

# Sample Code - Nurse Call Volume Analysis

Yiyao Zhou

1/24/2019

```
library(dplyr)
library(ggplot2)
library(reshape2)
library(gbm)
```

## Data Preparation

```
# Load the data pulled from database
setwd("~/Desktop/Sample Code")
df <- read.csv("Data.csv")
# check the dataframe structure
# summary(df)
# head(df)

# drop rows with "N/A"
df[df == "N/A"] <- NA
df <- na.omit(df)
# 30-min time interval
df <- df %>% mutate(time = Hour + X30.Min/2)
df <- df[,c(-6, -7)]
# simplify the original nurse call message type
df$Message.Type <- as.character(df$Message.Type)
df$Message.Type[df$Message.Type == "Bed Exit Call - Patient exit from bed has
  been detected"] <- "Bed Exit Call"
df$Message.Type[df$Message.Type == "Bed Disconnect - A bed has been disconnect
  ed from the system."] <- "Bed Disconnect"
df$Message.Type[df$Message.Type == "Pillow Speaker Disconnect - the patient's
  pillow speaker has been disconnected from the system."] <- "Pillow Speaker D
  isconnect"
df$Message.Type[df$Message.Type == "Patient is requesting pain medication."]
  <- "Request Pain Medication"
df$Message.Type[df$Message.Type == "Call Cord Disconnect - Patient's call cor
  d has been disconnected from the system. This patient does not have the abil
  ity to place a call."] <- "Call Cord Disconnect"
df$Message.Type[df$Message.Type == "Patient Equipment 1"] <- "Patient Equipme
  nt"
df$Message.Type <- as.factor(df$Message.Type)
df$Month <- as.factor(df$Month)

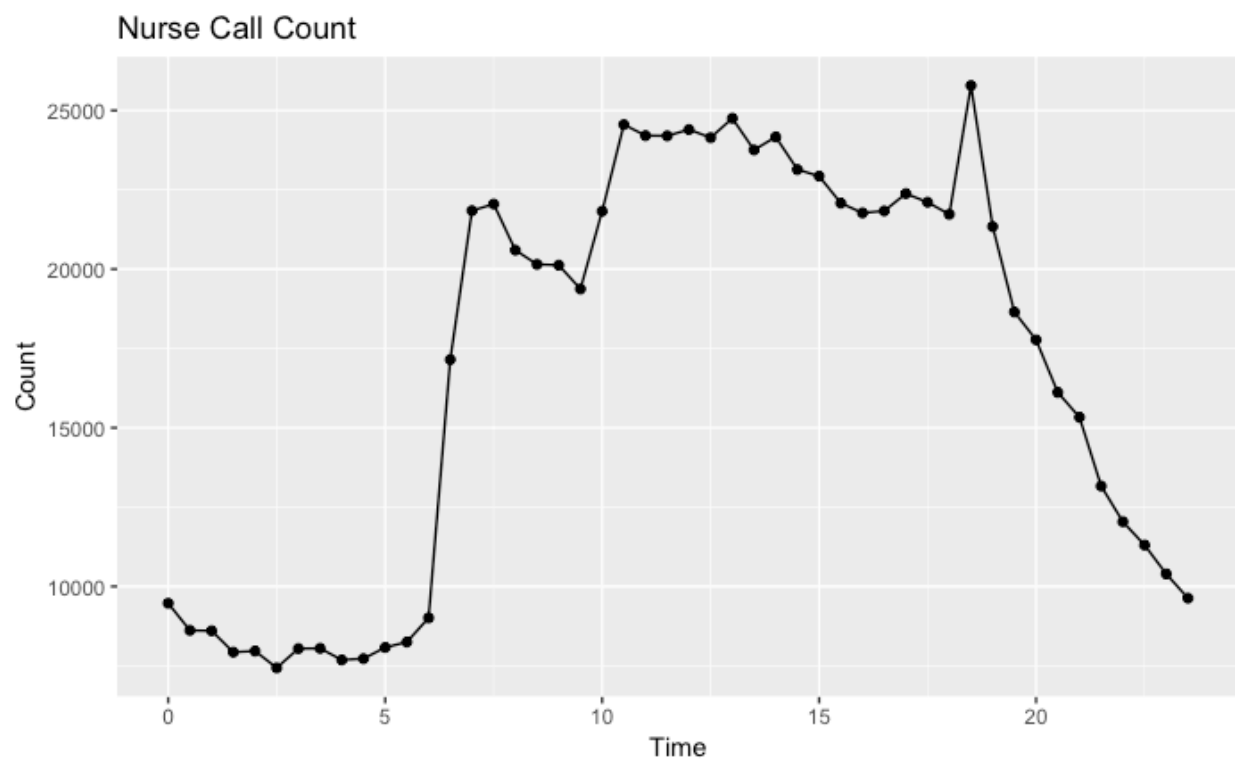
head(df)
```

```
## Recipient.Hospital Recipient.Unit Recipient.ID Message.Type
## 1 Alaska *Alaska PJU7225 Bath switch triggered
## 2 Alaska *Alaska PJU7225 Bed Exit Call
## 3 Alaska *Alaska PJU7225 Normal
## 4 Alaska *Alaska CSU8661 Normal
## 5 Alaska *Alaska PJU7225 Bath switch triggered
## 6 Alaska *Alaska PJU7225 Normal
## Month Count time
## 1 11 2 9.5
## 2 11 1 9.5
## 3 11 1 9.5
## 4 2 2 10.5
## 5 11 1 10.5
## 6 11 2 10.5
```

## Visualization

*# nurse call count vs time*

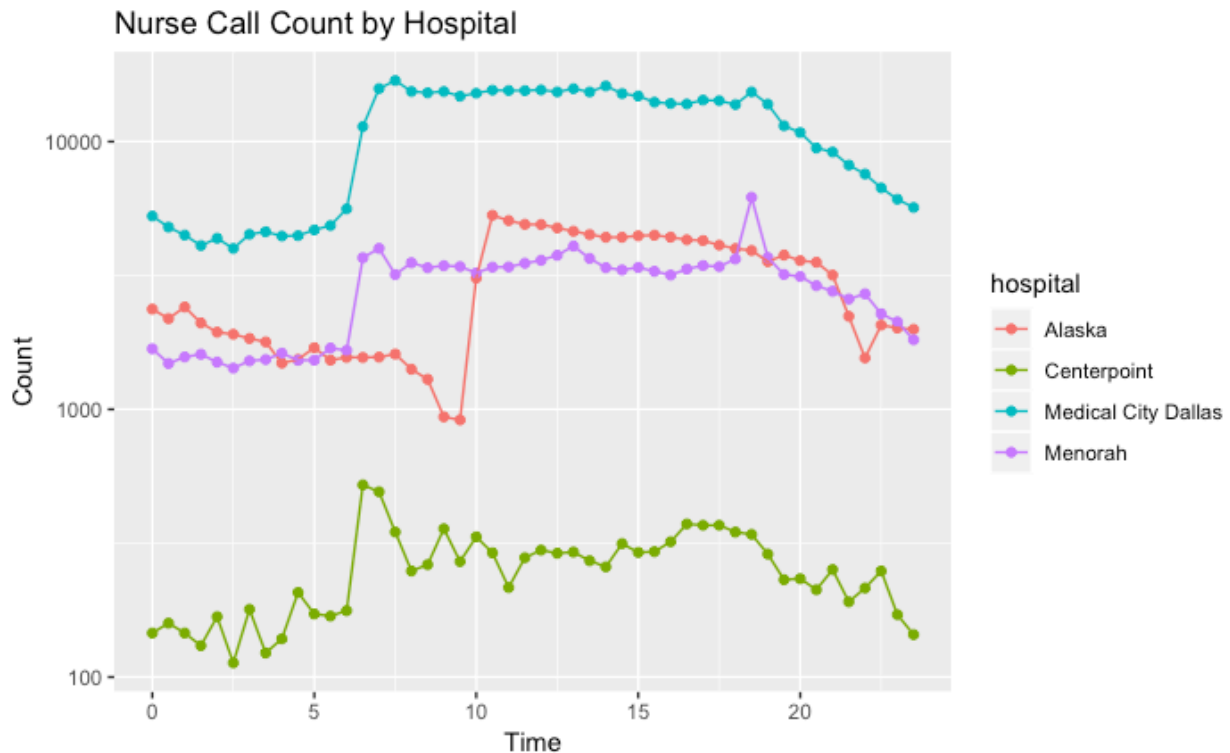
```
all <- aggregate(df$Count, by=list(time = df$time), FUN=sum)
ggplot(all, aes(x = time, y = x)) +
  geom_point() +
  geom_line() +
  labs(title = "Nurse Call Count", x = "Time", y = "Count")
```



```

# nurse call count vs time - group by hospital
hospital <- aggregate(df$Count, by=list(time = df$time, hospital = df$Recipient.Hospital), FUN=sum)
# select hospital that the nurse call is frequently auditted
hospital <- hospital %>% filter(hospital %in% c("Alaska", "Centerpoint", "Medical City Dallas", "Menorah"))
ggplot(hospital, aes(x = time, y = x, col = hospital)) +
  geom_point() +
  geom_line() +
  scale_y_log10() +
  labs(title = "Nurse Call Count by Hospital", x = "Time", y = "Count")

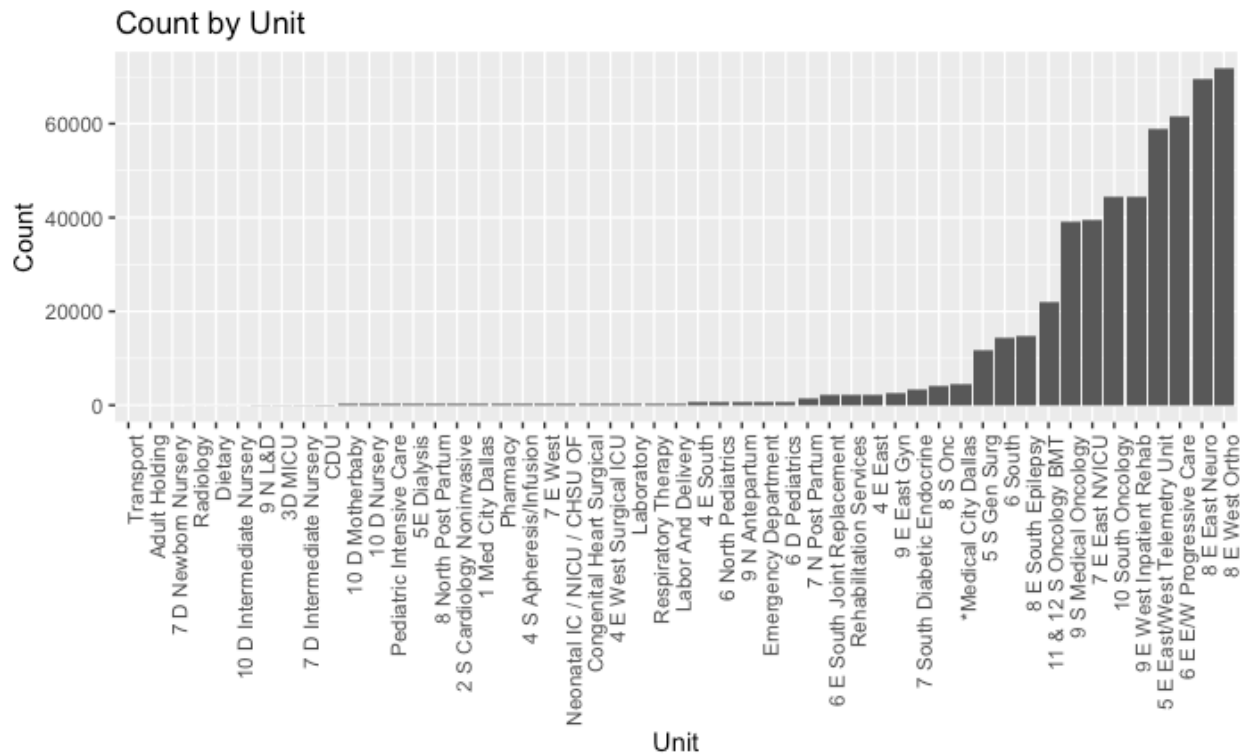
```



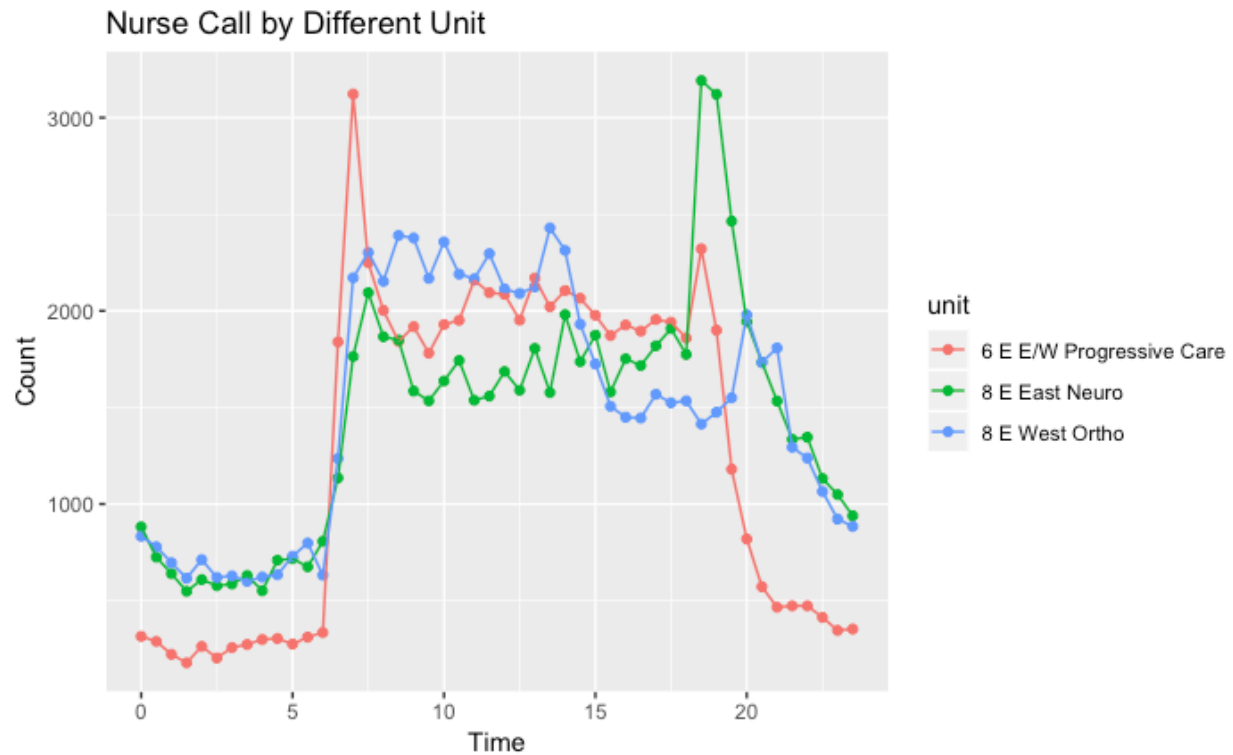
```

# nurse call count - group by unit
# dallas as example
dallas <- subset(df, Recipient.Hospital == "Medical City Dallas")
dallas.agg <- aggregate(dallas$Count, by=list(unit = dallas$Recipient.Unit), FUN=sum)
dallas.agg$unit <- factor(dallas.agg$unit, levels = dallas.agg$unit[order(dallas.agg$x)])
ggplot(dallas.agg, aes(x = unit, y = x)) +
  geom_bar(stat = "identity") +
  labs(title = "Count by Unit", x = "Unit", y = "Count") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

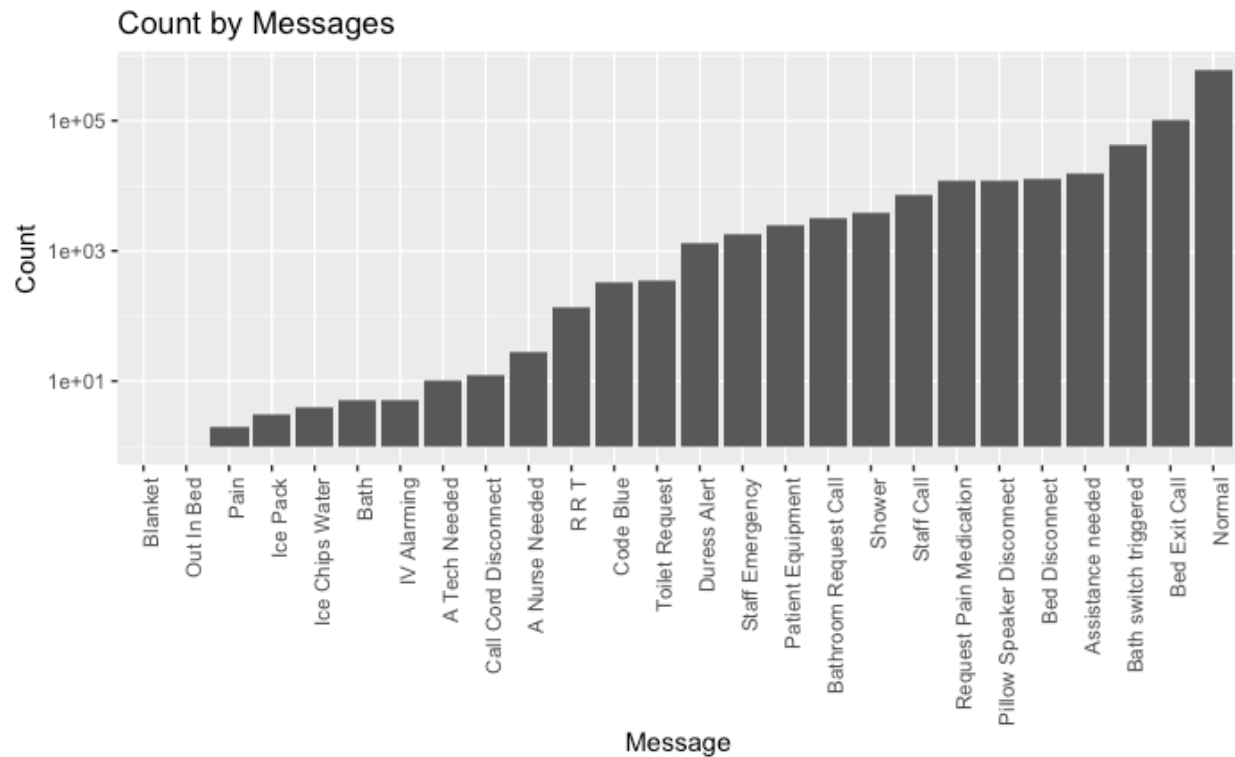
```



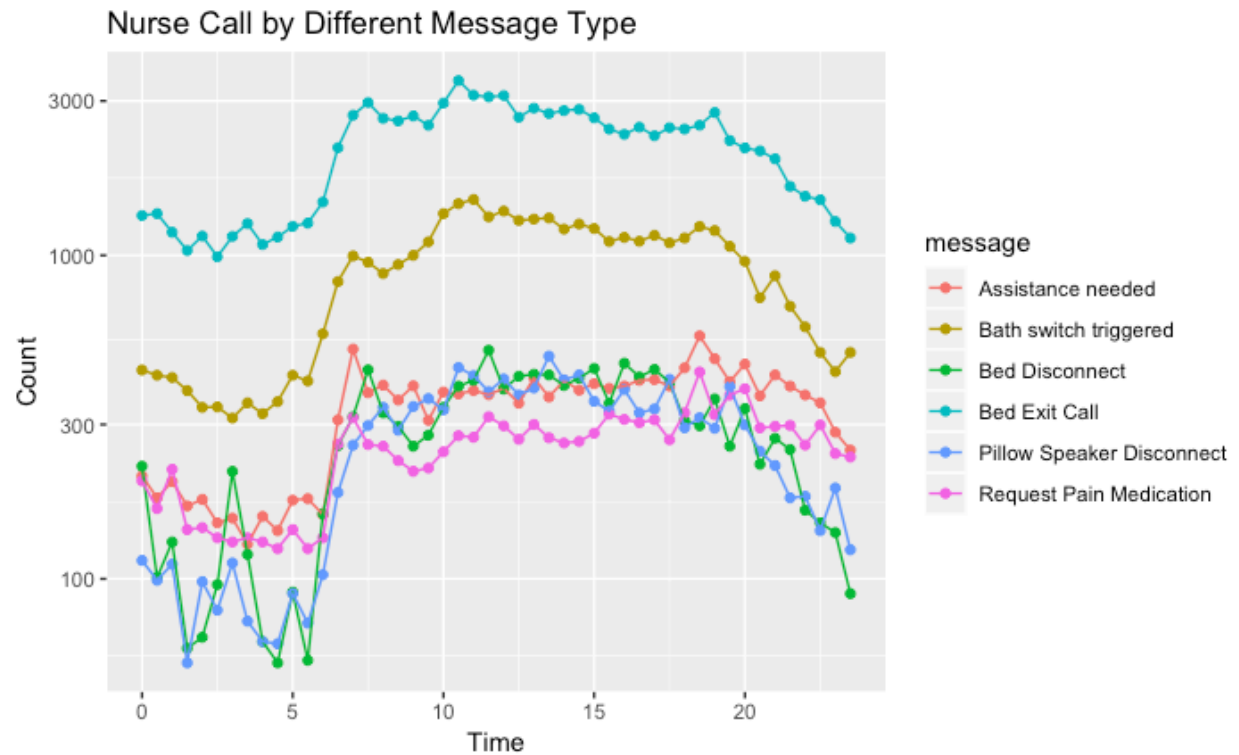
```
# nurse call count vs time - group by unit
dallas.unit <- aggregate(dallas$Count, by=list(time = dallas$time, unit = dal
las$Recipient.Unit), FUN=sum)
# plot top 3 busiest unit
dallas.unit <- dallas.unit %>% filter(unit %in% c("8 E West Ortho", "8 E East
Neuro", "6 E E/W Progressive Care"))
ggplot(dallas.unit, aes(time, x, col = unit)) +
  geom_point() +
  geom_line() +
  labs(title = "Nurse Call by Different Unit", x = "Time", y = "Count")
```



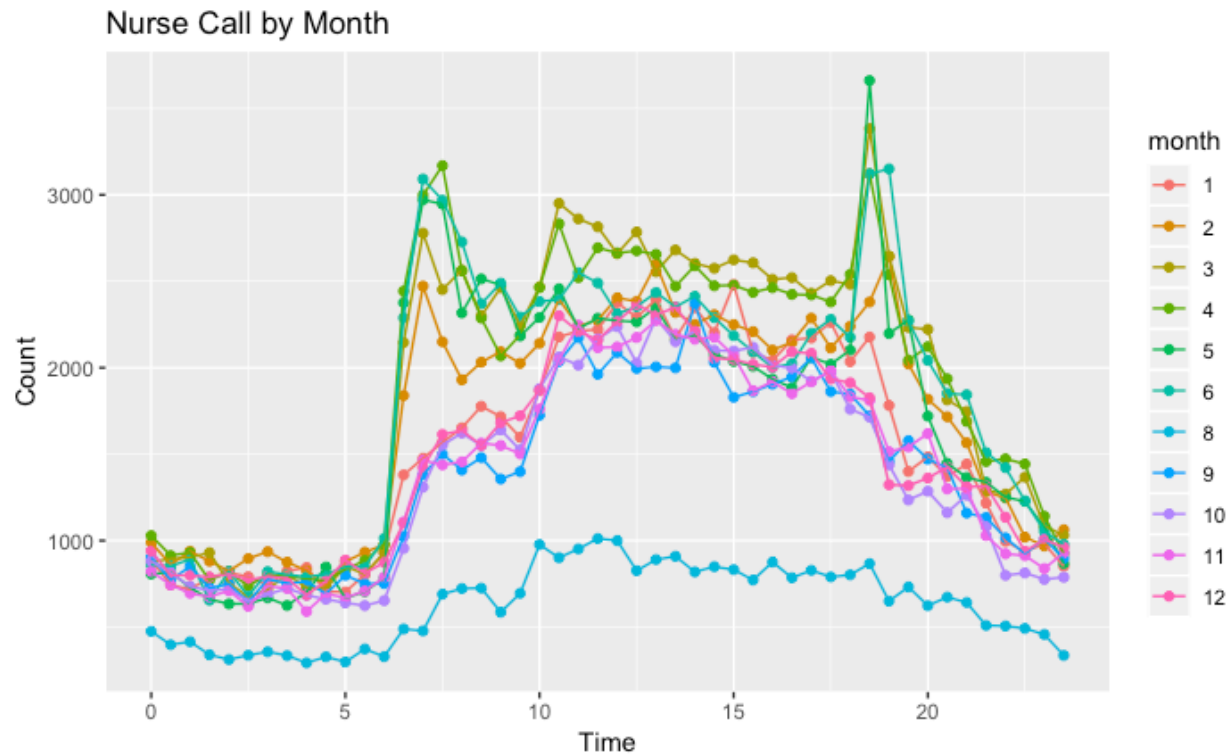
```
# nurse call count - group by message type
message.agg <- aggregate(df$Count, by=list(message = df$Message.Type), FUN=sum)
message.agg$message <- factor(message.agg$message, levels = message.agg$message[order(message.agg$x)])
ggplot(message.agg, aes(x = message, y = x)) +
  geom_bar(stat = "identity") +
  labs(title = "Count by Messages", x = "Message", y = "Count") +
  scale_y_log10() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



```
# nurse call count vs time - group by message type
message <- aggregate(df$Count, by=list(time = df$time, message = df$Message.Type), FUN=sum)
# plot top 6 frequent message type
message <- message %>% filter(message %in% c("Bed Exit Call", "Bath switch triggered", "Assistance needed", "Bed Disconnect", "Pillow Speaker Disconnect", "Request Pain Medication"))
ggplot(message, aes(time, x, col = message)) +
  geom_point() +
  geom_line() +
  scale_y_log10() +
  labs(title = "Nurse Call by Different Message Type", x = "Time", y = "Count")
```



```
# nurse call count vs time - group by month
month <- aggregate(df$Count, by=list(time = df$time, month = df$Month), FUN=sum)
ggplot(month, aes(time, x, col = month))+
  geom_point() +
  geom_line() +
  labs(title = "Nurse Call by Month", x = "Time", y = "Count")
```



## Model with Best Performance

```
df.model <- df %>%
  group_by(Recipient.Hospital, Recipient.Unit, Message.Type, Month, time) %>%

  dplyr::summarise(count = sum(Count))
train.index <- sample(c(1:dim(df.model)[1]), dim(df.model)[1]*0.8)
train.df <- df.model[train.index, ]
test.df <- df.model[-train.index, ]
boost <- gbm(count ~ ., data = train.df, distribution = "gaussian", n.trees = 10000,
             shrinkage = 0.01, interaction.depth = 4)
boost

## gbm(formula = count ~ ., distribution = "gaussian", data = train.df,
##       n.trees = 10000, interaction.depth = 4, shrinkage = 0.01)
## A gradient boosted model with gaussian loss function.
## 10000 iterations were performed.
## There were 5 predictors of which 5 had non-zero influence.

boost.pred <- predict(boost, test.df, n.trees = 10000)
boost.outcome <- data.frame(test.df, boost.pred)
# performance
res1 <- boost.pred - test.df$count
```



```

# Calculate RMSE, assign it to the variable rmse and print it
(rmse <- sqrt(mean(res1^2)))

## [1] 10.71557

# Calculate the standard deviation of review_scores_rating and print it
(sd_review <- sd(test.df$count))

## [1] 29.06742

(R2 <- 1 - (sum((test.df$count-boost.pred)^2)/sum((test.df$count-mean(test.df
$count))^2)))

## [1] 0.8640895

# Prediction Visualization - Example: Normal Nurse Call
n_outcome <- boost.outcome %>% filter(Message.Type == "Normal") %>%
  group_by(Recipient.Hospital,time) %>%
  dplyr::summarise(count = sum(count), pre_count = sum(boost.pred))
melt_outcome <- melt(n_outcome, id.vars = c("Recipient.Hospital","time"))
ggplot(melt_outcome, aes(time, value, col = Recipient.Hospital)) +
  geom_line(aes(linetype=variable)) +
  geom_point(aes(shape=variable)) +
  labs(title ="Normal Nurse Call Prediction", x = "Time", y = "Count")

```

