VUMC Surgery

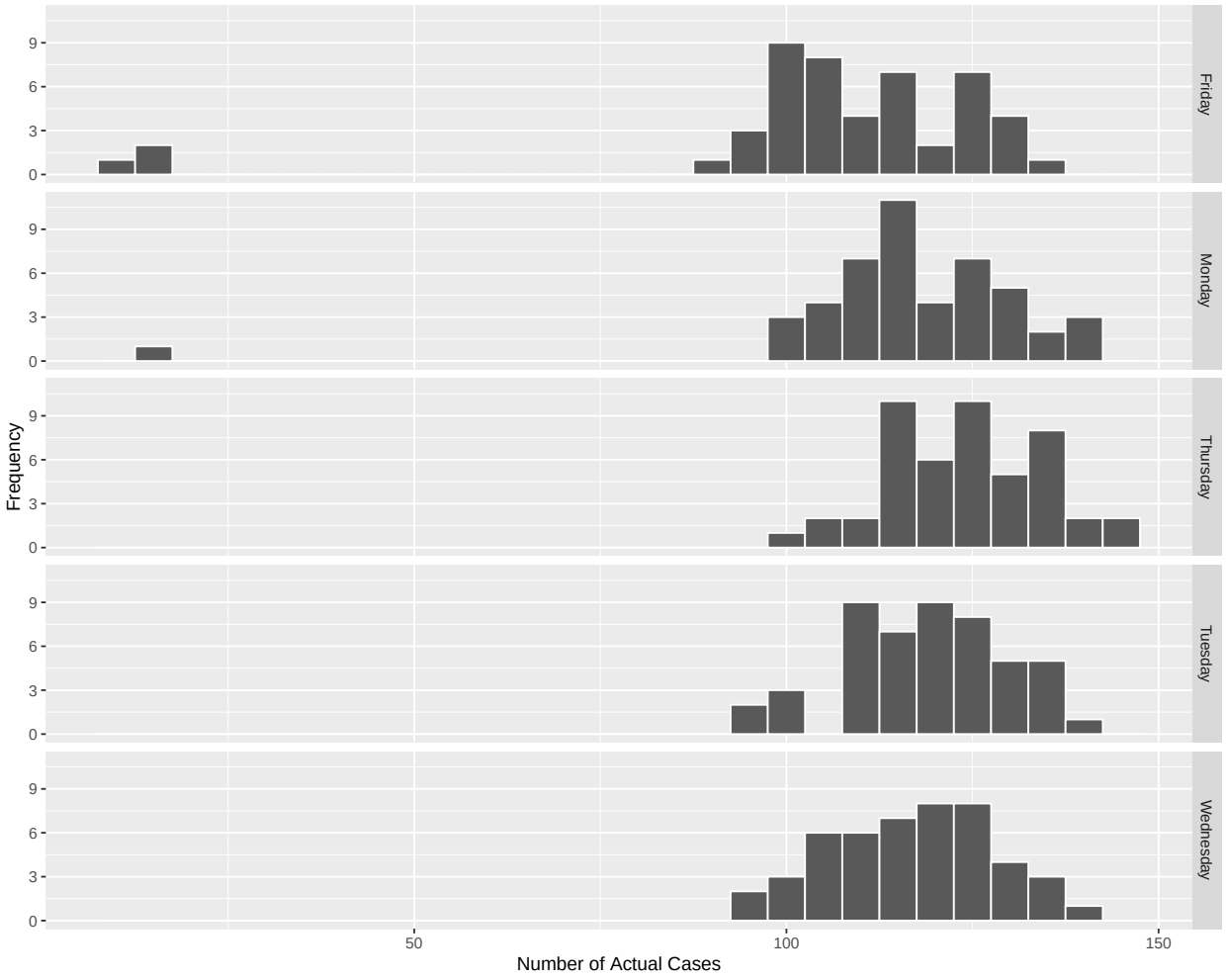
Sina Bahrami

2023-02-08

```
train_set_maker <- function(dataframe, train_ratio) {
  n_total <-   nrow(dataframe)
  n_train <- floor(train_ratio*nrow(dataframe))
  n_test <- n_total - n_train
  df_train <- dataframe[1:n_train,]
  df_test <- dataframe[n_train+1:n_total,] %>% na.omit()
  return(list(df_train, df_test, n_train, n_test))
}
```

Part 1: Comparison of Weekdays

The input data is almost, however, day of week are separately derived form the dates to ensure they correspond to the dates. Also type of SurgDate variable is changed to date.



In the figure below, average and variance of number of surgeries on each weekday are visualized. It shows that Friday has the maximum variance, which may be due to spillover from Thursday, and Thursdays have the maximum average, probably as staff prefer not to have surgeries on Fridays so they shift Fridays' surgeries to Thursday.

Now to test whether all days of the week have equal averages or variances, the following t-tests and F-tests are conducted.

```
df_temp <- df_main %>%
  select(DOW, Actual) %>%
  group_by(DOW)

dows <- unique(df_main$DOW)

df_stat <- data.frame(dow_i=NA, dow_j=NA, ttest_pvalue=NA, ftest_pvalue=NA)

for (dow1 in dows) {
  for (dow2 in dows) {
    new_row <- c(dow_i=dow1, dow_j=dow2,
                 ttest_pvalue=t.test(df_temp$Actual[df_temp$DOW==dow1],
                                     df_temp$Actual[df_temp$DOW==dow2],
                                     paired=FALSE)["p.value"][[1]],
                 ftest_pvalue=var.test(df_temp$Actual[df_temp$DOW==dow1],
```

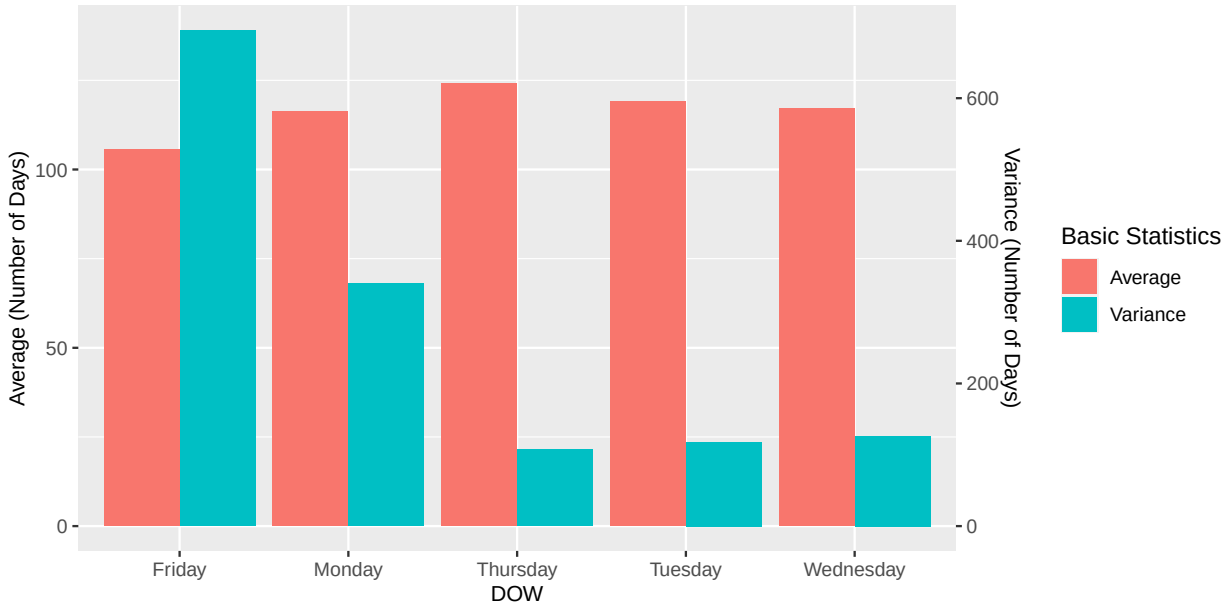
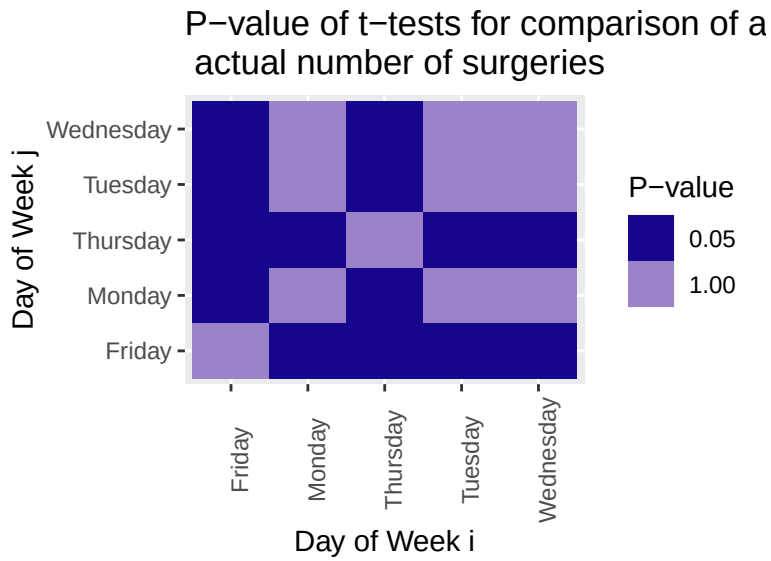


Figure 1: Average and variance of number of surgeries on each weekday

```
                            df_temp$Actual[df_temp$DOW==dow2],
                            paired=FALSE)["p.value"][[1]])
    df_stat <- rbind(df_stat, new_row)
  }
}
df_stat %<>% na.omit() %>%
  mutate(ttest_pvalue=as.numeric(ttest_pvalue),
         ftest_pvalue=as.numeric(ftest_pvalue))
```

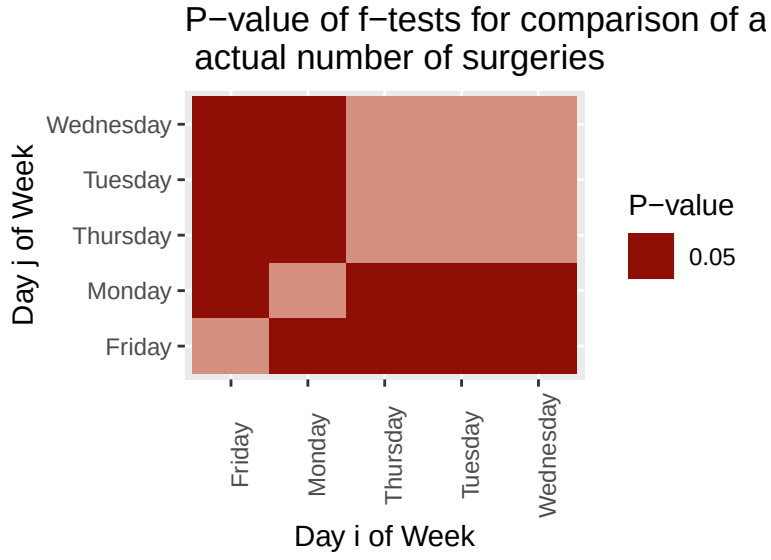
To compare averages of actual surgeries on different days, the result of t-tests between every pair of days are provided below. The results show that there are days with different average number of surgeries. Days with different averages at $\alpha = 0.05$ are colored in dark blue.

```
ggplot(df_stat, aes(x=dow_i, y=dow_j, fill=ttest_pvalue)) +
  geom_tile()+
  labs(x="Day of Week i", y="Day of Week j",
       title="P-value of t-tests for comparison of average\n actual number of surgeries")+
  scale_fill_steps(breaks=c(0,0.05,1), low="blue4", high="white") +
  guides(fill=guide_legend(title="P-value")) +
  theme(axis.text.x=element_text(angle=90))
```



To compare variances of actual surgeries on different days, the result of f-tests between every pair of days are provided below. The results show that there are days with different variances. Days with different variances at $\alpha = 0.05$ are colored in dark red.

```
ggplot(df_stat, aes(x=dow_i, y=dow_j, fill=ftest_pvalue)) +
  geom_tile()+
  labs(x="Day i of Week", y="Day j of Week",
       title="P-value of f-tests for comparison of average\n actual number of surgeries")+
  scale_fill_steps(breaks=c(0,0.05,1), low="red4", high="white") +
  guides(fill=guide_legend(title="P-value")) +
  theme(axis.text.x=element_text(angle=90))
```



Part 2: Longer prediction time and precision trade-off

In this part, model 1 is used to predict actual number of surgeries based on the prediction at i days before (T_i), so it becomes: $T_{actual} = \beta_0 + \beta_1 T_i$

```
predictors <- paste(c("T - "), 5:9, sep="")
df_model1 <- data.frame(Predictor=NA, b0=NA, b1=NA,
                        MSE_train=NA, MSE_test=NA, R2=NA)

# selecing 80% as trianing data and 20% as test data
out <- train_set_maker(df_main, 0.8)
df_train <- out[[1]]
df_test <- out[[2]]
n_train <- out[[3]]
n_test <- out[[4]]

for (predictor in predictors) {
  model <- lm(df_train$Actual ~ df_train[predictor][[1]], data=df_train)
  b0 <- model$coefficients[[1]] %>% round(3)
  b1 <- model$coefficients[[2]] %>% round(3)
  MSE_train <-
    (sum((df_train$Actual - (b0 + b1*df_train[predictor]))^2)/(n_train-2)) %>%
    round(3)
  MSE_test <-
    (sum((df_test$Actual - (b0 + b1*df_test[predictor]))^2)/(n_test-2)) %>%
    round(3)
  R2 <- summary(model)["r.squared"][[1]] %>% round(3)
  df_model1 %<>% rbind(c(predictor, b0, b1, MSE_train, MSE_test, R2))
}
df_model1 %<>% na.omit()

df_model1 %>%
  ggplot(aes(x=Predictor, y=as.numeric(MSE_test))) +
  geom_col() +
  labs(y="MSE (Mean Squared Error)")
```

```
df_model1 %>%
  ggplot(aes(x=Predictor, y=as.numeric(R2))) +
  geom_col() +
  labs(y="R-squared (Goodness of Fit)")
```

The visualization of test MSE vs predictor shows that the earlier is the day of prediction, the higher is MSE, and MSE seems to increase by a constant value every day earlier. On the other hand, R^2 becomes worse (decreases) for any earlier day of prediction, which means the goodness of fit of the linear model is worse for any earlier day of prediction.
Based on the results of training R^2 and test MSE, later days linearly improve error and goodness of fit but there is less difference between "T - 8" and next predictors, so "T - 8" seems to be a better choice in the trade-off.

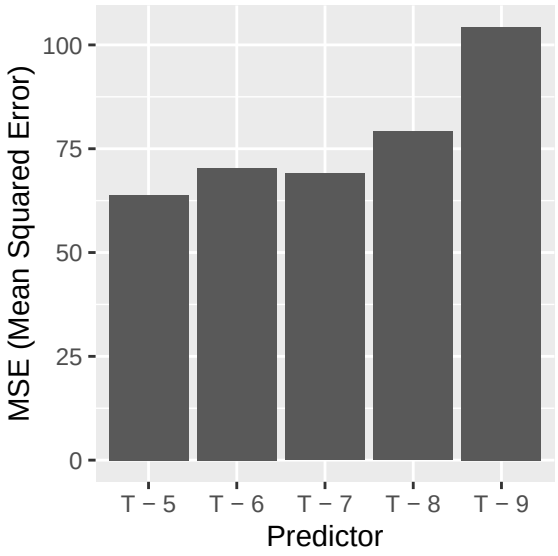


Figure 2: MSE and R-squared for different distance of prediction form the actual day

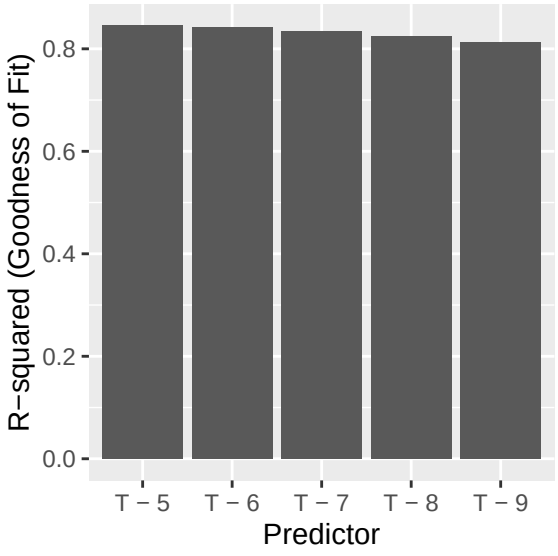


Figure 3: MSE and R-squared for different distance of prediction form the actual day

Part 3: Time Series vs Regression

3.1 Reducing correlation between predictors

Based on the proposed solution for correlation between different predictors, add-on surgeries (difference between two columns) is considered as the new predictor and model 1 and 2 are implemented based on them.

```
df_inc <- df_main
colnames <- names(df_main[3:18])

# Adding increment values to the original data frame
df_inc[gsub("T -", "I -", colnames)] <- (df_main[3:18] - df_main[4:19])


# Adding DOW  dummy variables to df_inc
for (dow in dows) {
  df_inc[dow] <- as.numeric(df_inc$DOW==dow)
}

out <- train_set_maker(df_inc, 0.8)
df_inc_train <- out[[1]]
df_inc_test <- out[[2]]

# Model 1

model1 <- lm(Actual ~ `T - 7` + `I - 7`, data=df_inc_train)

predicted1 <- predict(model1, df_inc_test)
MSE_test1 <- (sum((df_inc_test$Actual - predicted1)^2)/
                (n_test-2)) %>% round(3)
R21 <- summary(model1)["r.squared"][[1]] %>% round(3)

# Model 2
model2 <- lm(Actual ~ `T - 7` + `I - 7` + Tuesday + Wednesday + Thursday +
            Friday, data=df_inc_train)

predicted2 <- predict(model2, df_inc_test)
MSE_test2 <- (sum((df_inc_test$Actual - predicted2)^2)/
                (n_test-2)) %>% round(3)
R22 <- summary(model2)["r.squared"][[1]] %>% round(3)
```

The R-squared obtained for the model 1 is 0.844 and for model 2 is 0.862. So, the addition of add-on surgeries and the effects of day of week have both improved the goodness of fit.

Part 3.2

The figure below represents comparison of prediction by moving average and model 2 of regression which is stratified by days and uses add-on surgery of "T-6" day. The result shows the model 2 of regression provides better predictions of the actual number of surgeries.

```r
n_MA <- 5

df_test <- df_inc %>%
  filter(SurgDate>=as.Date("2012-09-04") & SurgDate<=as.Date("2012-09-14")) %>%
  arrange(-desc(SurgDate))
df_train <- df_inc %>%
  filter(!(SurgDate>=as.Date("2012-09-04") & SurgDate<=as.Date("2012-09-14"))) %>%
  arrange(-desc(SurgDate))
for (row in (n_MA:nrow(df_inc))) {
  df_inc[row+1, "MA"] <- mean(mean(df_inc$Actual[(row-n_MA+1):row]))
}
df_inc %<>% na.omit()
df_test$MA <- df_inc$MA[df_inc$SurgDate>=as.Date("2012-09-04") &
                          df_inc$SurgDate<=as.Date("2012-09-14")]

df_test$Model2 <- predict(model2, df_test)

df_test %>%
  ggplot(aes(x=SurgDate)) +
  geom_line(aes(y=Actual)) +
  geom_line(aes(y=MA), color="blue", linetype="dashed") +
  geom_line(aes(y=Model2), color="red", linetype="dotdash") +
  scale_y_continuous(limits=c(80,150)) +
  labs(x="Surgery Date", y="Number of Surgeries")
```