 10 6 10 6 10 10
##  $ hosp_overall_rating         : int  3 3 2 3 NA 2 3 4 NA 3 ...
##  $ census_region               : Factor w/ 4 levels "Midwest","Northeast",..: 3 3 3 3 3 3 3 3 3 3 ..
##  $ census_division             : Factor w/ 9 levels "East North Central",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ median_age                  : num  39.1 38.3 33.8 47.4 41.7 37.1 36.4 37.6 28.4 45 ...
##  $ per_white_non_hisp          : num  61.8 81 71.6 83 72.9 32.9 71.6 70.6 54 73.4 ...
##  $ med_inc_15plus_12mo         : num  25326 22419 21362 21490 21429 ...
##  $ per_below_poverty           : num  20.7 24.2 22.9 18.4 8.1 19.5 12.7 7.8 38 23.3 ...
##  $ per_college_grad_deg_25_plus: num  17.2 13.7 23.5 12.1 20.8 27 13.9 34.9 83.8 16.7 ...
##  $ COMP_HIP_KNEE               : num  2.4 1.8 3.4 NA NA 2.4 2.4 2.5 NA 2.8 ...
##  $ MORT_30_AMI                 : num  12.4 12.6 16.5 NA NA 13.8 12.9 11.8 NA NA ...
##  $ MORT_30_CABG                : num  4.7 NA 3.5 NA NA 3.5 NA 2.9 NA NA ...
##  $ MORT_30_COPD                : num  8.5 8.1 7.8 10.3 NA 8.2 8 8.2 NA 7.3 ...
##  $ MORT_30_HF                  : num  8.3 16.9 12.2 13.9 NA 12.2 11.5 12.1 NA 13.4 ...
##  $ MORT_30_PN                  : num  15.9 21.8 17.8 21.7 19.7 17.1 20.2 16.7 NA 21.7 ...
##  $ MORT_30_STK                 : num  16.4 16.6 18.9 NA NA 15.2 NA 12 NA 17.5 ...
##  $ PSI_03                      : num  0.23 0.86 1.83 0.32 0.5 0.05 0.22 0.07 0.55 0.85 ...
##  $ PSI_04                      : num  173 143 157 NA NA ...
##  $ PSI_06                      : num  0.17 0.17 0.26 0.18 0.19 0.16 0.18 0.16 0.19 0.17 ...
##  $ PSI_08                      : num  0.1 0.06 0.05 0.07 0.07 0.05 0.07 0.05 0.07 0.13 ...
##  $ PSI_09                      : num  2.33 2.08 3.46 2.37 NA 2.31 2.33 3.35 2.34 2.24 ...
##  $ PSI_10                      : num  0.61 0.76 0.65 0.91 NA 0.98 0.9 1.56 0.91 NA ...
##  $ PSI_11                      : num  8.92 6.87 3.89 6.01 NA ...
##  $ PSI_12                      : num  3.33 2.54 2.8 3.8 NA 2.68 3.15 2.68 3.33 4.25 ...
##  $ PSI_13                      : num  5.98 3.44 3.72 4.05 NA 4.63 3.92 4.07 NA NA ...
##  $ PSI_14                      : num  0.65 0.76 0.68 NA NA 0.75 0.78 0.99 NA 0.74 ...
##  $ PSI_15                      : num  1.21 0.87 1.33 1.02 NA 0.8 0.99 0.81 NA 1.24 ...
##  $ PSI_90                      : num  1.01 0.91 1.1 0.99 NA 0.82 0.91 1.07 0.98 1.09 ...
```

Initial EDA - Overall Star Rating

```
#Histogram of Overall Hospital Star Rating

hist(data$hosp_overall_rating)
```




```
#Figure 2
#Table of Overall Hospital Star Rating
(counts <- table(data$hosp_overall_rating))
```

```
##
##   1   2   3   4   5
## 179 667 850 859 415
```

```
#Figure 2
#total number of rows in data
(nrows <- nrow(data))
```

```
## [1] 4407
```

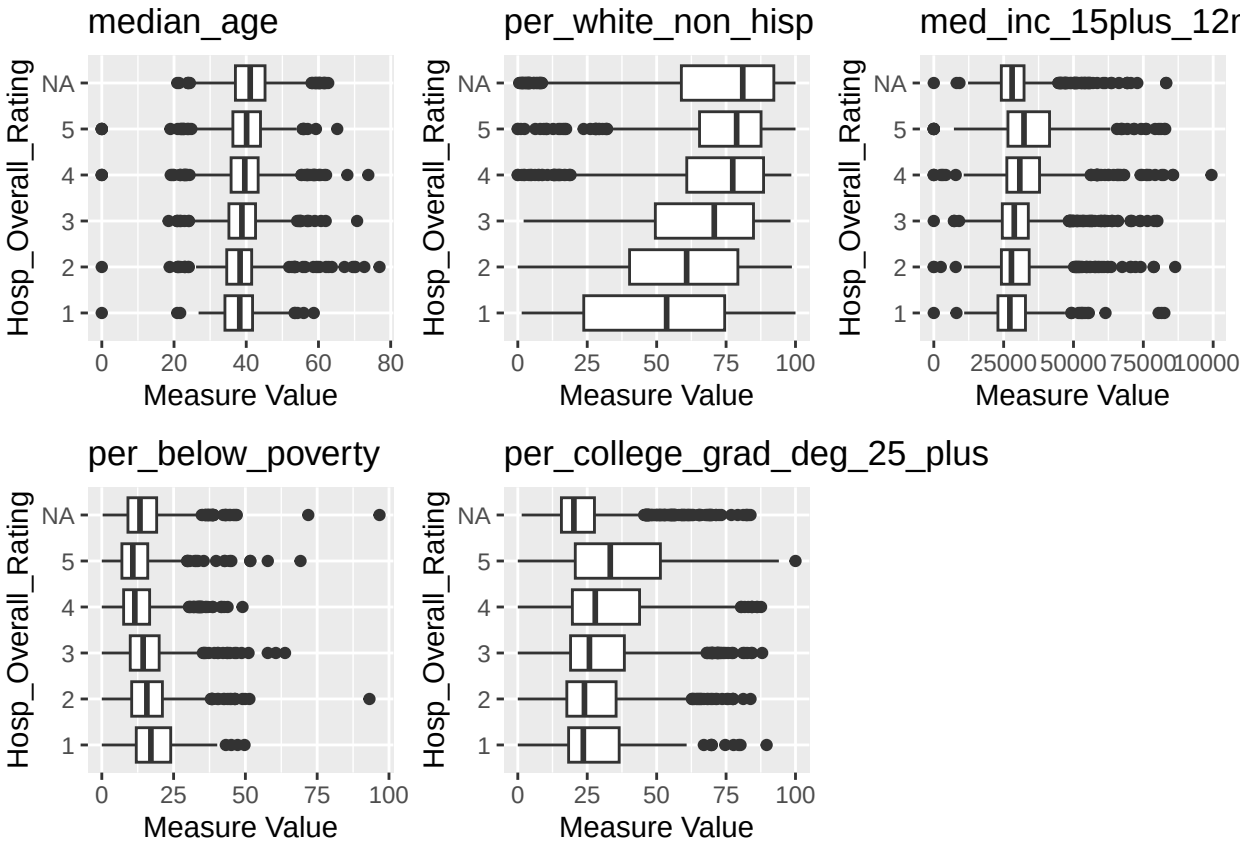
```
#------- Create a Data Table to Support Model Overview ----------
avg.PSI_90 <-
data %>% filter(!is.na(PSI_90)) %>%
summarize(Average= round(mean(PSI_90),3), Count = n(), Std_Dev = round(sd(PSI_90),3),.groups = 'drop')
row.names(avg.PSI_90) <- c("PSI_90")

avg.PSI_90 <- as.data.frame(avg.PSI_90)

avg.MORT_30_PN <-
data %>% filter(!is.na(MORT_30_PN)) %>%
```

```
summarize(Average= round(mean(MORT_30_PN),3), Count = n(), Std_Dev = round(sd(MORT_30_PN),3),.groups =
row.names(avg.MORT_30_PN) <- c("MORT_30_PN")

avg.MORT_30_PN <- as.data.frame(avg.MORT_30_PN)

avg.MORT_30_HF <-
data %>% filter(!is.na(MORT_30_HF)) %>%
summarize(Average= round(mean(MORT_30_HF),3), Count = n(), Std_Dev = round(sd(MORT_30_HF),3),.groups =
row.names(avg.MORT_30_HF) <- c("MORT_30_HF")

avg.MORT_30_HF <- as.data.frame(avg.MORT_30_HF)

avg.MORT_30_COPD <-
data %>% filter(!is.na(MORT_30_COPD)) %>%
summarize(Average= round(mean(MORT_30_COPD),3), Count = n(), Std_Dev = round(sd(MORT_30_COPD),3),.groups
row.names(avg.MORT_30_COPD) <- c("MORT_30_COPD")

avg.MORT_30_COPD <- as.data.frame(avg.MORT_30_COPD)

avg.rating <-
data %>% filter(!is.na(hosp_overall_rating)) %>%
summarize(Average= round(mean(hosp_overall_rating),3), Count = n(), Std_Dev = round(sd(hosp_overall_rati
row.names(avg.rating) <- c("Hospital Overall Rating")

avg.rating <- as.data.frame(avg.rating)


Blended <- rbind(avg.MORT_30_COPD,avg.MORT_30_HF,avg.MORT_30_PN,avg.PSI_90,avg.rating)

t(Blended)
```

```
##         MORT_30_COPD MORT_30_HF MORT_30_PN   PSI_90 Hospital Overall Rating
## Average        8.497     11.456     16.833    0.975                   3.224
## Count       2724.000   2969.000   3448.000 2820.000                2970.000
## Std_Dev        1.142      1.751      2.268    0.161                   1.124
```

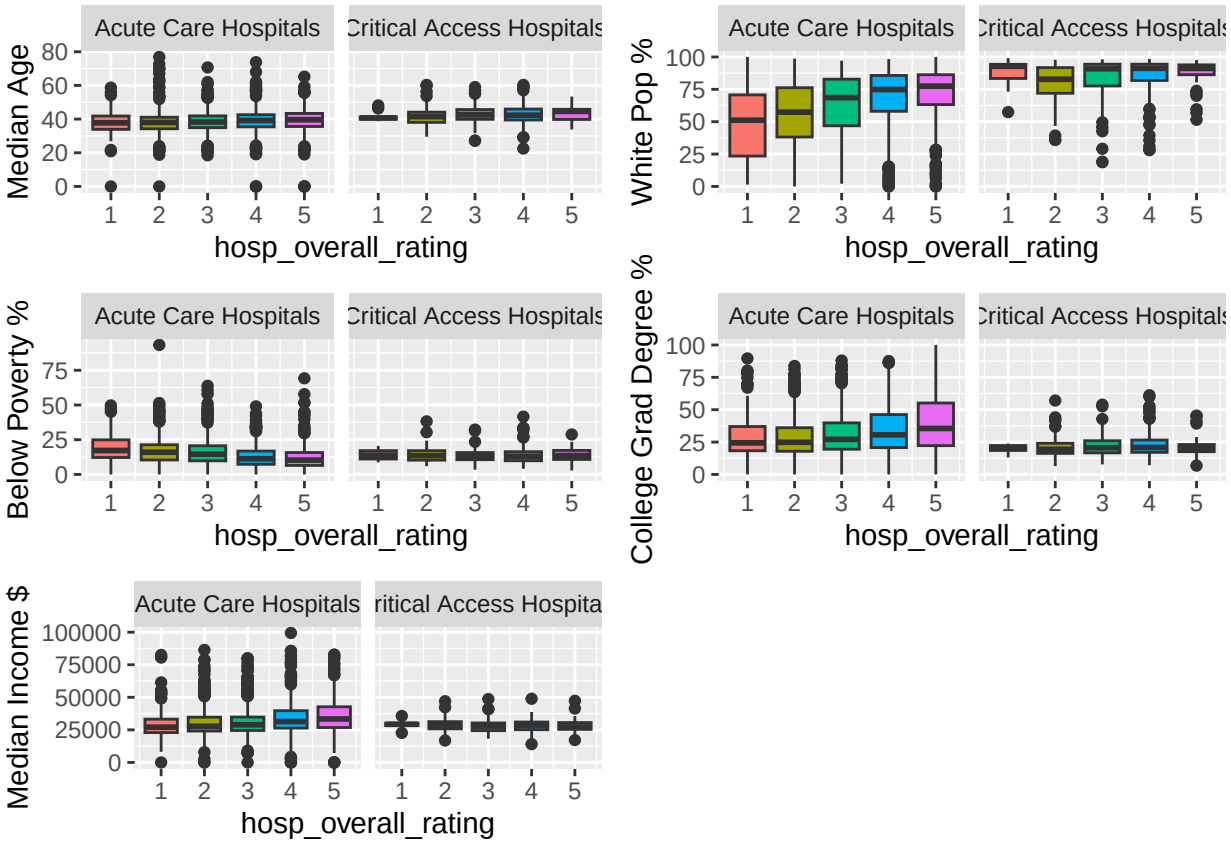
```
##SDOH Measures by Overall Hospital Rating

require(gridExtra)

plot1 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,13]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[13]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")


plot2 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,14]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[14]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")
```

```
plot3 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,15]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[15]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")


plot4 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,16]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[16]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")

plot5 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,17]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[17]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")

#Aggregate into a single graph
grid.arrange(plot1, plot2, plot3, plot4, plot5, ncol=3)
```




```
#Save

g <- arrangeGrob(plot1, plot2, plot3, plot4, plot5, ncol=3) #generates g
```

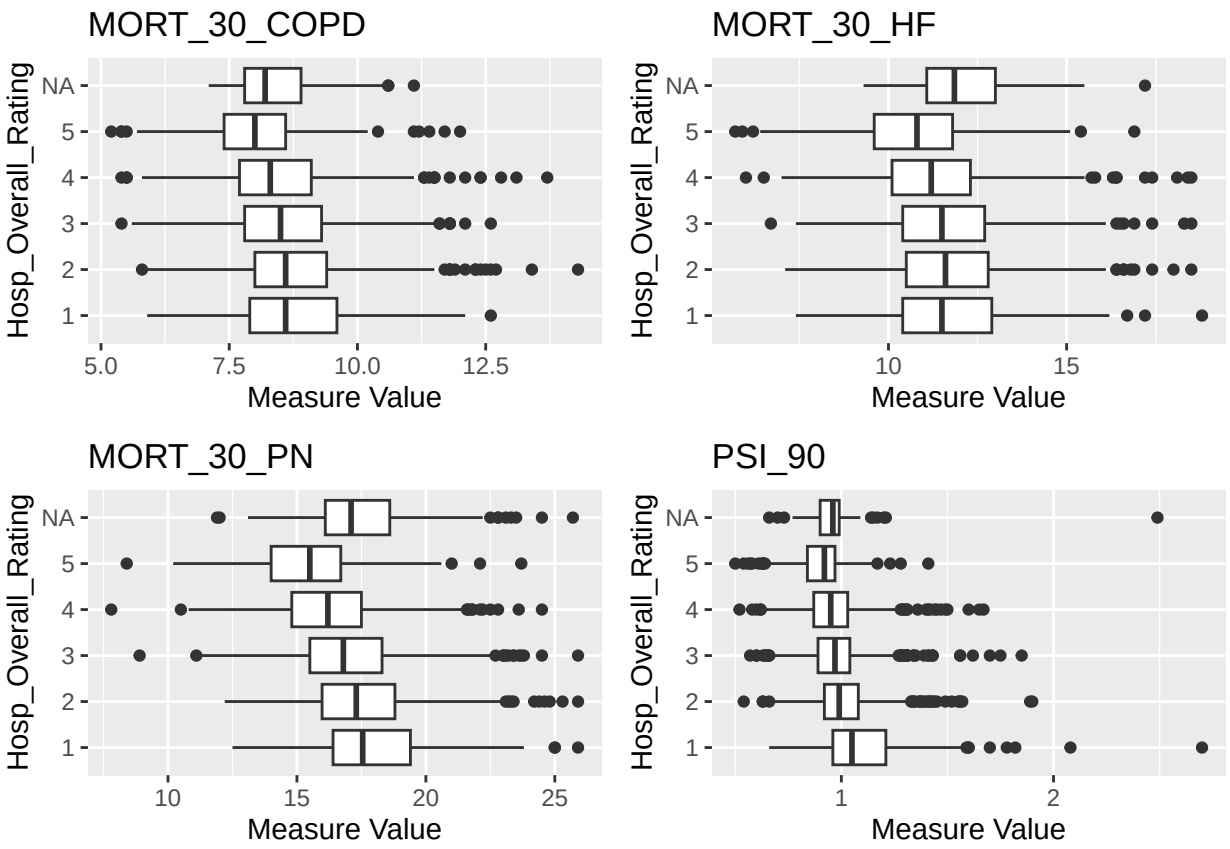
```r
ggsave("SDOH.png", g, width = 20, height = 15, units = "cm")

##------- Boxplot: Star Rating vs Social Indicators -------------

#------- Boxplot: Star Rating vs Median Age -------------
social.ind1 <- data %>% filter(!is.na(hosp_overall_rating)) %>%
ggplot(aes(x = hosp_overall_rating, y = median_age, fill = as.factor(hosp_overall_rating))) +
geom_boxplot(show.legend = F)  + facet_wrap(~hospital_type) +
ylab('Median Age')

#------- Boxplot: Star Rating vs Percentage White Population -------------
social.ind2 <- data %>% filter(!is.na(hosp_overall_rating)) %>%
ggplot(aes(x = hosp_overall_rating, y = per_white_non_hisp, fill = as.factor(hosp_overall_rating))) +
geom_boxplot(show.legend = F) + facet_wrap(~hospital_type) +
ylab('White Pop %')

#------- Boxplot: Star Rating vs Population Below Poverty -------------
social.ind3 <- data %>% filter(!is.na(hosp_overall_rating)) %>%
ggplot(aes(x = hosp_overall_rating, y = per_below_poverty, fill = as.factor(hosp_overall_rating))) +
geom_boxplot(show.legend = F)  + facet_wrap(~hospital_type) +
ylab('Below Poverty %')

#------- Boxplot: Star Rating vs per_college_grad_deg_25_plus -------------
social.ind4 <- data %>% filter(!is.na(hosp_overall_rating)) %>%
ggplot(aes(x = hosp_overall_rating, y = per_college_grad_deg_25_plus, fill = as.factor(hosp_overall_rat
geom_boxplot(show.legend = F)  + facet_wrap(~hospital_type) +
ylab('College Grad Degree %')

#------- Boxplot: Star Rating vs med_inc_15plus_12mo -------------
social.ind5 <- data %>% filter(!is.na(hosp_overall_rating)) %>%
ggplot(aes(x = hosp_overall_rating, y = med_inc_15plus_12mo, fill = as.factor(hosp_overall_rating))) +
geom_boxplot(show.legend = F) + facet_wrap(~hospital_type) +
ylab('Median Income $')

#Five Graphs in One
grid.arrange(social.ind1, social.ind2, social.ind3, social.ind4, social.ind5, ncol=2)
```




```
grid.social.ind3 <- arrangeGrob(social.ind1, social.ind2,social.ind3, social.ind4, social.ind5, ncol=2)
```

```
##Mortality and Complication Measures by Overall Hospital Rating -- FINAL GROUP

require(gridExtra)

#21,22,23,36

plot1 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,21]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[21]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")


plot2 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,22]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[22]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")

plot3 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,23]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[23]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")
```

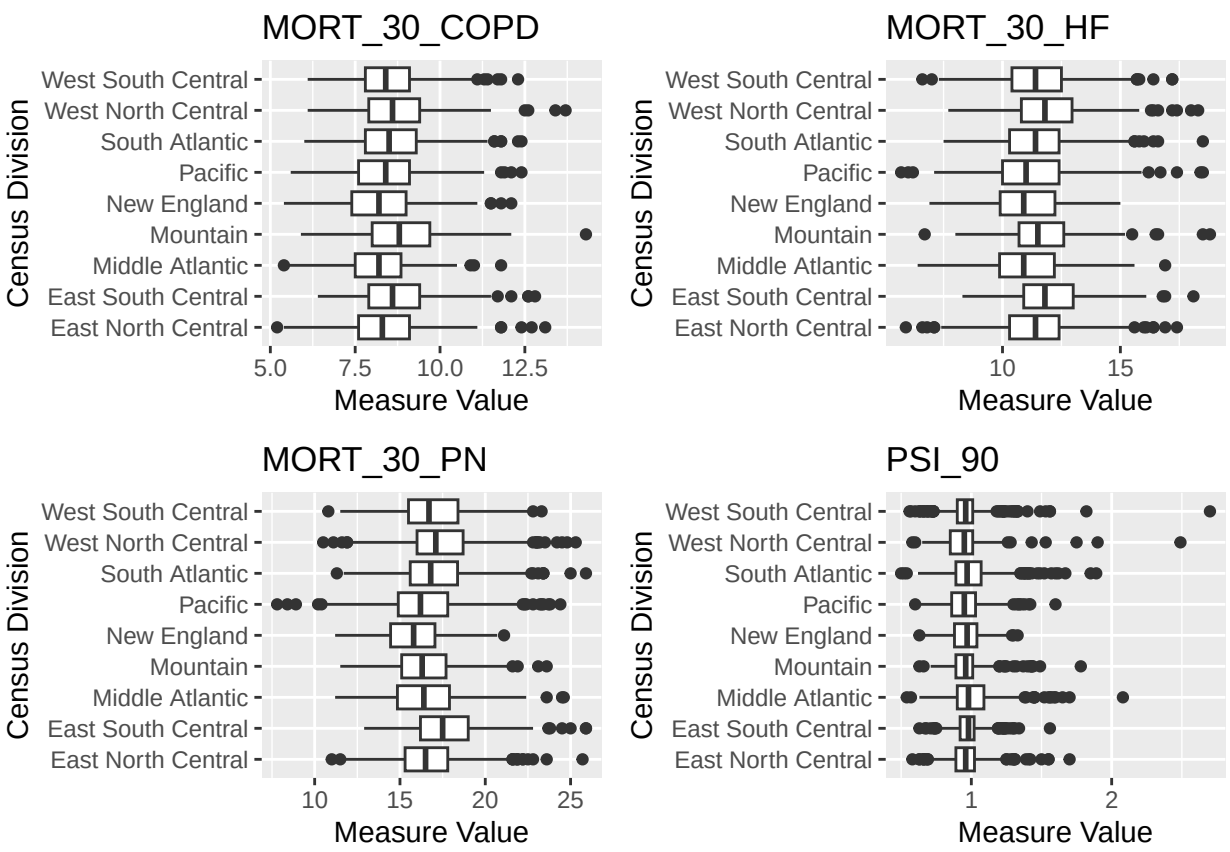
```
plot4 <- data %>% ggplot(aes(x=factor(as.factor(hosp_overall_rating)),y =data[,36]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[36]) +
  ylab('Measure Value') + xlab("Hosp_Overall_Rating")

#Aggregate into a single graph
grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```




```
#Save
g <- arrangeGrob(plot1, plot2, plot3, plot4, ncol=2) #generates g
ggsave("Top4_Star.png", g, width = 25, height = 20, units = "cm")
```

```
##Mortality and Complication Measures by Census Region -- FINAL GROUP

require(gridExtra)

#21,22,23,36

plot1 <- data  %>% ggplot(aes(x=census_division,y =data[,21]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[21]) +
```

```
  ylab('Measure Value') + xlab("Census Division")


plot2 <- data %>% ggplot(aes(x=census_division,y =data[,22]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[22]) +
  ylab('Measure Value') + xlab("Census Division")

plot3 <- data %>% ggplot(aes(x=census_division,y =data[,23]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[23]) +
  ylab('Measure Value') + xlab("Census Division")


plot4 <- data %>% ggplot(aes(x=census_division,y =data[,36]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[36]) +
  ylab('Measure Value') + xlab("Census Division")

#Aggregate into a single graph
grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```

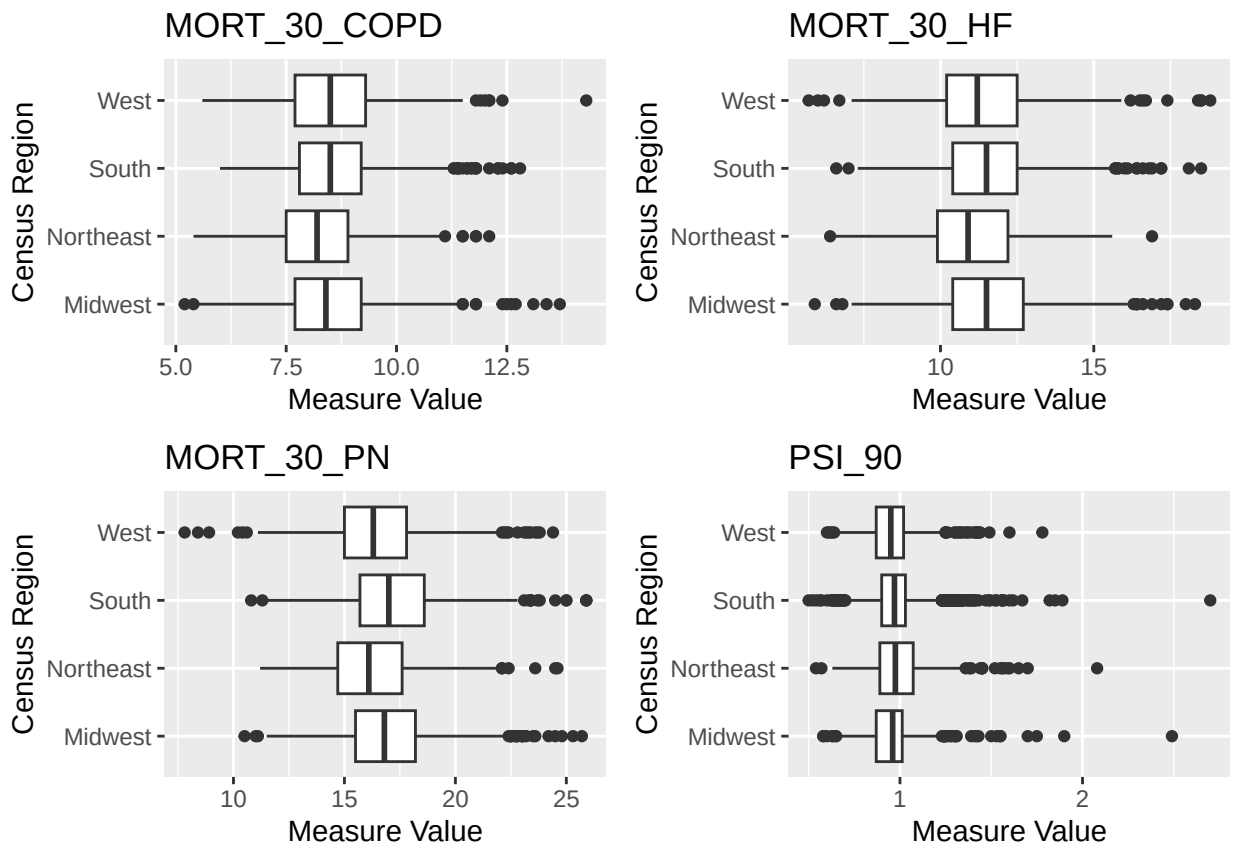


```
#Save
g <- arrangeGrob(plot1, plot2, plot3, plot4, ncol=2) #generates g
ggsave("Census_Division_Meas.png", g, width = 25, height = 20, units = "cm")
```

```
##Mortality and Complication Measures by Census Region -- FINAL GROUP

require(gridExtra)
require(forcats)
```

```
## Loading required package: forcats
```

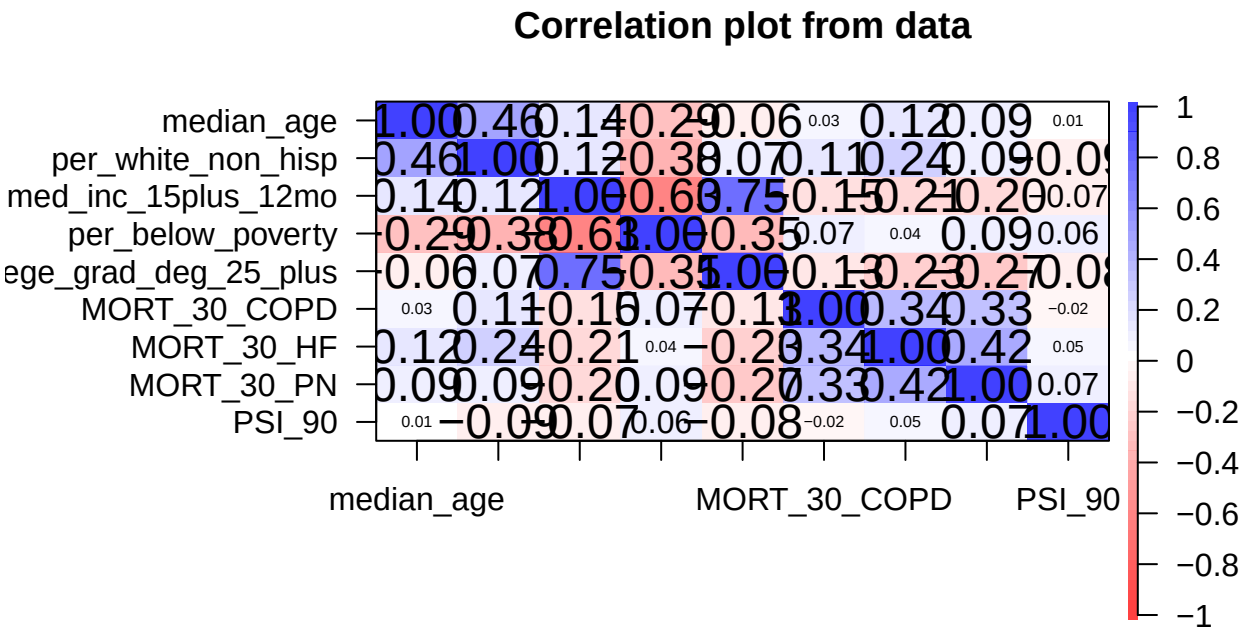
```
#21,22,23,36

plot1 <- data  %>% ggplot(aes(x=census_region,y =data[,21]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[21]) +
  ylab('Measure Value') + xlab("Census Region")


plot2 <- data %>% ggplot(aes(x=census_region,y =data[,22]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[22]) +
  ylab('Measure Value') + xlab("Census Region")

plot3 <- data %>% ggplot(aes(x=census_region,y =data[,23]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[23]) +
  ylab('Measure Value') + xlab("Census Region")


plot4 <- data %>% ggplot(aes(x=census_region,y =data[,36]))+
  geom_boxplot(show.legend = F) +
  coord_flip() +
  ggtitle(colnames(data)[36]) +
  ylab('Measure Value') + xlab("Census Region")

#Aggregate into a single graph
grid.arrange(plot1, plot2, plot3, plot4, ncol=2)
```




```
#Save
g <- arrangeGrob(plot1, plot2, plot3, plot4, ncol=2) #generates g
ggsave("Census_Region_Meas.png", g, width = 25, height = 20, units = "cm")
```

#Correlation Plot

```
#Correlation Plots for Key Numeric Variables

cor_data <- data %>% dplyr::select(13:17,21,22,23,36)

cor.plot(cor_data)
```



Linear regression - FOur Different Measures - 1 for each!

```
#Refresh on column names
colnames(data)
```

```
##  [1] "facility_ID"                   "facility_name"
##  [3] "address"                       "city"
##  [5] "state"                         "zip"
##  [7] "county"                        "hospital_type"
##  [9] "hospital_ownership"            "hosp_overall_rating"
## [11] "census_region"                 "census_division"
## [13] "median_age"                    "per_white_non_hisp"
## [15] "med_inc_15plus_12mo"           "per_below_poverty"
## [17] "per_college_grad_deg_25_plus" "COMP_HIP_KNEE"
## [19] "MORT_30_AMI"                   "MORT_30_CABG"
## [21] "MORT_30_COPD"                  "MORT_30_HF"
## [23] "MORT_30_PN"                    "MORT_30_STK"
## [25] "PSI_03"                        "PSI_04"
## [27] "PSI_06"                        "PSI_08"
## [29] "PSI_09"                        "PSI_10"
## [31] "PSI_11"                        "PSI_12"
## [33] "PSI_13"                        "PSI_14"
## [35] "PSI_15"                        "PSI_90"
```

#Regression on the Composite measure for complications

```
#Data cleaning -> Quality Measure 90 - > Composite Score for Other P Measures

mod_data <- data %>%
   dplyr::select(4:5,8:17,36) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter(PSI_90 != 'NA')  #Remove NAs from measure

# treat overall rating as a factor
mod_data$hosp_overall_rating <- as.factor(mod_data$hosp_overall_rating)

table(mod_data$hospital_type)
```

```
##
##      Acute Care Hospitals Critical Access Hospitals
##                      2588                         0
```

```
#Take a full model with all numeric terms

mod_mod <- lm(PSI_90~median_age+per_white_non_hisp+med_inc_15plus_12mo+
                per_below_poverty+per_college_grad_deg_25_plus+census_region,mod_data)

summary(mod_mod)
```

```
##
## Call:
## lm(formula = PSI_90 ~ median_age + per_white_non_hisp + med_inc_15plus_12mo +
##     per_below_poverty + per_college_grad_deg_25_plus + census_region,
##     data = mod_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47432 -0.09114 -0.01345  0.06641  1.71062
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   9.996e-01  2.843e-02  35.154  < 2e-16 ***
## median_age                    9.448e-04  5.603e-04   1.686  0.09185 .
## per_white_non_hisp           -6.549e-04  1.593e-04  -4.110 4.07e-05 ***
## med_inc_15plus_12mo          -3.476e-07  5.455e-07  -0.637  0.52403
## per_below_poverty             9.452e-05  5.177e-04   0.183  0.85514
## per_college_grad_deg_25_plus -4.943e-04  3.038e-04  -1.627  0.10383
## census_regionNortheast        3.315e-02  1.049e-02   3.162  0.00159 **
## census_regionSouth            1.127e-02  8.563e-03   1.316  0.18819
## census_regionWest            -7.662e-03  1.025e-02  -0.747  0.45493
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1617 on 2579 degrees of freedom
## Multiple R-squared:  0.02227,    Adjusted R-squared:  0.01924
## F-statistic: 7.343 on 8 and 2579 DF,  p-value: 1.071e-09
```

```r
#conduct a step AIC analysis to prioritize functions

new <- stepAIC(mod_mod)
```

```
## Start:  AIC=-9422.77
## PSI_90 ~ median_age + per_white_non_hisp + med_inc_15plus_12mo +
##     per_below_poverty + per_college_grad_deg_25_plus + census_region
##
##                                Df Sum of Sq    RSS     AIC
## - per_below_poverty             1   0.00087 67.407 -9424.7
## - med_inc_15plus_12mo           1   0.01061 67.417 -9424.4
## <none>                                      67.406 -9422.8
## - per_college_grad_deg_25_plus  1   0.06920 67.475 -9422.1
## - median_age                    1   0.07433 67.481 -9421.9
## - census_region                 3   0.42780 67.834 -9412.4
## - per_white_non_hisp            1   0.44160 67.848 -9407.9
##
## Step:  AIC=-9424.74
## PSI_90 ~ median_age + per_white_non_hisp + med_inc_15plus_12mo +
##     per_college_grad_deg_25_plus + census_region
##
##                                Df Sum of Sq    RSS     AIC
## - med_inc_15plus_12mo           1   0.02143 67.429 -9425.9
## <none>                                      67.407 -9424.7
## - per_college_grad_deg_25_plus  1   0.06964 67.477 -9424.1
## - median_age                    1   0.07363 67.481 -9423.9
## - census_region                 3   0.43322 67.840 -9414.2
## - per_white_non_hisp            1   0.50293 67.910 -9407.5
##
## Step:  AIC=-9425.92
## PSI_90 ~ median_age + per_white_non_hisp + per_college_grad_deg_25_plus +
##     census_region
##
##                                Df Sum of Sq    RSS     AIC
## <none>                                      67.429 -9425.9
## - median_age                    1   0.05423 67.483 -9425.8
## - per_college_grad_deg_25_plus  1   0.34556 67.774 -9414.7
## - census_region                 3   0.45203 67.881 -9414.6
## - per_white_non_hisp            1   0.48910 67.918 -9409.2
```

```r
#Use model selected by stepAIC function
mod_modf <- lm(PSI_90~median_age+per_white_non_hisp+per_college_grad_deg_25_plus+census_region,mod_data)

summary(mod_modf)
```

```
##
## Call:
## lm(formula = PSI_90 ~ median_age + per_white_non_hisp + per_college_grad_deg_25_plus +
##     census_region, data = mod_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47169 -0.09161 -0.01371  0.06637  1.71100
```

```
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.0035572  0.0201266  49.862  < 2e-16 ***
## median_age                    0.0007431  0.0005157   1.441 0.149760
## per_white_non_hisp           -0.0006525  0.0001508  -4.327 1.57e-05 ***
## per_college_grad_deg_25_plus -0.0006854  0.0001885  -3.637 0.000281 ***
## census_regionNortheast        0.0332783  0.0104823   3.175 0.001518 **
## census_regionSouth            0.0118594  0.0085364   1.389 0.164869
## census_regionWest            -0.0084123  0.0101868  -0.826 0.408996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1616 on 2581 degrees of freedom
## Multiple R-squared:  0.02195,    Adjusted R-squared:  0.01967
## F-statistic: 9.653 on 6 and 2581 DF,  p-value: 1.572e-10
```

```
#Median age not significant so remove it
mod_modff <- lm(PSI_90~per_white_non_hisp+per_college_grad_deg_25_plus+census_region,mod_data)

summary(mod_modff)
```

```
##
## Call:
## lm(formula = PSI_90 ~ per_white_non_hisp + per_college_grad_deg_25_plus +
##     census_region, data = mod_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46422 -0.09109 -0.01357  0.06654  1.70835
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.0259602  0.0127820  80.266  < 2e-16 ***
## per_white_non_hisp           -0.0005653  0.0001381  -4.092 4.41e-05 ***
## per_college_grad_deg_25_plus -0.0007032  0.0001881  -3.739 0.000189 ***
## census_regionNortheast        0.0352728  0.0103927   3.394 0.000699 ***
## census_regionSouth            0.0136722  0.0084450   1.619 0.105573
## census_regionWest            -0.0068090  0.0101280  -0.672 0.501453
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1617 on 2582 degrees of freedom
## Multiple R-squared:  0.02116,    Adjusted R-squared:  0.01927
## F-statistic: 11.16 on 5 and 2582 DF,  p-value: 1.156e-10
```

```
#VIF for PSI_90
vif(mod_modff)
```

```
##                                 GVIF Df GVIF^(1/(2*Df))
## per_white_non_hisp           1.145756  1        1.070400
## per_college_grad_deg_25_plus 1.056643  1        1.027932
## census_region                1.138664  3        1.021879
```

Regression for Heart Failure

```
#Data cleaning -> MORT_30_HF - > 30 Day Heart Failure Death Rate

hf_data <- data %>%
  dplyr::select(4:5,8:17,22) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter(MORT_30_HF != 'NA')  #Remove NAs from measure

# treat overall rating as a factor
hf_data$hosp_overall_rating <- as.factor(hf_data$hosp_overall_rating)


#Data cleaning -> MORT_30_HF - > 30 Day Heart Failure Death Rate
detach(package:MASS, unload = TRUE)
```

```
## Warning: 'MASS' namespace cannot be unloaded:
##   namespace 'MASS' is imported by 'ipred', 'TH.data' so cannot be unloaded
```

```
hf_data <- data %>% select(11,13:17,22)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:olsrr':
##
##     cement

## The following object is masked from 'package:dplyr':
##
##     select

## The following object is masked from 'package:DAAG':
##
##     hills
```

```
hf_data <- hf_data[complete.cases(hf_data),] #Only complete cases


multi.model.hf <- lm(MORT_30_HF ~ census_region + per_white_non_hisp +
                       med_inc_15plus_12mo + per_college_grad_deg_25_plus, data = hf_data)

summary(multi.model.hf)
```

```
##
## Call:
## lm(formula = MORT_30_HF ~ census_region + per_white_non_hisp +
##     med_inc_15plus_12mo + per_college_grad_deg_25_plus, data = hf_data)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9876 -1.0865 -0.0969  1.0285  6.5660
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.097e+01  1.380e-01  79.452  < 2e-16 ***
## census_regionNortheast       -1.364e-01  9.666e-02  -1.411    0.158
## census_regionSouth            3.369e-01  7.831e-02   4.302 1.74e-05 ***
## census_regionWest             4.385e-01  9.374e-02   4.677 3.04e-06 ***
## per_white_non_hisp            2.173e-02  1.307e-03  16.629  < 2e-16 ***
## med_inc_15plus_12mo          -1.924e-05  4.043e-06  -4.758 2.05e-06 ***
## per_college_grad_deg_25_plus -1.711e-02  2.657e-03  -6.437 1.41e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.623 on 2962 degrees of freedom
## Multiple R-squared:  0.142,  Adjusted R-squared:  0.1402
## F-statistic: 81.68 on 6 and 2962 DF,  p-value: < 2.2e-16
```

```
vif(multi.model.hf)
```

```
##                                  GVIF Df GVIF^(1/(2*Df))
## census_region                1.191546  3        1.029639
## per_white_non_hisp           1.164497  1        1.079119
## med_inc_15plus_12mo          2.253382  1        1.501127
## per_college_grad_deg_25_plus 2.223385  1        1.491102
```

Regression for Pneumonia

```
# regression analysis  for  Mort_30_PN

pn_data <- data %>%
   dplyr::select(4:5,8:17,23) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter( !is.na(MORT_30_PN))  %>%  #Remove NAs from measure
  filter(census_region != '#N/A') %>%
  filter(!is.na(per_college_grad_deg_25_plus))  %>%
  filter (!is.na(per_below_poverty))  %>%
  filter(!is.na(med_inc_15plus_12mo))  %>%
  filter (!is.na(per_white_non_hisp))

# treat overall rating as a factor
pn_data$hosp_overall_rating <- as.factor(pn_data$hosp_overall_rating)

set.seed(1234)

Mod <- lm (MORT_30_PN ~ census_region +per_college_grad_deg_25_plus + per_below_poverty + med_inc_15plus

summary(Mod)
```

```
##
```

```
## Call:
## lm(formula = MORT_30_PN ~ census_region + per_college_grad_deg_25_plus +
##     per_below_poverty + med_inc_15plus_12mo + per_white_non_hisp,
##     data = pn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8237 -1.4690 -0.1637  1.2996  8.6013
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.638e+01  3.158e-01  51.870  < 2e-16 ***
## census_regionNortheast       -1.082e-01  1.326e-01  -0.816   0.4145
## census_regionSouth            6.445e-01  1.079e-01   5.975 2.58e-09 ***
## census_regionWest            -7.078e-03  1.291e-01  -0.055   0.9563
## per_college_grad_deg_25_plus -3.765e-02  3.746e-03 -10.052  < 2e-16 ***
## per_below_poverty             1.524e-02  6.865e-03   2.220   0.0265 *
## med_inc_15plus_12mo           8.722e-06  6.933e-06   1.258   0.2085
## per_white_non_hisp            1.195e-02  1.915e-03   6.242 4.95e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.195 on 2883 degrees of freedom
## Multiple R-squared:  0.1009, Adjusted R-squared:  0.09875
## F-statistic: 46.24 on 7 and 2883 DF,  p-value: < 2.2e-16
```

```
#lm(formula = MORT_30_PN ~ per_white_non_hisp + census_region +  per_college_grad_deg_25_plus, data = p
```

```
#Step AIC for model
modaic <- stepAIC(Mod)
```

```
## Start:  AIC=4554.61
## MORT_30_PN ~ census_region + per_college_grad_deg_25_plus + per_below_poverty +
##     med_inc_15plus_12mo + per_white_non_hisp
##
##                                Df Sum of Sq   RSS    AIC
## - med_inc_15plus_12mo           1      7.63 13902 4554.2
## <none>                                      13895 4554.6
## - per_below_poverty             1     23.75 13918 4557.6
## - per_white_non_hisp            1    187.80 14083 4591.4
## - census_region                 3    300.68 14196 4610.5
## - per_college_grad_deg_25_plus  1    486.96 14382 4652.2
##
## Step:  AIC=4554.2
## MORT_30_PN ~ census_region + per_college_grad_deg_25_plus + per_below_poverty +
##     per_white_non_hisp
##
##                                Df Sum of Sq   RSS    AIC
## <none>                                      13902 4554.2
## - per_below_poverty             1     16.20 13919 4555.6
## - per_white_non_hisp            1    180.54 14083 4589.5
## - census_region                 3    297.37 14200 4609.4
## - per_college_grad_deg_25_plus  1    833.45 14736 4720.5
```

```r
# based on  Lowest step AIC model

modf <- lm(MORT_30_PN ~ census_region + per_college_grad_deg_25_plus + per_below_poverty +
    per_white_non_hisp, pn_data)
summary(modf)
```

```
##
## Call:
## lm(formula = MORT_30_PN ~ census_region + per_college_grad_deg_25_plus +
##     per_below_poverty + per_white_non_hisp, data = pn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8203 -1.4721 -0.1611  1.3005  8.6345
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  16.658550   0.226972  73.395  < 2e-16 ***
## census_regionNortheast       -0.106889   0.132634  -0.806   0.4204
## census_regionSouth            0.640431   0.107830   5.939 3.21e-09 ***
## census_regionWest            -0.007886   0.129121  -0.061   0.9513
## per_college_grad_deg_25_plus -0.034266   0.002606 -13.149  < 2e-16 ***
## per_below_poverty             0.009954   0.005430   1.833   0.0669 .
## per_white_non_hisp            0.011572   0.001891   6.120 1.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.196 on 2884 degrees of freedom
## Multiple R-squared:  0.1004, Adjusted R-squared:  0.09857
## F-statistic: 53.67 on 6 and 2884 DF,  p-value: < 2.2e-16
```

```r
#After doing the summary of  suggestive setpAic model  per_below_poverty is not significant
# at 95+ CI, Remove it.

modf.1 <- lm(MORT_30_PN ~ census_region + per_college_grad_deg_25_plus  + per_white_non_hisp, pn_data)
summary(modf.1)
```

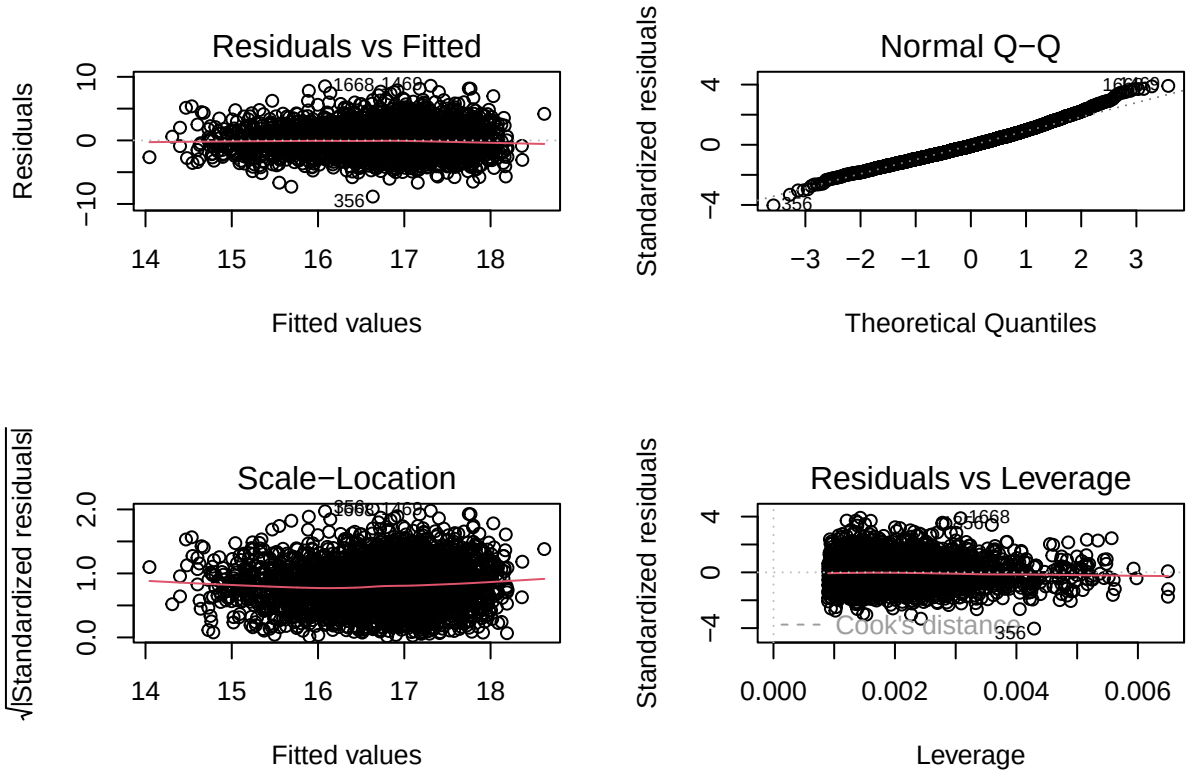
```
##
## Call:
## lm(formula = MORT_30_PN ~ census_region + per_college_grad_deg_25_plus +
##     per_white_non_hisp, data = pn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8347 -1.4860 -0.1613  1.3141  8.5936
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  16.934750   0.169817  99.723  < 2e-16 ***
## census_regionNortheast       -0.109480   0.132681  -0.825    0.409
## census_regionSouth            0.647393   0.107807   6.005 2.15e-09 ***
## census_regionWest            -0.032930   0.128449  -0.256    0.798
## per_college_grad_deg_25_plus -0.035783   0.002472 -14.475  < 2e-16 ***
```

```
## per_white_non_hisp            0.010405   0.001781   5.841 5.75e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.196 on 2885 degrees of freedom
## Multiple R-squared:  0.09939,    Adjusted R-squared:  0.09783
## F-statistic: 63.68 on 5 and 2885 DF,  p-value: < 2.2e-16
```

```
vif(modf.1)
```

```
##                                 GVIF Df GVIF^(1/(2*Df))
## census_region                1.161156  3        1.025215
## per_college_grad_deg_25_plus 1.040165  1        1.019885
## per_white_non_hisp           1.142433  1        1.068846
```

```
# Analysis plots
par(mfrow=c(2,2))
plot(modf.1)
```



Regression for COPD

```
#Data cleaning -> MORT_30_COPD - > 30 Day COPD death rate
```

```r
copd_data <- data %>%
  dplyr::select(4:5,8:17,21) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter(MORT_30_COPD != 'NA')  #Remove NAs from measure

# treat overall rating as a factor
copd_data$hosp_overall_rating <- as.factor(copd_data$hosp_overall_rating)

#Data cleaning -> MORT_30_COPD - > 30 Day Heart Failure Death Rate
detach(package:MASS, unload = TRUE)
```

```
## Warning: 'MASS' namespace cannot be unloaded:
##   namespace 'MASS' is imported by 'ipred', 'TH.data' so cannot be unloaded
```

```r
copd_data <- data %>% select(11,13:17,21)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:olsrr':
##
##     cement

## The following object is masked from 'package:dplyr':
##
##     select

## The following object is masked from 'package:DAAG':
##
##     hills
```

```r
copd_data <- copd_data[complete.cases(copd_data),] #Only complete cases

multi.model.copd <- lm(MORT_30_COPD ~ census_region + per_white_non_hisp +
    med_inc_15plus_12mo, data = copd_data)
summary(multi.model.copd)
```

```
##
## Call:
## lm(formula = MORT_30_COPD ~ census_region + per_white_non_hisp +
##     med_inc_15plus_12mo, data = copd_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1323 -0.7556 -0.0923  0.6844  5.5281
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              8.388e+00  9.791e-02  85.674  < 2e-16 ***
## census_regionNortheast -1.884e-01  6.836e-02  -2.757 0.005881 **
```

```
## census_regionSouth      1.940e-01  5.535e-02   3.505 0.000464 ***
## census_regionWest       3.245e-01  6.843e-02   4.743 2.21e-06 ***
## per_white_non_hisp      7.787e-03  9.332e-04   8.345  < 2e-16 ***
## med_inc_15plus_12mo    -1.620e-05  1.964e-06  -8.249 2.47e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.109 on 2718 degrees of freedom
## Multiple R-squared:  0.05855,    Adjusted R-squared:  0.05681
## F-statistic: 33.81 on 5 and 2718 DF,  p-value: < 2.2e-16
```

```
vif(multi.model.copd)
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## census_region       1.162934  3        1.025477
## per_white_non_hisp  1.135241  1        1.065477
## med_inc_15plus_12mo 1.054808  1        1.027038
```

```
#Calculate sample counts for each model
nrow(mod_data)
```

```
## [1] 2588
```

```
nrow(copd_data)
```

```
## [1] 2724
```

```
nrow(pn_data)
```

```
## [1] 2891
```

```
nrow(hf_data)
```

```
## [1] 2969
```

Logistic Regression

```
##---------Actual Code begins to create logistic model -------------

hf_data <- data %>%
  dplyr::select(10:11,13:36) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter(MORT_30_HF != 'NA')   %>%  #Remove NAs from measure
  filter(census_region != '#N/A')

hf_data <- hf_data[complete.cases(hf_data),] #Only complete cases
```

```r
hf_data_log <- hf_data

#hf_data_log$hosp_overall_rating <- as.numeric(hf_data_log$hosp_overall_rating)
hf_data_log[hf_data_log$hosp_overall_rating < 4,]$hosp_overall_rating <- 0
hf_data_log[hf_data_log$hosp_overall_rating >= 4,]$hosp_overall_rating <- 1
hf_data_log$hosp_overall_rating <- as.factor(hf_data_log$hosp_overall_rating)

#table(hf_data_log$hosp_overall_rating)

  # Separate into test/train
  set.seed(1234)
  #data partitioning test/train
  ind <- sample(2, nrow(hf_data_log), replace = T, prob=c(0.6,0.4))
  hf_train_log <- hf_data_log[ind == 1,]
  hf_test_log <- hf_data_log[ind ==2, ]

#  model_train_log <- glm(hosp_overall_rating ~ census_region + median_age +
#    per_white_non_hisp + per_below_poverty + COMP_HIP_KNEE +
#    MORT_30_AMI + MORT_30_CABG + MORT_30_COPD + MORT_30_PN +
#    MORT_30_STK + PSI_03 + PSI_04 + PSI_06 + PSI_08 + PSI_13, data = hf_train_log, family #= 'binomial

  model_train_log <- glm(hosp_overall_rating ~ census_region + median_age +
    per_white_non_hisp + per_below_poverty + COMP_HIP_KNEE +
    MORT_30_CABG + MORT_30_PN +
    MORT_30_STK + PSI_03 + PSI_04, data = hf_train_log, family = 'binomial' )

    summary(model_train_log)
```

```
##
## Call:
## glm(formula = hosp_overall_rating ~ census_region + median_age +
##     per_white_non_hisp + per_below_poverty + COMP_HIP_KNEE +
##     MORT_30_CABG + MORT_30_PN + MORT_30_STK + PSI_03 + PSI_04,
##     family = "binomial", data = hf_train_log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0498  -0.7953  -0.2923   0.7920   2.7261
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            17.433397   2.110213   8.261  < 2e-16 ***
## census_regionNortheast -1.286837   0.458665  -2.806 0.005022 **
## census_regionSouth     -0.705203   0.316991  -2.225 0.026103 *
## census_regionWest      -0.557869   0.356979  -1.563 0.118111
## median_age             -0.060143   0.017785  -3.382 0.000721 ***
## per_white_non_hisp      0.021772   0.006516   3.341 0.000834 ***
## per_below_poverty      -0.046076   0.013822  -3.333 0.000858 ***
## COMP_HIP_KNEE          -1.300612   0.275225  -4.726 2.29e-06 ***
## MORT_30_CABG           -0.684959   0.189931  -3.606 0.000311 ***
## MORT_30_PN             -0.267691   0.066785  -4.008 6.12e-05 ***
## MORT_30_STK            -0.164051   0.075076  -2.185 0.028879 *
## PSI_03                 -0.518775   0.211956  -2.448 0.014383 *
```

```
## PSI_04                -0.026320   0.007768  -3.388 0.000703 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 587.10  on 434  degrees of freedom
## Residual deviance: 412.88  on 422  degrees of freedom
## AIC: 438.88
##
## Number of Fisher Scoring iterations: 5
```

```
  (full_model <- model_train_log %>% stepAIC(trace = FALSE))
```

```
##
## Call:  glm(formula = hosp_overall_rating ~ census_region + median_age +
##     per_white_non_hisp + per_below_poverty + COMP_HIP_KNEE +
##     MORT_30_CABG + MORT_30_PN + MORT_30_STK + PSI_03 + PSI_04,
##     family = "binomial", data = hf_train_log)
##
## Coefficients:
##            (Intercept)  census_regionNortheast      census_regionSouth
##               17.43340               -1.28684               -0.70520
##      census_regionWest             median_age      per_white_non_hisp
##               -0.55787               -0.06014                0.02177
##      per_below_poverty          COMP_HIP_KNEE           MORT_30_CABG
##               -0.04608               -1.30061               -0.68496
##             MORT_30_PN            MORT_30_STK                PSI_03
##               -0.26769               -0.16405               -0.51877
##                 PSI_04
##               -0.02632
##
## Degrees of Freedom: 434 Total (i.e. Null);  422 Residual
## Null Deviance:       587.1
## Residual Deviance: 412.9     AIC: 438.9
```

```
#CM - train data
p_log_train <- predict(model_train_log, hf_train_log, type = 'response')
pred_log_train <- ifelse(p_log_train > 0.5, 1, 0)
confusionMatrix(factor(pred_log_train), factor(hf_train_log$hosp_overall_rating), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 212  50
##          1  47 126
##
##                Accuracy : 0.777
##                  95% CI : (0.7349, 0.8153)
##     No Information Rate : 0.5954
##     P-Value [Acc > NIR] : 8.236e-16
##
```

```
##                  Kappa : 0.5359
##
##  Mcnemar's Test P-Value : 0.8391
##
##             Sensitivity : 0.7159
##             Specificity : 0.8185
##          Pos Pred Value : 0.7283
##          Neg Pred Value : 0.8092
##              Prevalence : 0.4046
##          Detection Rate : 0.2897
##    Detection Prevalence : 0.3977
##       Balanced Accuracy : 0.7672
##
##        'Positive' Class : 1
##
```

```
#CM - test data
p_log_test <- predict(model_train_log, hf_test_log, type = 'response')
pred_log_test <- ifelse(p_log_test > 0.5, 1, 0)
confusionMatrix(factor(pred_log_test), factor(hf_test_log$hosp_overall_rating), positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 142  40
##          1  34  86
##
##                Accuracy : 0.755
##                  95% CI : (0.7024, 0.8024)
##     No Information Rate : 0.5828
##     P-Value [Acc > NIR] : 2.849e-10
##
##                   Kappa : 0.4927
##
##  Mcnemar's Test P-Value : 0.5611
##
##             Sensitivity : 0.6825
##             Specificity : 0.8068
##          Pos Pred Value : 0.7167
##          Neg Pred Value : 0.7802
##              Prevalence : 0.4172
##          Detection Rate : 0.2848
##    Detection Prevalence : 0.3974
##       Balanced Accuracy : 0.7447
##
##        'Positive' Class : 1
##
```

```
#ROC Curve
p_log_train <- predict(model_train_log, hf_train_log, type = 'response')
r_train <- multiclass.roc(hf_train_log$hosp_overall_rating, p_log_train, percent = TRUE)
```

```
## Setting direction: controls < cases
```

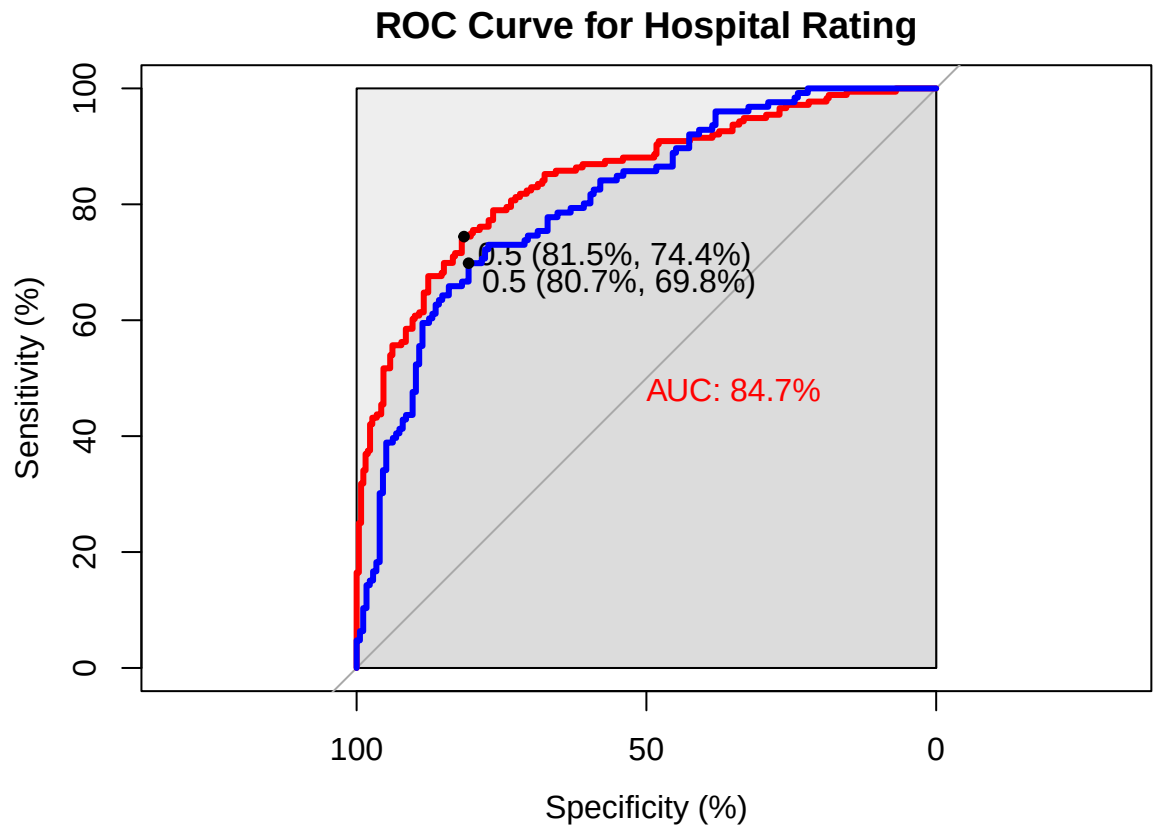
```
roc_train <- r_train[['rocs']]
r1_train <- roc_train[[1]]
```

```
p_log_test <- predict(model_train_log, hf_test_log, type = 'response')
r_test <- multiclass.roc(hf_test_log$hosp_overall_rating, p_log_test, percent = TRUE)
```

```
## Setting direction: controls < cases
```

```
roc_test <- r_test[['rocs']]
r1_test <- roc_test[[1]]
```

```
plot.roc(r1_train,col= "red", lwd = 3,
         print.auc = T,
         auc.polygon= T,
         max.auc.polygon = T,
         print.thres = T,
         main = "ROC Curve for Hospital Rating")
plot(r1_test, add = T, col = "blue",
     lwd = 3, print.thres = T)
```




```
(coords(r1_train, "best", ret="threshold", transpose = FALSE))
```

```
##   threshold
## 1 0.4771103
```

```
(coords(r1_test, "best", ret="threshold", transpose = FALSE))
```

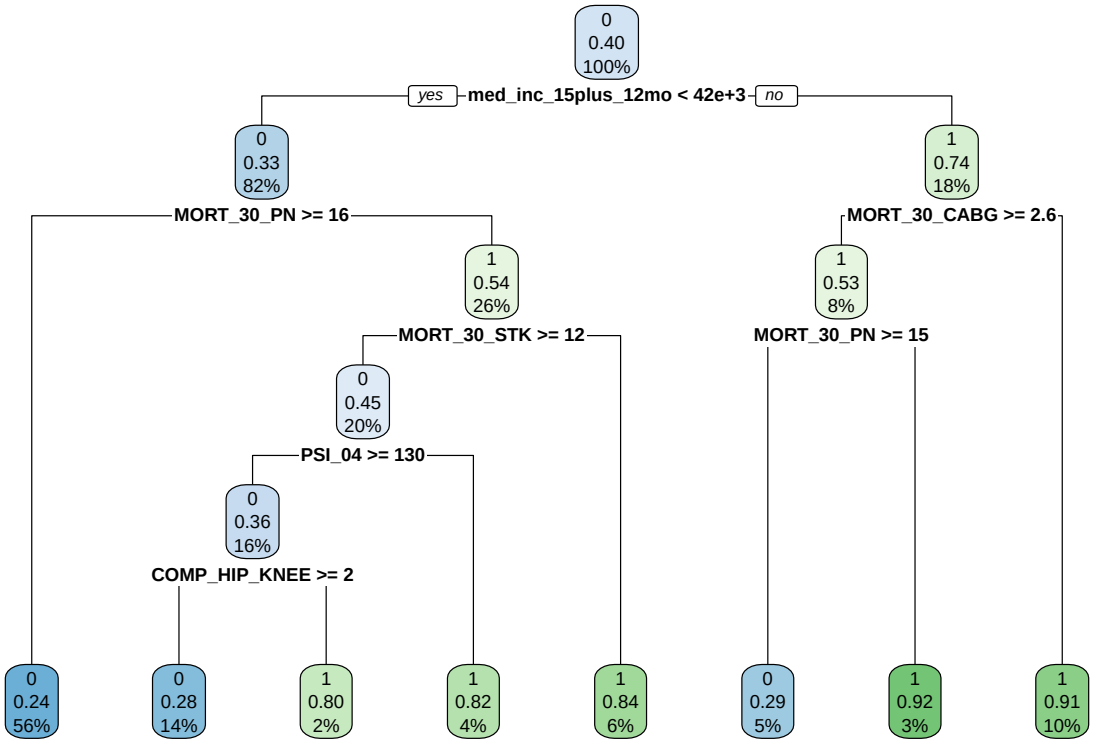
```
##   threshold
## 1 0.4783859
```

Decision Tree Analysis

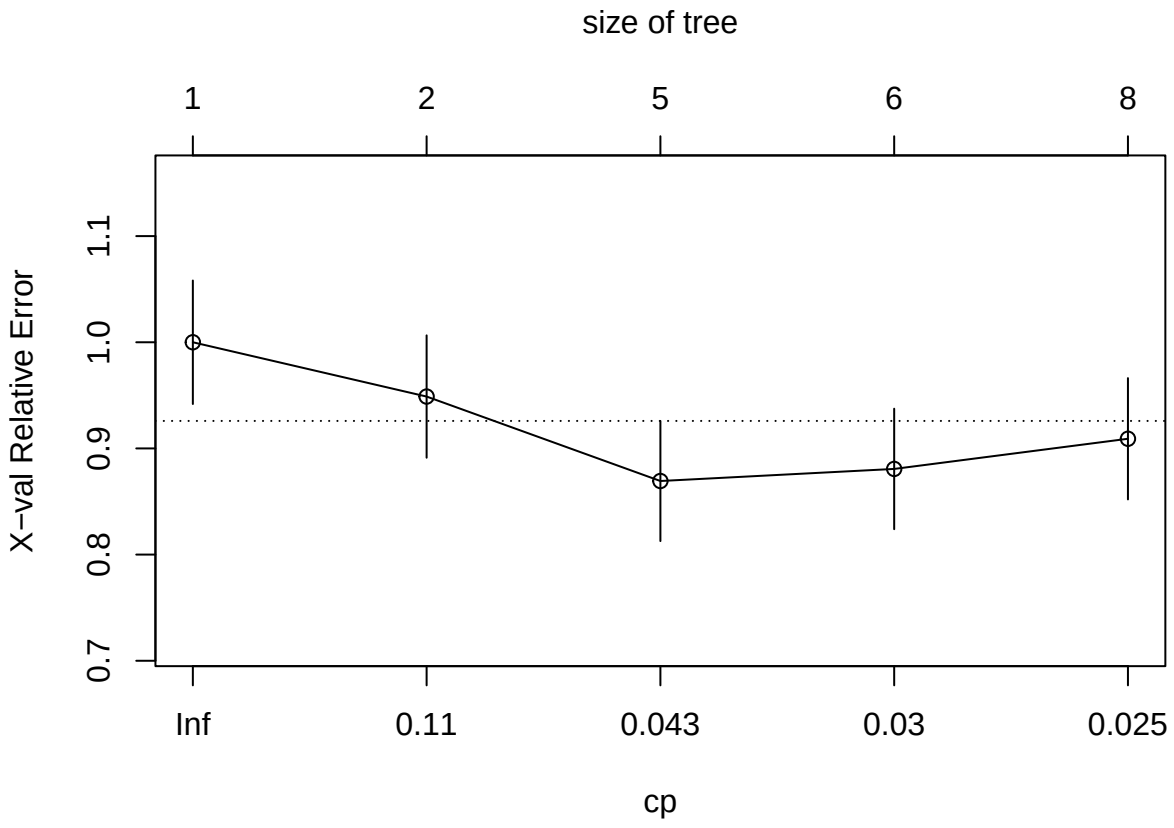
```
## ---------------------- This section is to split the data into train and test --------------

hf_data <- data %>%
  dplyr::select(10:11,13:36) %>% #Narrow to features to use in the model (plus one quality measure)
  filter(hosp_overall_rating != 'NA') %>% #remove NAs
  filter(MORT_30_HF != 'NA')   %>%  #Remove NAs from measure
  filter(census_region != '#N/A')

hf_data <- hf_data[complete.cases(hf_data),] #Only complete cases


hf_data_log <- hf_data

#hf_data_log$hosp_overall_rating <- as.numeric(hf_data_log$hosp_overall_rating)
hf_data_log[hf_data_log$hosp_overall_rating < 4,]$hosp_overall_rating <- 0
hf_data_log[hf_data_log$hosp_overall_rating >= 4,]$hosp_overall_rating <- 1
hf_data_log$hosp_overall_rating <- as.factor(hf_data_log$hosp_overall_rating)

#table(hf_data_log$hosp_overall_rating)

  # Separate into test/train
  set.seed(1234)
  #data partitioning test/train
  ind <- sample(2, nrow(hf_data_log), replace = T, prob=c(0.6,0.4))
  train_tree <- hf_data_log[ind == 1,]
  test_tree <- hf_data_log[ind ==2, ]
```

Single Tree Analysis

```
### Single Tree
tree <- rpart(hosp_overall_rating ~., data = train_tree, cp=0.024)
rpart.plot(tree)
```


0
0.40
100%
yes
med_inc_15plus_12mo < 42e+3
no
0
0.33
82%
1
0.74
18%
MORT_30_PN >= 16
MORT_30_CABG >= 2.6
1
0.54
26%
1
0.53
8%
MORT_30_STK >= 12
MORT_30_PN >= 15
0
0.45
20%
PSI_04 >= 130
0
0.36
16%
COMP_HIP_KNEE >= 2
0
0.24
56%
0
0.28
14%
1
0.80
2%
1
0.82
4%
1
0.84
6%
0
0.29
5%
1
0.92
3%
1
0.91
10%

```
plotcp(tree)
```




```
# Confusion matrix -train
p <- predict(tree, train_tree, type = 'class')
confusionMatrix(p, train_tree$hosp_overall_rating, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 245  81
##          1  14  95
##
##                Accuracy : 0.7816
##                  95% CI : (0.7398, 0.8196)
##     No Information Rate : 0.5954
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5173
##
##  Mcnemar's Test P-Value : 1.275e-11
##
##             Sensitivity : 0.5398
##             Specificity : 0.9459
##          Pos Pred Value : 0.8716
##          Neg Pred Value : 0.7515
##              Prevalence : 0.4046
##          Detection Rate : 0.2184
```

```
##    Detection Prevalence : 0.2506
##       Balanced Accuracy : 0.7429
##
##        'Positive' Class : 1
##
```

```
# Confusion matrix -test
p <- predict(tree, test_tree, type = 'class')
confusionMatrix(p, test_tree$hosp_overall_rating, positive = '1')
```

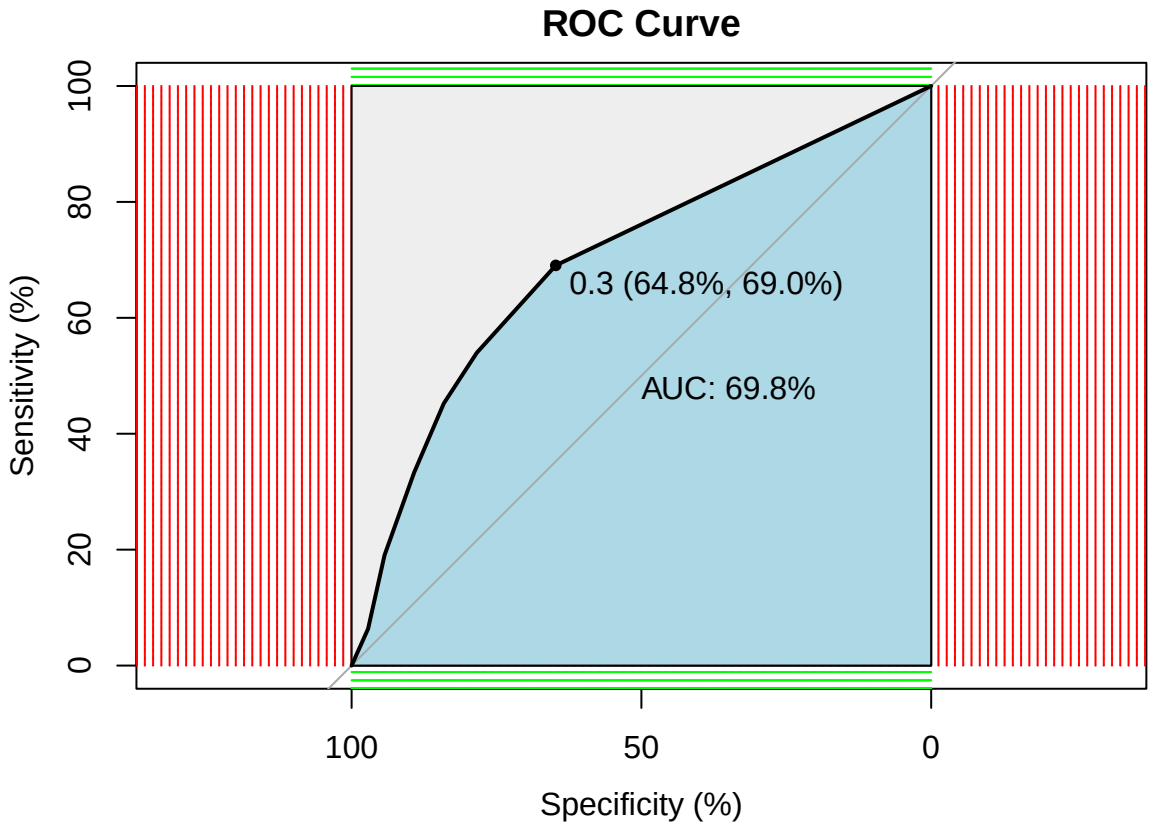
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 148  69
##          1  28  57
##
##                Accuracy : 0.6788
##                  95% CI : (0.6229, 0.7311)
##     No Information Rate : 0.5828
##     P-Value [Acc > NIR] : 0.0003768
##
##                   Kappa : 0.3075
##
##  Mcnemar's Test P-Value : 4.878e-05
##
##             Sensitivity : 0.4524
##             Specificity : 0.8409
##          Pos Pred Value : 0.6706
##          Neg Pred Value : 0.6820
##              Prevalence : 0.4172
##          Detection Rate : 0.1887
##    Detection Prevalence : 0.2815
##       Balanced Accuracy : 0.6466
##
##        'Positive' Class : 1
##
```

```
## ROC Curves
p1 <- predict(tree, test_tree, type = 'prob')
p1 <- p1[,2]
r <- multiclass.roc(test_tree$hosp_overall_rating, p1, percent = TRUE)
```

```
## Setting direction: controls < cases
```

```
roc <- r[['rocs']]
r1 <- roc[[1]]
plot.roc(r1,
         print.auc=TRUE,
         auc.polygon=TRUE,
         grid=c(0.1, 0.2),
         grid.col=c("green", "red"),
```

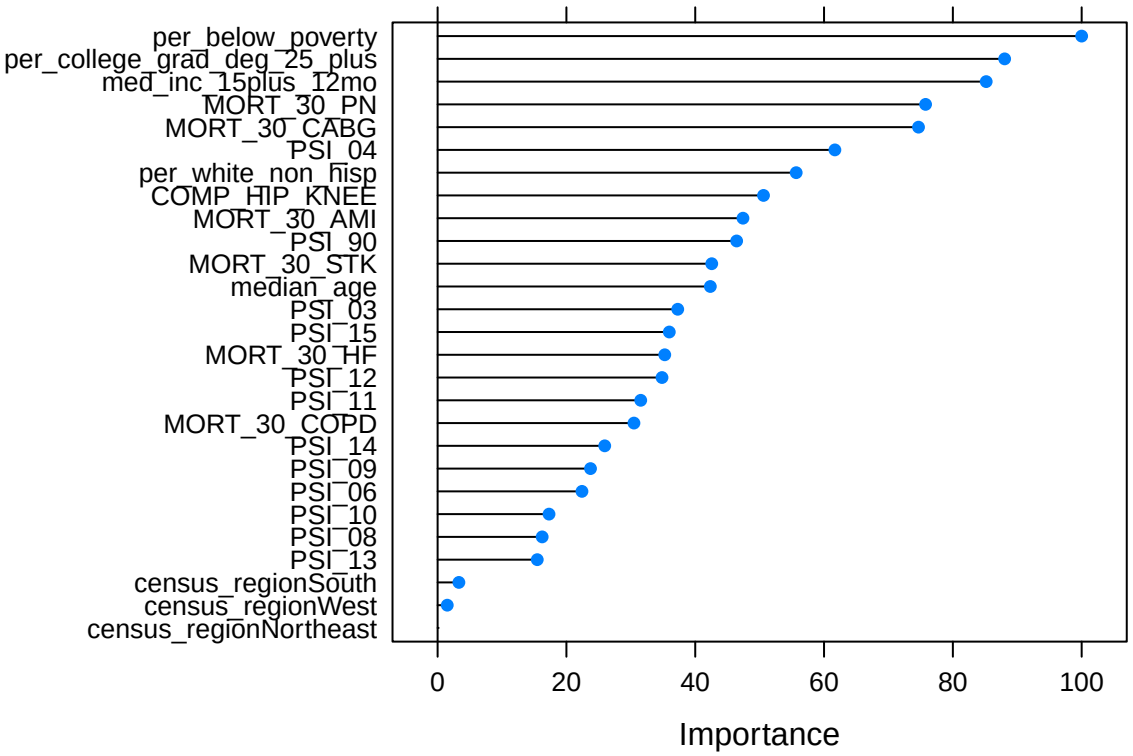
```
        max.auc.polygon=TRUE,
        auc.polygon.col="lightblue",
        print.thres=TRUE,
        main= 'ROC Curve')
```



Bagging

```
#require(caret)

set.seed(1234)
cvcontrol <- trainControl(method="repeatedcv",
                          number = 5, #split 5 times
                          repeats = 2, #repeat 2 times
                          allowParallel=TRUE)
set.seed(1234)
#preProc <- preProcess(train_tree,"corr")
bag <- train(hosp_overall_rating ~ .,
             data=train_tree,
             method="treebag",
             #preProcOptions = preProc,
             trControl=cvcontrol,#implement your train control method from above
             importance=TRUE)# if you want the importance plot
```

```
plot(varImp(bag))
```




```
#Train - Bagginh Confusion Matrix
p1.bag <- predict(bag, train_tree, type = 'raw')
confusionMatrix(p1.bag, train_tree$hosp_overall_rating, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 259   2
##          1   0 174
##
##                Accuracy : 0.9954
##                  95% CI : (0.9835, 0.9994)
##     No Information Rate : 0.5954
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9904
##
##  Mcnemar's Test P-Value : 0.4795
##
##             Sensitivity : 0.9886
```

```
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9923
##              Prevalence : 0.4046
##          Detection Rate : 0.4000
##    Detection Prevalence : 0.4000
##       Balanced Accuracy : 0.9943
##
##        'Positive' Class : 1
##
```

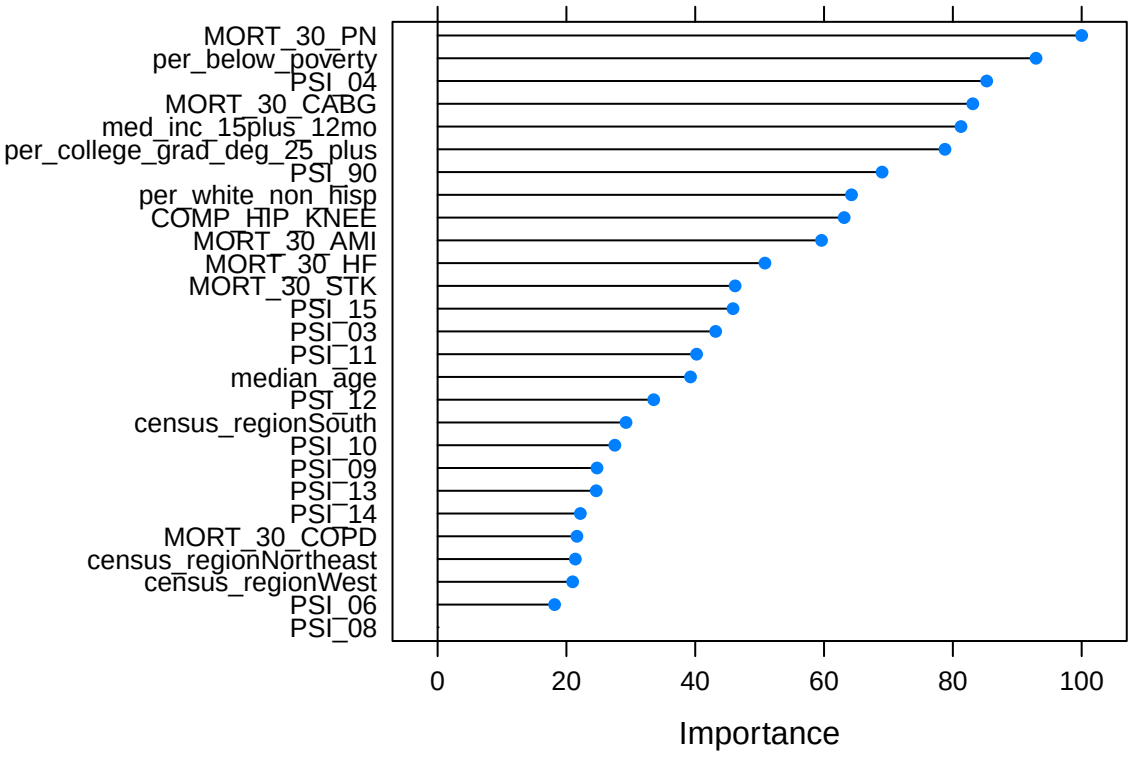
```
#Test - Bagging Confusion Matrix
p2.bag <- predict(bag, test_tree, type = 'raw')
confusionMatrix(p2.bag, test_tree$hosp_overall_rating, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 142  48
##          1  34  78
##
##                Accuracy : 0.7285
##                  95% CI : (0.6746, 0.7778)
##     No Information Rate : 0.5828
##     P-Value [Acc > NIR] : 9.983e-08
##
##                   Kappa : 0.4327
##
##  Mcnemar's Test P-Value : 0.1511
##
##             Sensitivity : 0.6190
##             Specificity : 0.8068
##          Pos Pred Value : 0.6964
##          Neg Pred Value : 0.7474
##              Prevalence : 0.4172
##          Detection Rate : 0.2583
##    Detection Prevalence : 0.3709
##       Balanced Accuracy : 0.7129
##
##        'Positive' Class : 1
##
```

Random Forest

```
# Random Forest
set.seed(1234)
cvcontrol <- trainControl(method="repeatedcv",
                          number = 5, #split 5 times
                          repeats = 2, #repeat 2 times
                          allowParallel=TRUE)
```

```
set.seed(1234)
forest <- train(hosp_overall_rating ~ .,
             data=train_tree,
             method="rf",
             trControl=cvcontrol,
             importance=TRUE)
plot(varImp(forest))
```




```
#Conf Matrix: Train - Random Forest
p1.rf <- predict(forest, train_tree, type = 'raw', positive = '1')
confusionMatrix(p1.rf, train_tree$hosp_overall_rating)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 259   0
##          1   0 176
##
##                Accuracy : 1
##                  95% CI : (0.9916, 1)
##     No Information Rate : 0.5954
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

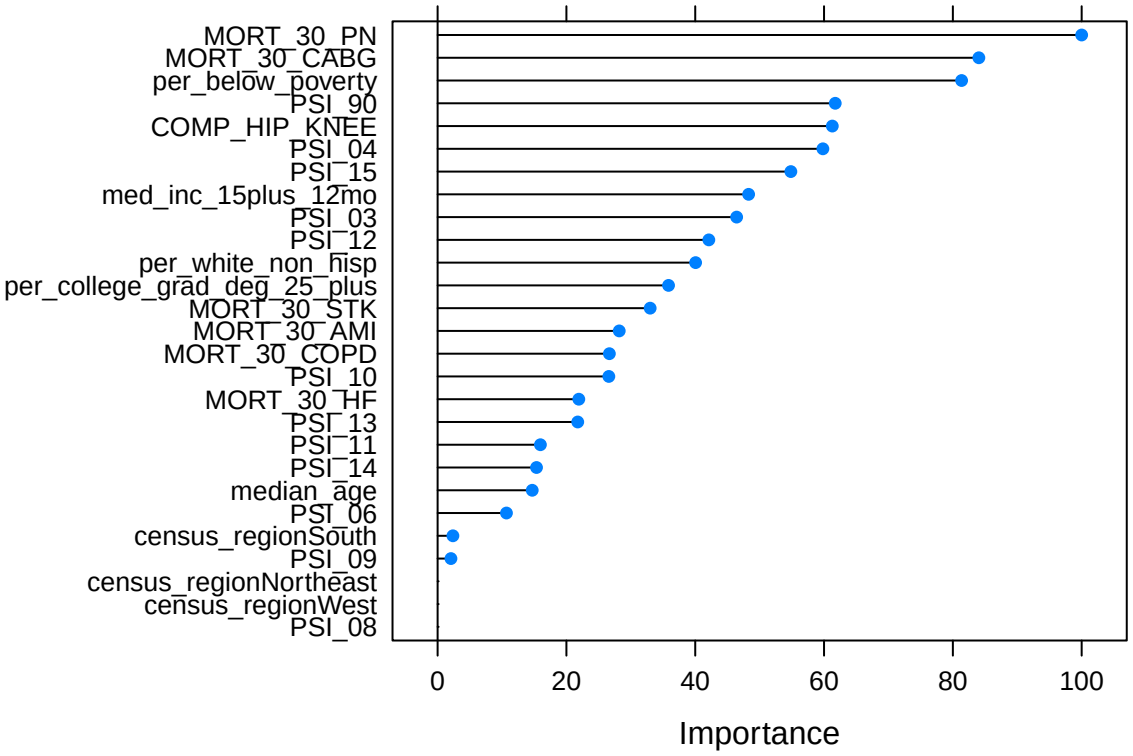
```
##                  Kappa : 1
##
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.5954
##          Detection Rate : 0.5954
##    Detection Prevalence : 0.5954
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : 0
##
```

```
#Conf Matrix: TEST - Random Forest

p2.rf <- predict(forest, test_tree, type = 'raw')
confusionMatrix(p2.rf, test_tree$hosp_overall_rating, positive = '1')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 154  53
##          1  22  73
##
##                Accuracy : 0.7517
##                  95% CI : (0.6989, 0.7994)
##     No Information Rate : 0.5828
##     P-Value [Acc > NIR] : 6.288e-10
##
##                   Kappa : 0.4708
##
##  Mcnemar's Test P-Value : 0.000532
##
##             Sensitivity : 0.5794
##             Specificity : 0.8750
##          Pos Pred Value : 0.7684
##          Neg Pred Value : 0.7440
##              Prevalence : 0.4172
##          Detection Rate : 0.2417
##    Detection Prevalence : 0.3146
##       Balanced Accuracy : 0.7272
##
##        'Positive' Class : 1
##
```

```
### Boosting
set.seed(1234)
cvcontrol <- trainControl(method="repeatedcv",
                          number = 5, #split 5 times
```

```
                           repeats = 2, #repeat 2 times
                           allowParallel=TRUE)
set.seed(1234)
boo <- train(hosp_overall_rating ~ .,
             data=train_tree,
             method="xgbTree",
             trControl=cvcontrol,
             tuneGrid = expand.grid(nrounds = 500,
                                    max_depth = 4,
                                    eta = 0.28,
                                    gamma = 1.8,
                                    colsample_bytree = 1,
                                    min_child_weight = 1,
                                    subsample = 1))
plot(varImp(boo))
```




```
p1.boo <- predict(boo, train_tree, type = 'raw')
confusionMatrix(p1.boo, train_tree$hosp_overall_rating)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction   0   1
##          0 258   6
##          1   1 170
```

```
##
##                Accuracy : 0.9839
##                  95% CI : (0.9671, 0.9935)
##     No Information Rate : 0.5954
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9664
##
##  Mcnemar's Test P-Value : 0.1306
##
##             Sensitivity : 0.9961
##             Specificity : 0.9659
##          Pos Pred Value : 0.9773
##          Neg Pred Value : 0.9942
##              Prevalence : 0.5954
##          Detection Rate : 0.5931
##    Detection Prevalence : 0.6069
##       Balanced Accuracy : 0.9810
##
##        'Positive' Class : 0
##
```

```
p2.boo <- predict(boo, test_tree, type = 'raw')
confusionMatrix(p2.boo, test_tree$hosp_overall_rating)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 142  51
##          1  34  75
##
##                Accuracy : 0.7185
##                  95% CI : (0.6642, 0.7686)
##     No Information Rate : 0.5828
##     P-Value [Acc > NIR] : 6.844e-07
##
##                   Kappa : 0.4099
##
##  Mcnemar's Test P-Value : 0.08266
##
##             Sensitivity : 0.8068
##             Specificity : 0.5952
##          Pos Pred Value : 0.7358
##          Neg Pred Value : 0.6881
##              Prevalence : 0.5828
##          Detection Rate : 0.4702
##    Detection Prevalence : 0.6391
##       Balanced Accuracy : 0.7010
##
##        'Positive' Class : 0
##
```