VUMC Surgery

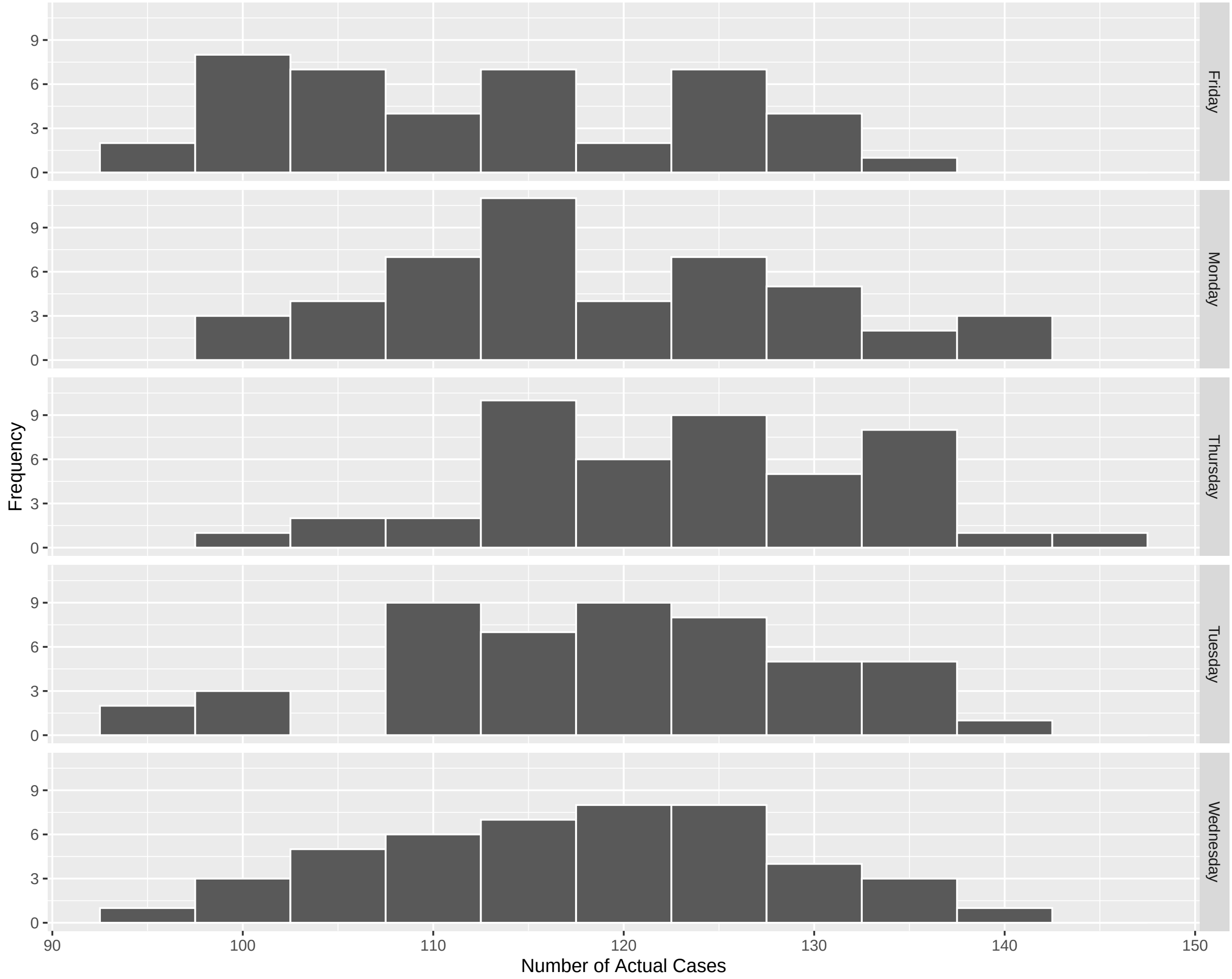
Sina Bahrami

2023-02-08

Part 1: Comparison of Weekdays

The input data is almost clean, however, some minor modifications are made as explained below:

- Day of week are separately derived from the dates to ensure they correspond to the dates.
- Type of SurgDate variable is changed to date.
- Outliers of data are removed only in the histogram below.



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.

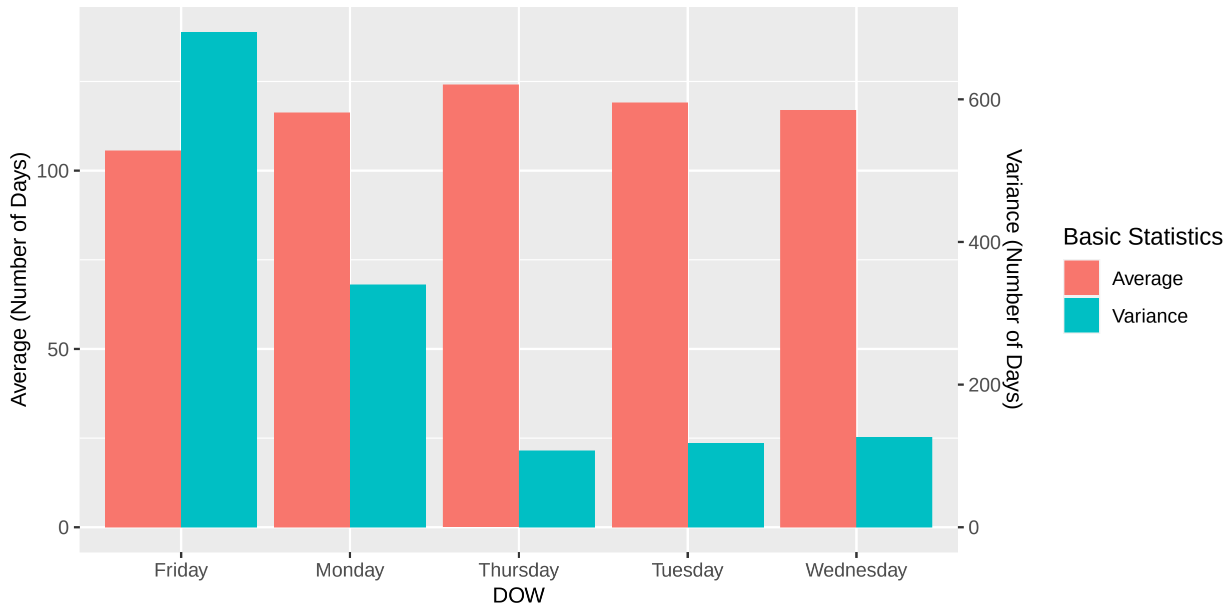
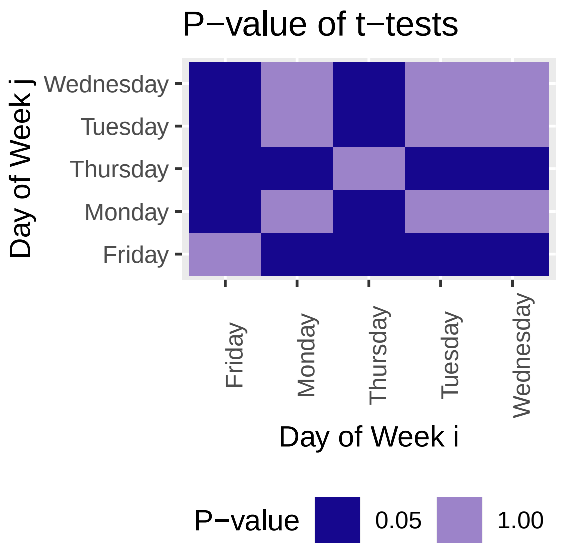
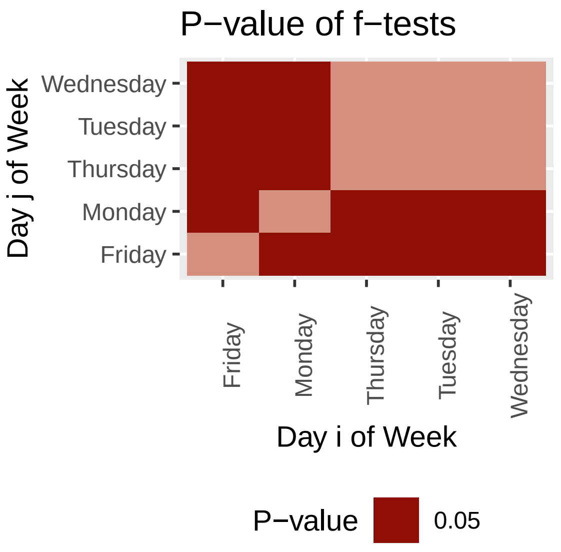


Figure 1: Average and variance of number of surgeries on each weekday

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

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.

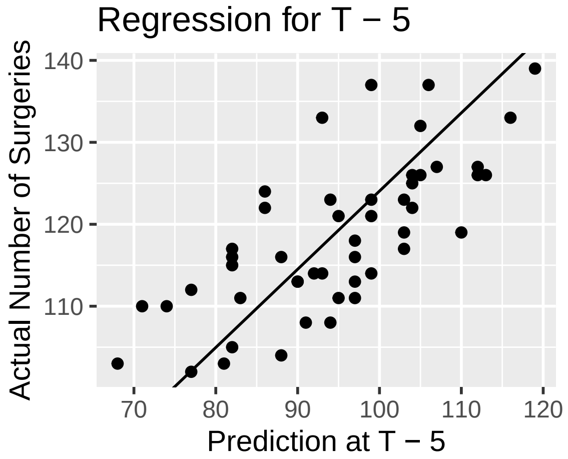


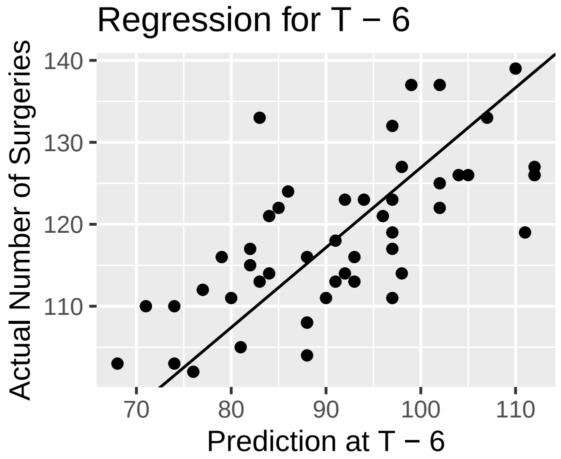


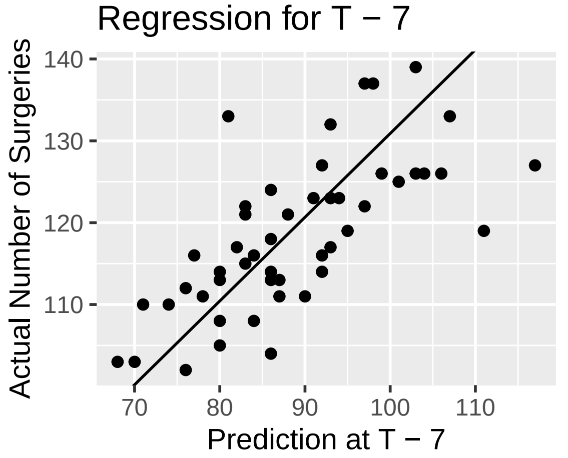
To compare variances of actual surgeries on different days, the result of f-tests between every pair of days are provided. The results show that there are days with different variances. Days with different variances at $\alpha = 0.05$ are colored in dark red.

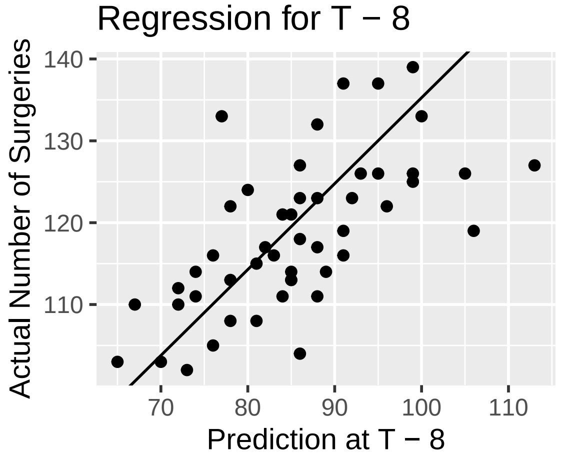
Part 2: Longer prediction time and precision trade-off

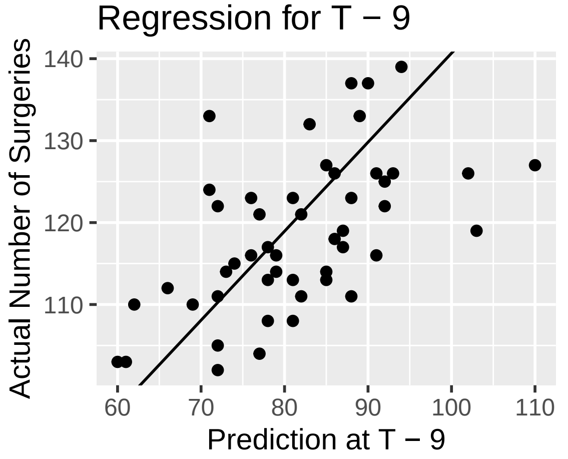
In this part, model 1 is used to predict actual number of surgeries based on the prediction at i days before (T_i), so it becomes: $T_{actual} = \beta_0 + \beta_1 T_i$. The regression plot for each selection of predictor is given below.

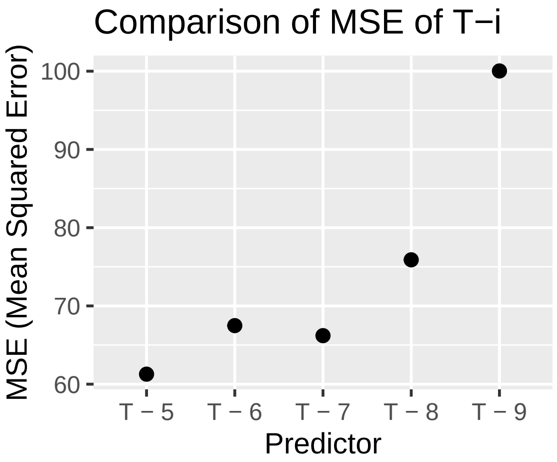

Regression for T − 5

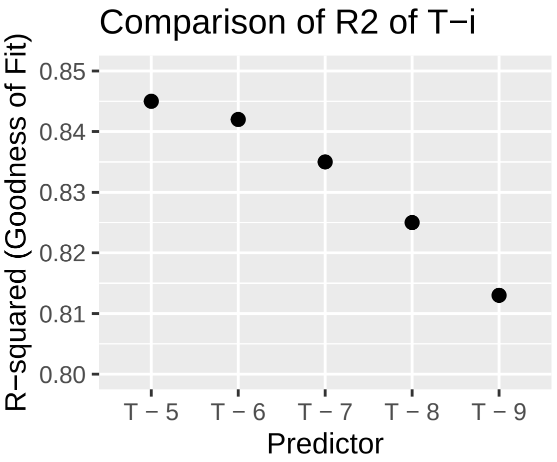

Regression for T − 6


Regression for T − 7


Regression for T − 8


Regression for T − 9


Comparison of MSE of T−i


Comparison of R2 of T−i

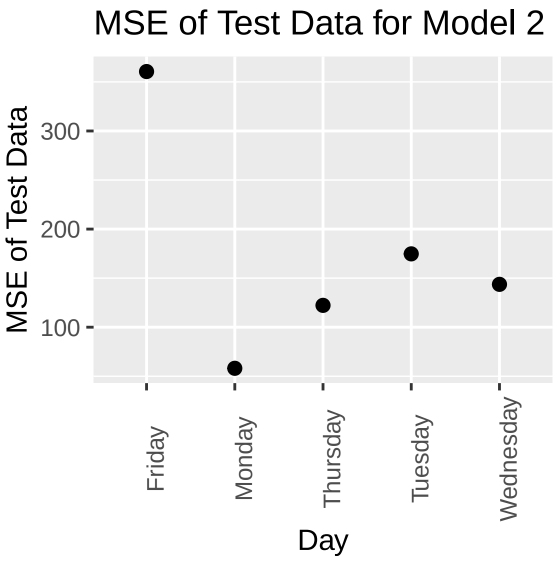
The visualization of test MSE vs predictor shows that the earlier is the day of prediction, the higher is MSE, and MSE seems to increase by a constant value every day earlier. On the other hand, R^2 becomes worse (decreases) for any earlier day of prediction, which means the goodness of fit of the linear model is worse for any earlier day of prediction.
Based on the results of training R^2 and test MSE, later day predictors improve error and goodness of fit but there is a smaller jump between “T - 7” and next predictors, so “T - 7” seems to be a better choice in the trade-off. However, in a real situation, the other side of the trade-off, which is scheduling constraint and the latest prediction day is also important and needs to be balanced with MSE.

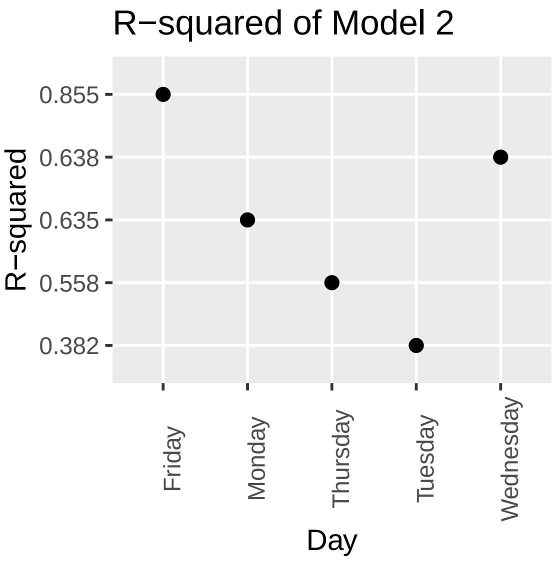
Part 3: Time Series vs Regression

3.1 Reducing correlation between predictors

Based on the proposed solution for correlation between different predictors, add-on surgeries for T-28 to T-7 (difference between each two consecutive columns) is considered as the predictor and model 1 and 2 are implemented based on them. For model 2, however, days are not stratified.

```
## [1] "Test MSE of model 1: 137.22  R-squared of model 1: 0.63"
```


MSE of Test Data for Model 2

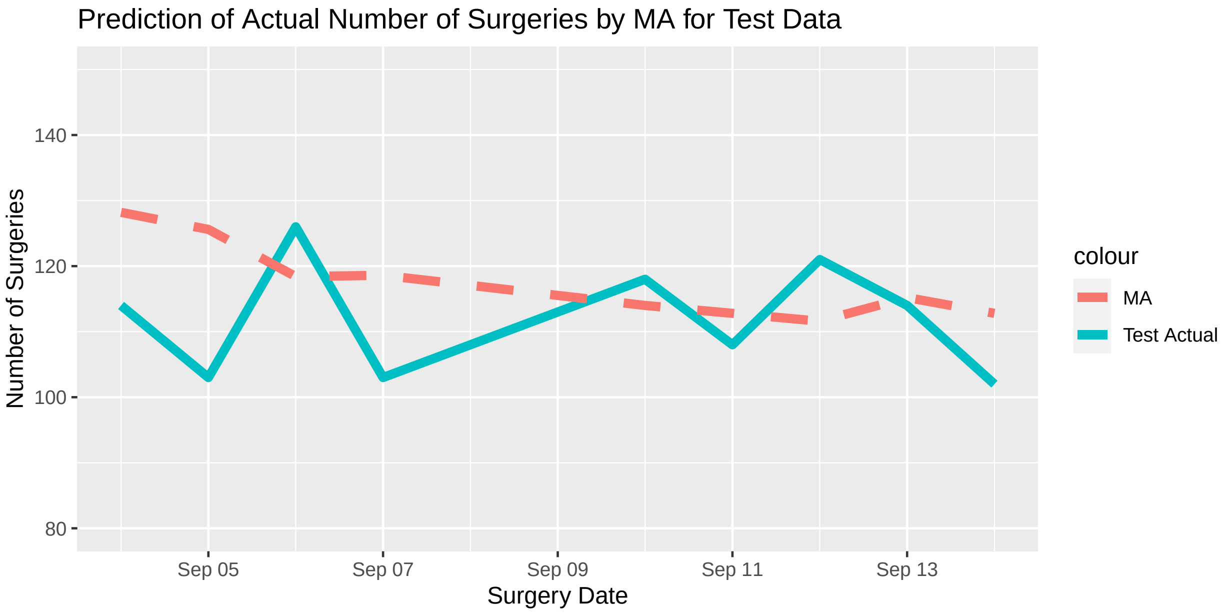

R−squared of Model 2

The R-squared obtained for the model 1 is 0.634 and its test MSE is is 137.223. On the other hand the obtained R-squared and test MSE for model 2 is shown in the figure above for every day of week. These models are compared with those in the lecture:

- The predictors in both models here represent differences between predictions of two consecutive days, which is different from the predictors used in the lecture, which uses only the estimated number of surgeries.
- Model 1 and 2 here use 10 predictors whereas the lecture used only one predictor (not considering the dummy variables). More predictors can improve accuracy by larger dataset for training and testing.
- Both the model 2 here and model 3 in the lecture use stratification by days of week.

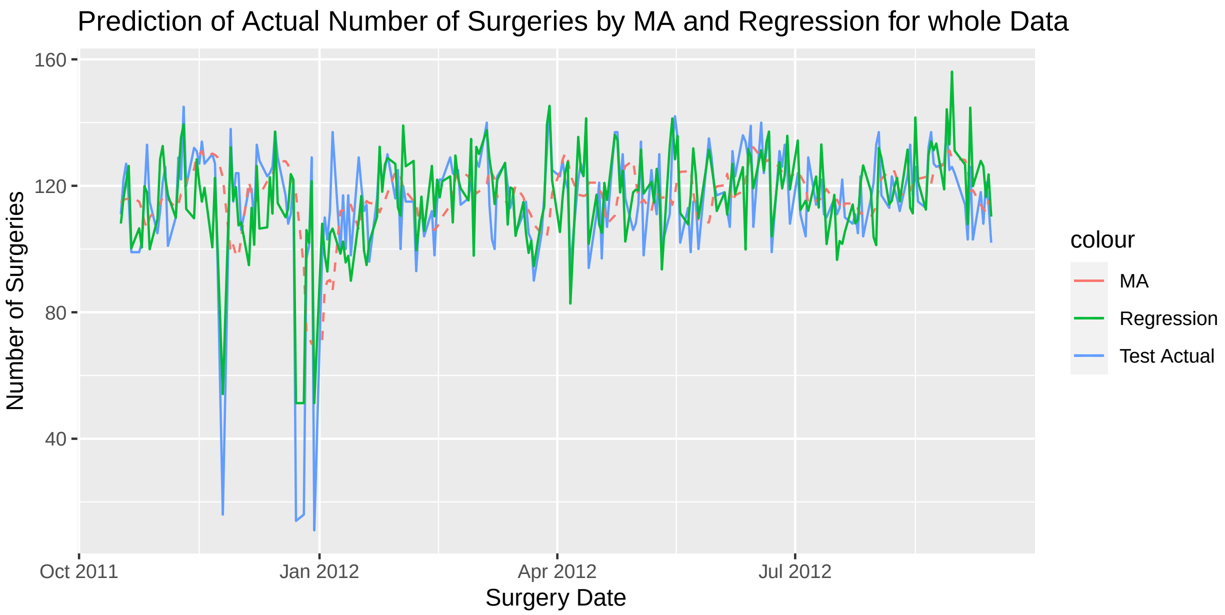
Part 3.2

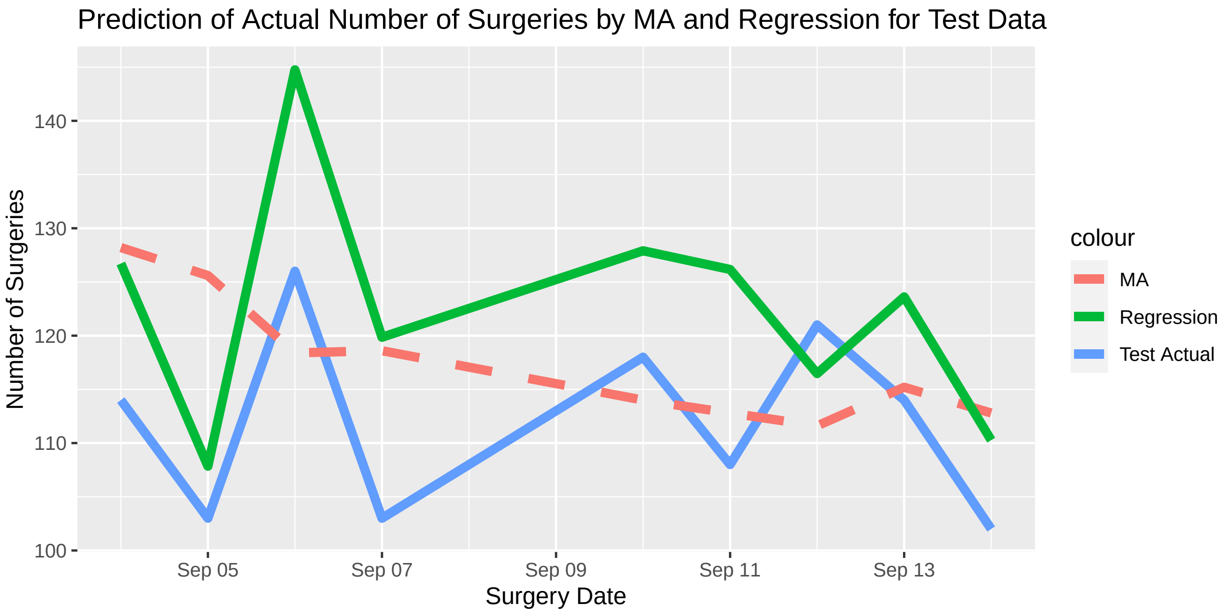
The figure below visualized prediction of number of surgeries by moving average and compares it with test actual values.



Part 3.3

The result of moving average and regression model 1 in part 3.1 are compared visually in the figures below for whole data and for test data. As shown, MA provides a better prediction.


Prediction of Actual Number of Surgeries by MA and Regression for whole Data


Prediction of Actual Number of Surgeries by MA and Regression for Test Data

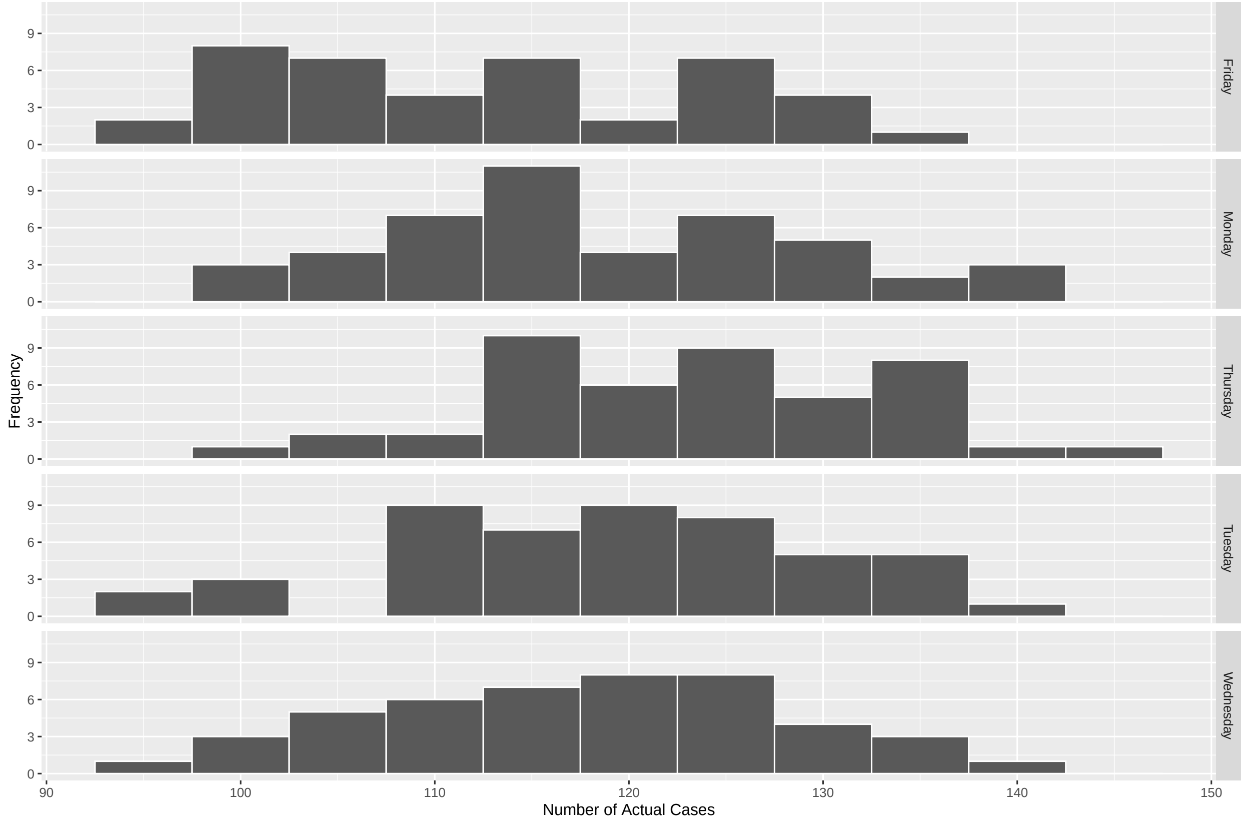
Also the obtained test MSE of regression model is 158.908, which is larger than test MSE of MA 139.889, which supports the comparison provided by the visualization.

There are several reasons that MA has offered better prediction: In calculation of MA the test data have updated and improved the prediction. Also, the regression used in model 1 of part 3.1 uses changes in advance predictions and not the predicted values themselves. In fact the changes in value do not provide any information about the absolute values of prediction. However, if these add-on surgery predictors were accompanied with the predition values (T-i), then a prediction better than MA could be provided.

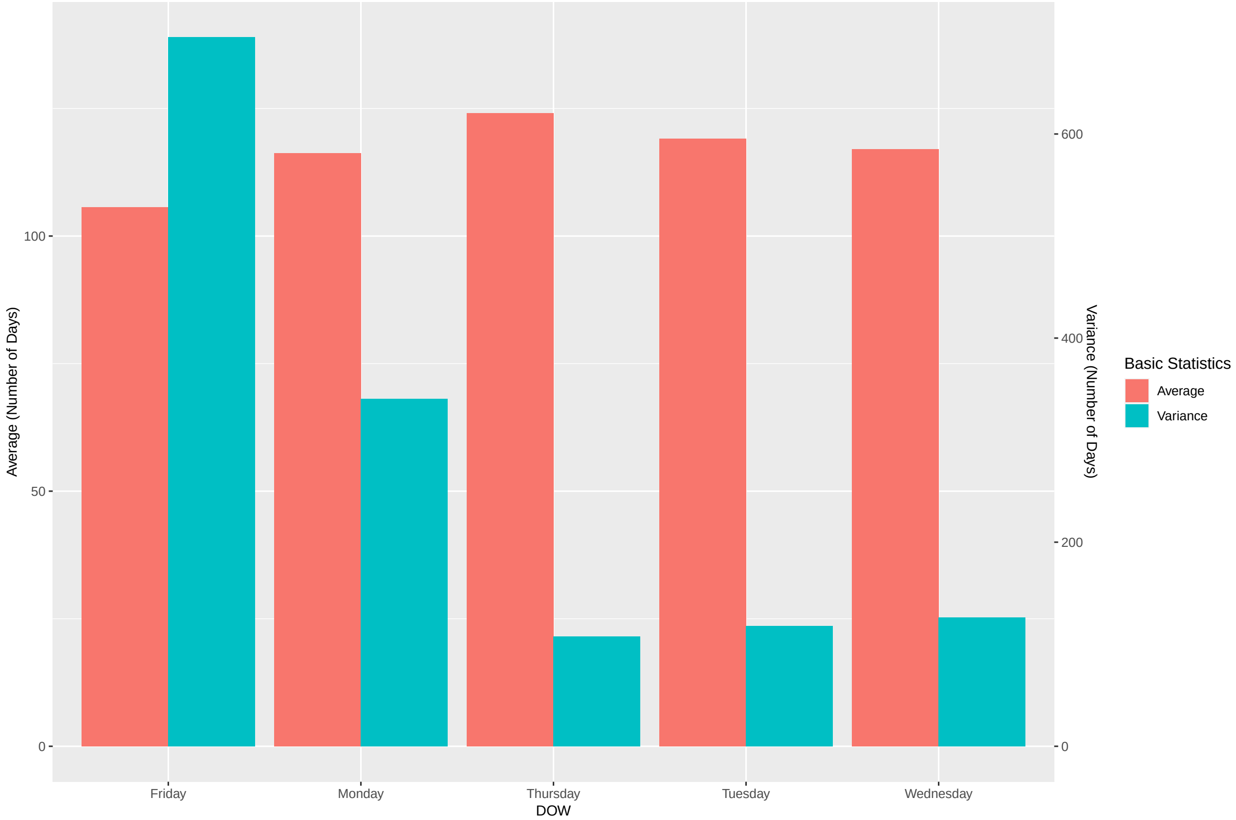
Appendix

Codes

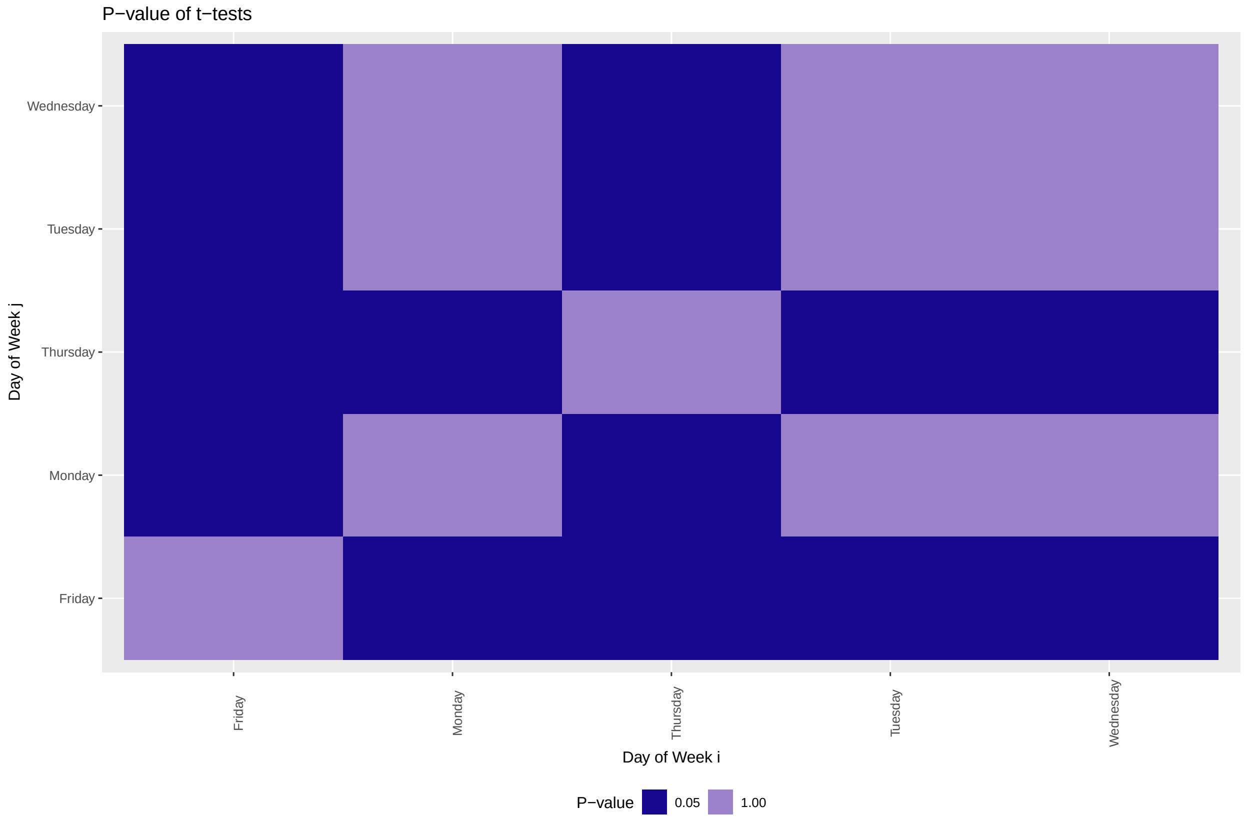
```r
knitr::opts_chunk$set(echo = TRUE)
remove(list=ls())
library("tidyr")
library("dplyr")
library("readr")
library("magrittr")
library("ggplot2")
train_set_maker <- function(dataframe, train_ratio) {
  n_total <-   nrow(dataframe)
  n_train <- floor(train_ratio*nrow(dataframe))
  n_test <- n_total - 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))
}

# Takes a column and detects outliers
detect_outlier <- function(x) {

  Quantile1 <- quantile(x, probs=.25, na.rm = TRUE)
  Quantile3 <- quantile(x, probs=.75, na.rm = TRUE)
  IQR = Quantile3-Quantile1
  x > Quantile3 + (IQR*1.5) | x < Quantile1 - (IQR*1.5)
}

# Takes a data frame and a column and removes the outliers from the
# original data frame
remove_outlier <- function(dataframe,
                           columns=names(dataframe)) {
  # for loop to traverse in columns vector
  for (colmn in columns) {
    # remove observation if it satisfies outlier function
    dataframe <- dataframe[!detect_outlier(dataframe[[colmn]]), ]
  }
  return(dataframe)
}
df_main <- read_csv("VUMC_Data.csv", show_col_types=FALSE )
df_main %<>%
  mutate(SurgDate=as.Date(SurgDate, format="%m/%d/%Y"),
         DOW=weekdays.Date(SurgDate))
main_cols <- names(df_main)
df_main %>%
  remove_outlier(columns=main_cols[startsWith(main_cols, "T -") |
                                     startsWith(main_cols, "Actual")]) %>%
  ggplot() +
  geom_histogram(aes(x=Actual), binwidth=5, color="white") +
  facet_grid(rows=vars(DOW)) +
  labs(x="Number of Actual Cases", y="Frequency")
```



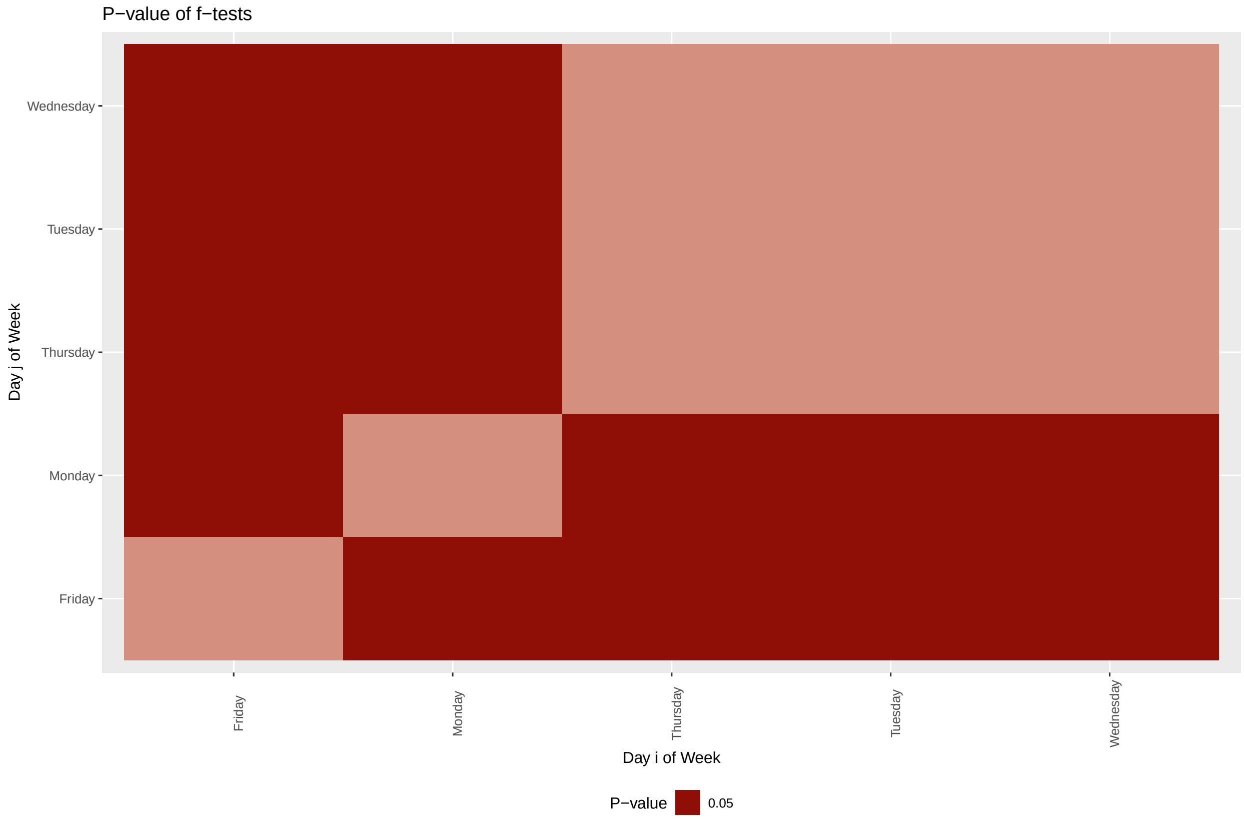
```
df_dow_actual <- df_main %>%
  group_by(DOW) %>%
  summarize(dow_actual_avg=mean(Actual),
            dow_actual_var=var(Actual))

coeff <- 5

df_dow_actual %>%
  mutate(dow_actual_var=dow_actual_var/coeff) %>%
  pivot_longer(cols=c(dow_actual_avg, dow_actual_var),
               names_to="stat", values_to="stat_val") %>%
  arrange(-desc(DOW)) %>%
  ggplot() +
  geom_col(aes(x=DOW, y=stat_val, fill=stat), position="dodge") +
  scale_y_continuous(name="Average (Number of Days)",
                     sec.axis =
                      sec_axis(~.*coeff, name="Variance (Number of Days)")) +
  scale_fill_discrete(labels=c("Average", "Variance"))+
  guides(fill=guide_legend(title="Basic Statistics", position="bottom")) +
  theme(axis.title = element_text(size=10))
```



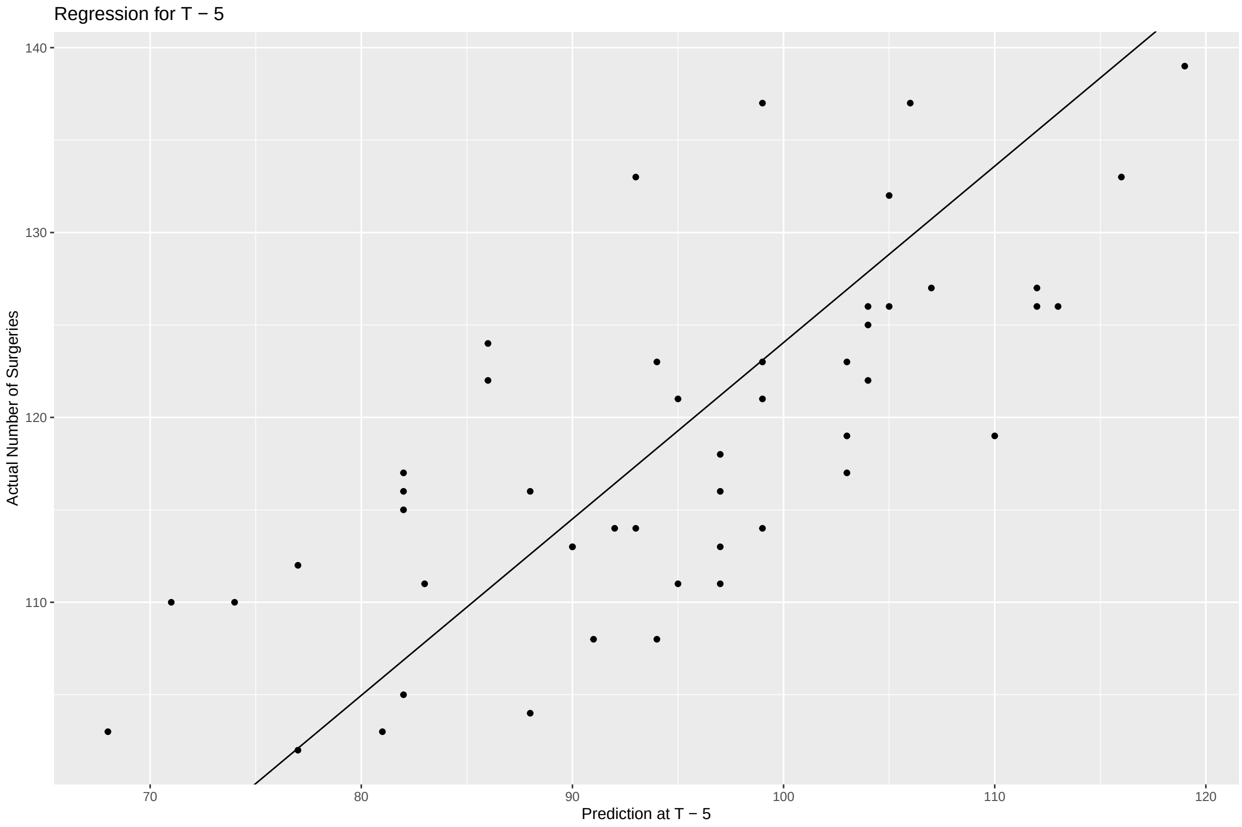
```r
df_temp <- df_main %>%
  select(DOW, Actual) %>%
  group_by(DOW)

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],
                                     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))
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")+
  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), legend.position="bottom")
```

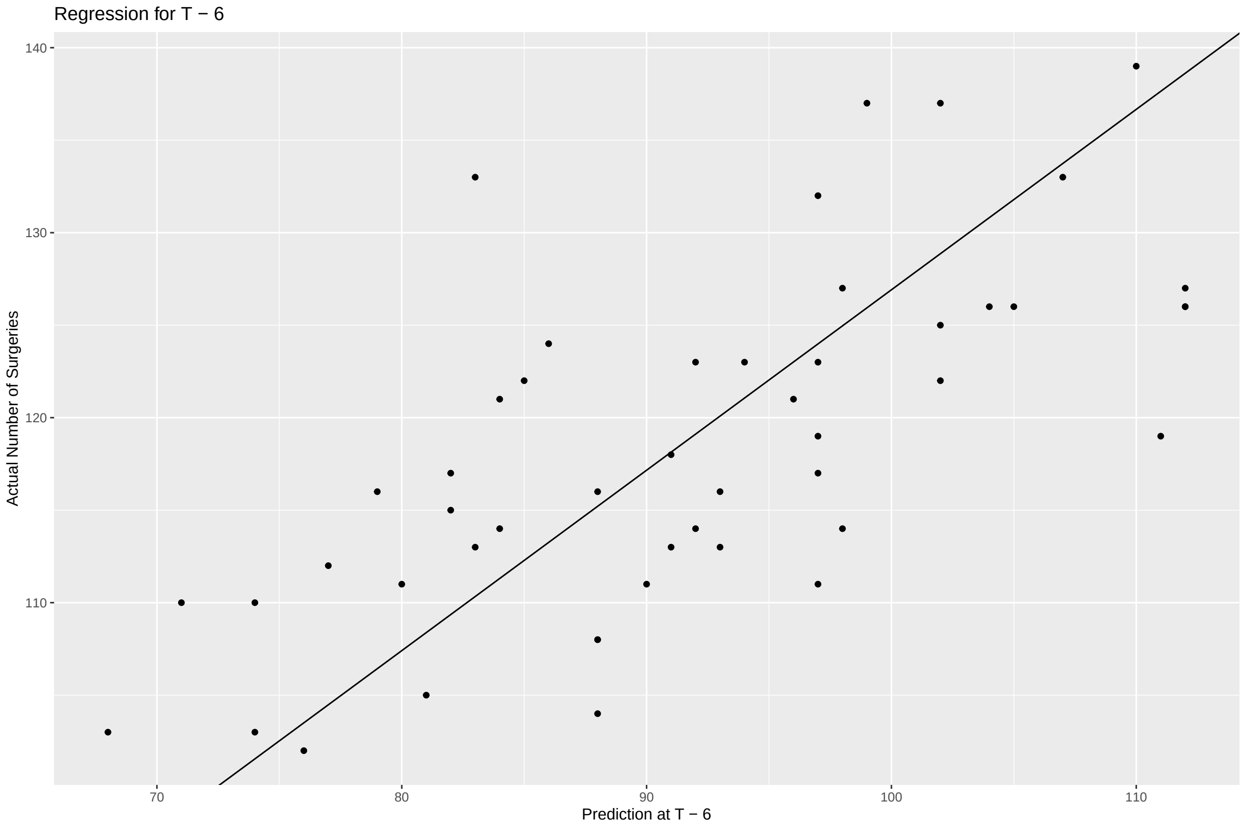


```
ggplot(df_stat, aes(x=dow_i, y=dow_j, fill=ftest_pvalue)) +
  geom_tile()+
  labs(x="Day i of Week", y="Day j of Week",
       title="P-value of f-tests")+
  scale_fill_steps(breaks=c(0,0.05,1), low="red4", high="white") +
  guides(fill=guide_legend(title="P-value")) +
  theme(axis.text.x=element_text(angle=90),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

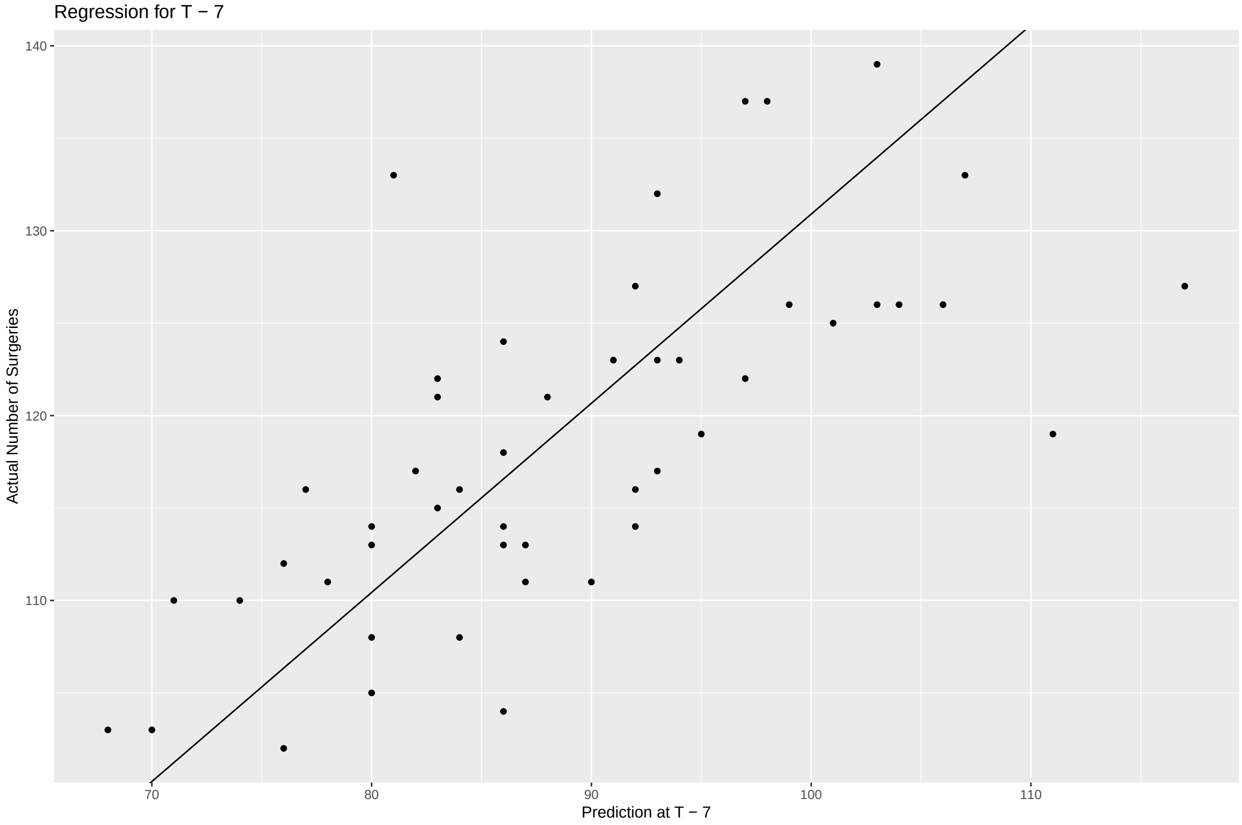


```
predictors <- paste(c("T - "), 5:9, sep="")
df_model1 <- data.frame(Predictor=NA, b0=NA, b1=NA,
                        MSE_train=NA, MSE_test=NA, R2=NA)

# 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)) %>%
    round(3)
  MSE_test <-
    (sum((df_test$Actual - (b0 + b1*df_test[predictor]))^2)/n_test) %>%
    round(3)
  R2 <- summary(model)["r.squared"][[1]] %>% round(3)
  df_model1 %<>% rbind(c(predictor, b0, b1, MSE_train, MSE_test, R2))

  print(ggplot(df_test) +
          geom_point(aes(x=df_test[predictor][[1]], y= Actual)) +
          geom_abline(intercept=model$coefficients[1],
                      slope=model$coefficients[2]) +
          labs(x=paste("Prediction at",predictor),
               y="Actual Number of Surgeries",
               title=paste("Regression for",predictor)))

}
```

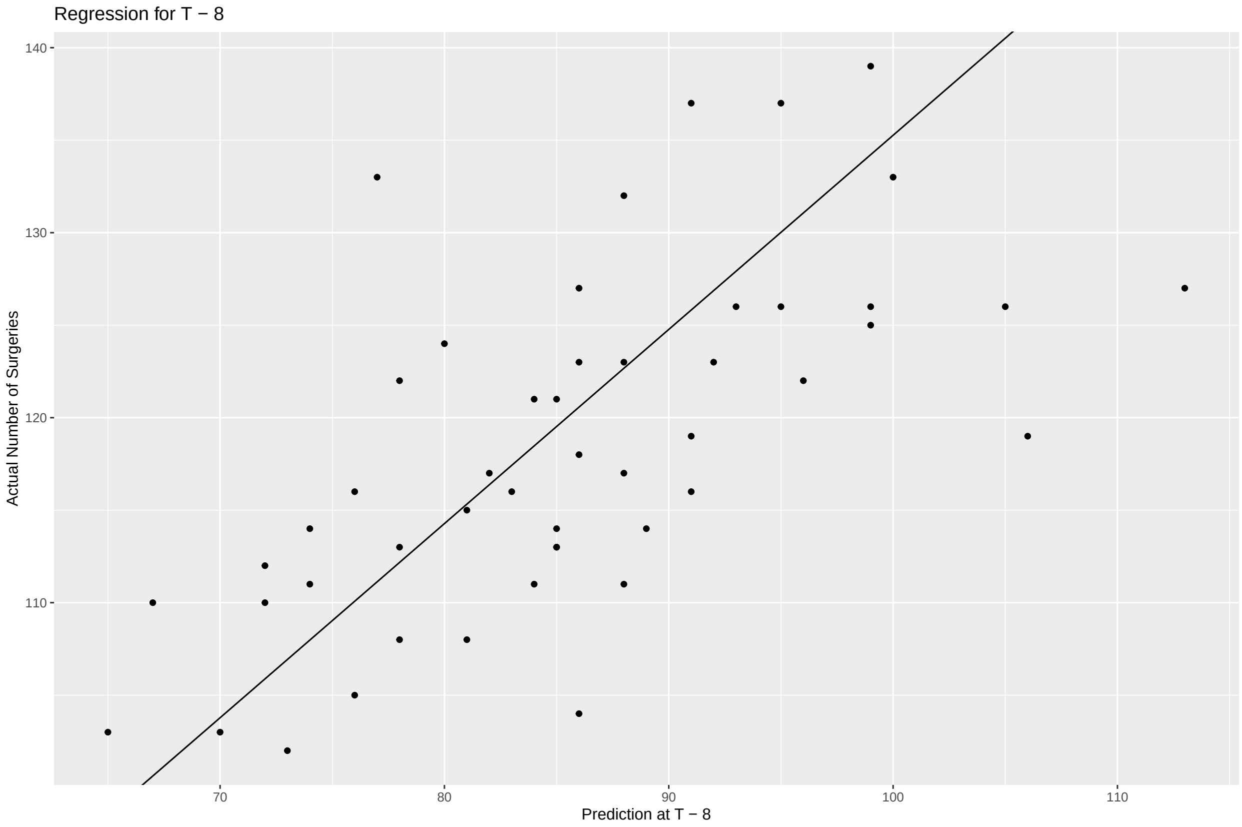


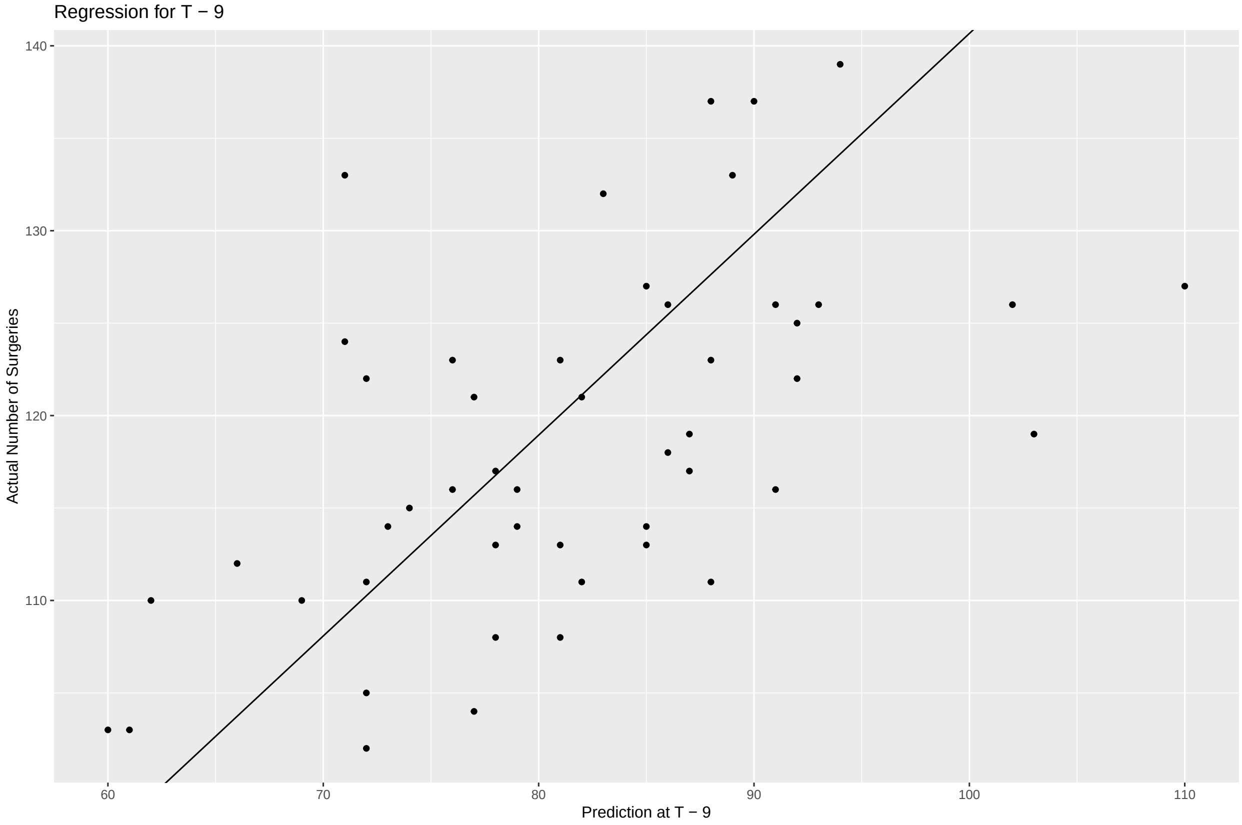


Regression for T − 6

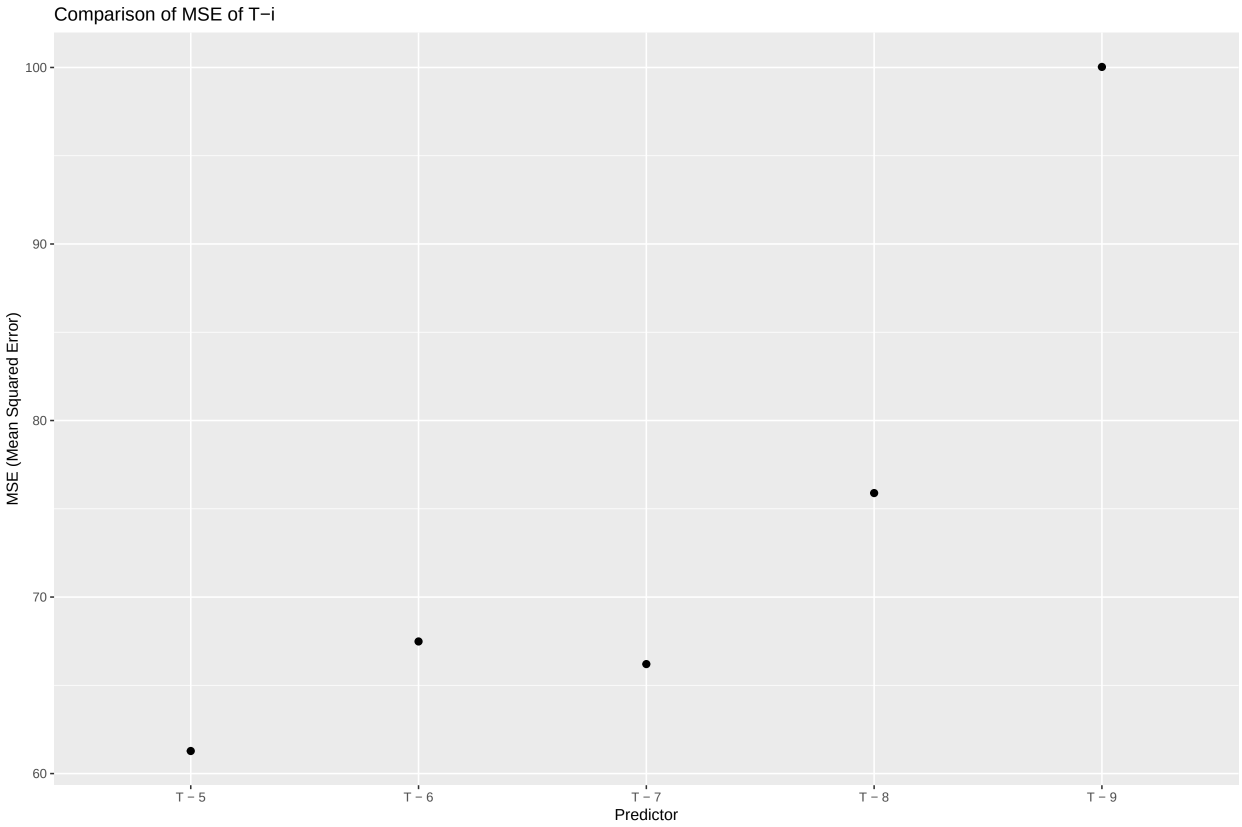


Regression for T − 7

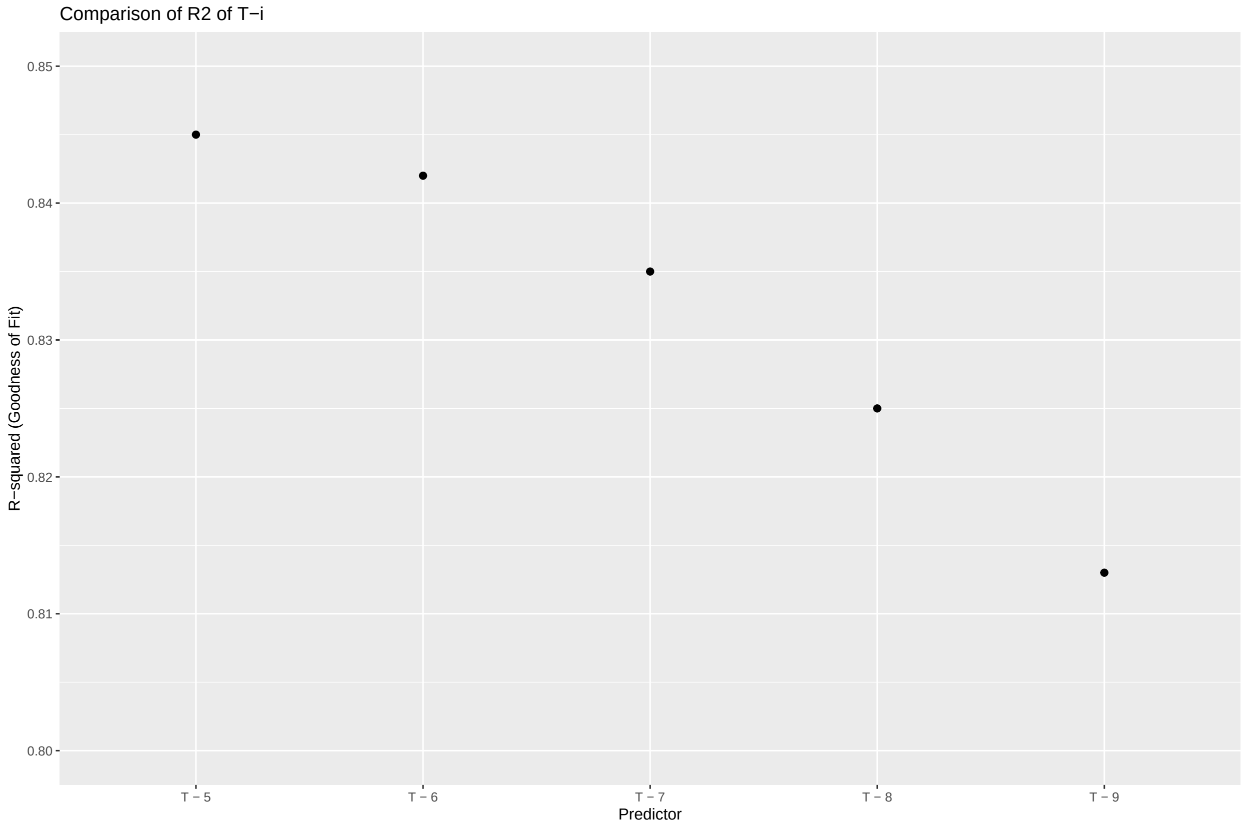

Regression for T − 8
Actual Number of Surgeries
140
130
120
110
70
80
90
100
110
Prediction at T − 8


Regression for T − 9
Actual Number of Surgeries
140
130
120
110
60
70
80
90
100
110
Prediction at T − 9

```
df_model1 %<>% na.omit()
df_model1 %>%
  ggplot(aes(x=Predictor, y=as.numeric(MSE_test))) +
  geom_point(size=2) +
  labs(y="MSE (Mean Squared Error)", title="Comparison of MSE of T-i")
```



```
df_model1 %>%
  mutate(R2=as.numeric(R2)) %>%
  ggplot(aes(x=Predictor, y=R2)) +
  geom_point(size=2) +
  labs(y="R-squared (Goodness of Fit)", title="Comparison of R2 of T-i") +
  scale_y_continuous(limits = c(0.8,0.85))
```



```
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])

# Separating training and test data
out <- train_set_maker(df_inc, 0.8)
df_inc_train <- out[[1]]
df_inc_test <- out[[2]]

# Model 1
model1 <- lm(Actual ~ `I - 28` + `I - 21` + `I - 14` + `I - 13` + `I - 12` +
               `I - 11` + `I - 10` + `I - 9` + `I - 8` + `I - 7`,
             data=df_inc_train)

predicted1 <- predict(model1, df_inc_test)
MSE_test1 <- (sum((df_inc_test$Actual - predicted1)^2)/
               n_test) %>% round(3)
R21 <- summary(model1)["r.squared"][[1]] %>% round(3)
sprintf("Test MSE of model 1: %.2f  R-squared of model 1: %.2f", MSE_test1, R21)
```

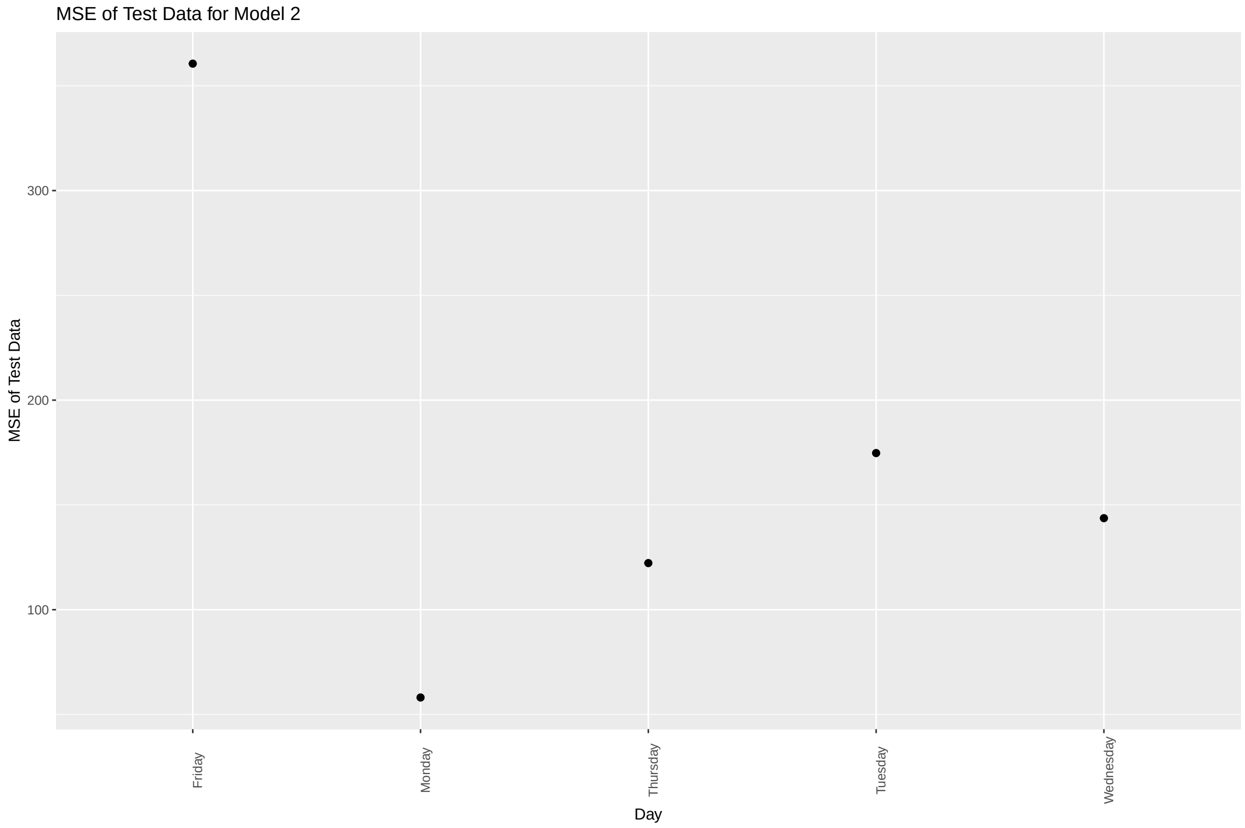
```
## [1] "Test MSE of model 1: 137.22  R-squared of model 1: 0.63"
```

```
df_model2 <- data.frame(DOW=NA, MSE_test2=NA, R22=NA)
# Stratifying days for model 2
```

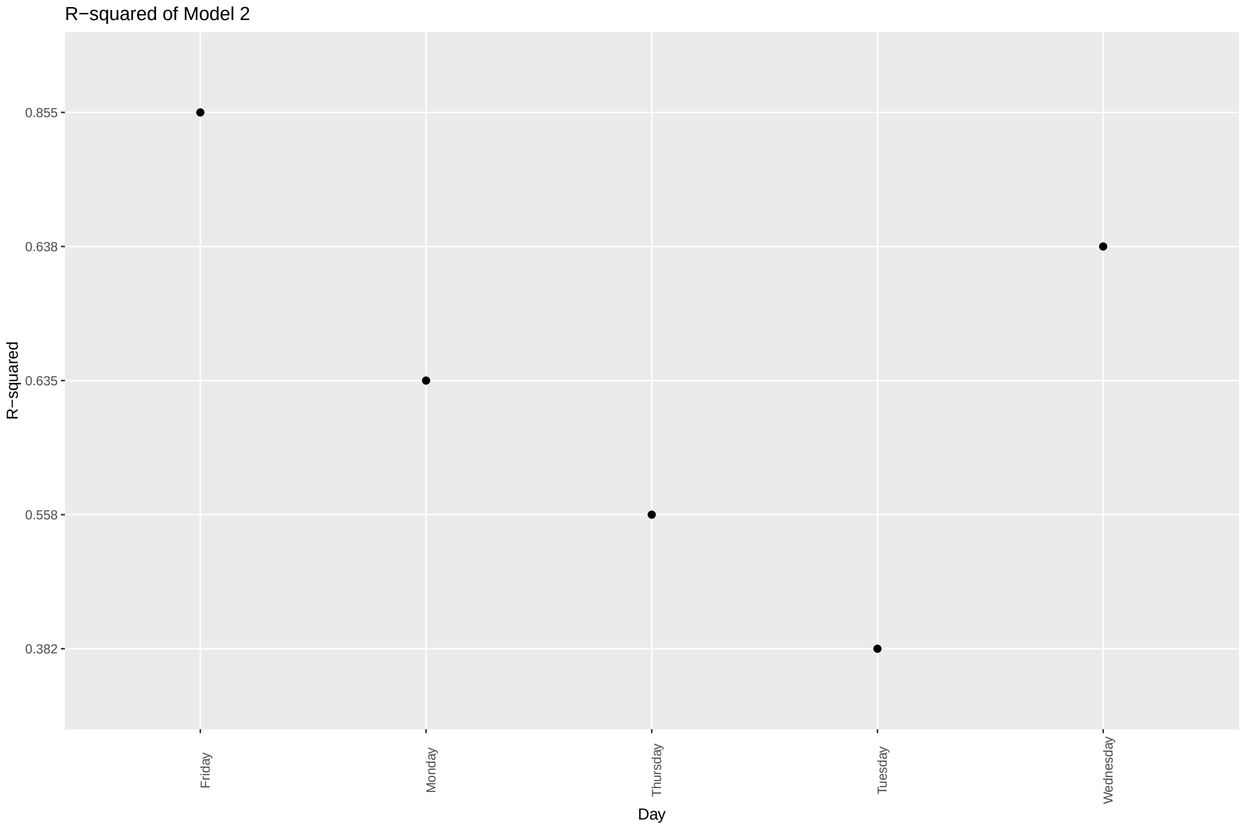
```
for (dow in dows) {
  df_inc_tr_str <- df_inc_train %>% filter(DOW==dow)
  df_inc_tst_str <- df_inc_test %>% filter(DOW==dow)
  n_test <- nrow(df_inc_tst_str)

  # Model 2
  model2 <- lm(Actual ~ `I - 28` + `I - 21` + `I - 14` + `I - 13` + `I - 12` +
                 `I - 11` + `I - 10` + `I - 9` + `I - 8` + `I - 7`,
                 data=df_inc_tr_str)

  predicted2 <- predict(model2, df_inc_tst_str)
  MSE_test2 <- (sum((df_inc_tst_str$Actual - predicted2)^2)/
                  n_test) %>% round(3)
  R22 <- summary(model2)["r.squared"][[1]] %>% round(3)
  df_model2 <- rbind(df_model2, c(dow, MSE_test2, R22))
}
```

```
## Warning in predict.lm(model2, df_inc_tst_str): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(model2, df_inc_tst_str): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(model2, df_inc_tst_str): prediction from a rank-deficient
## fit may be misleading

## Warning in predict.lm(model2, df_inc_tst_str): prediction from a rank-deficient
## fit may be misleading
```

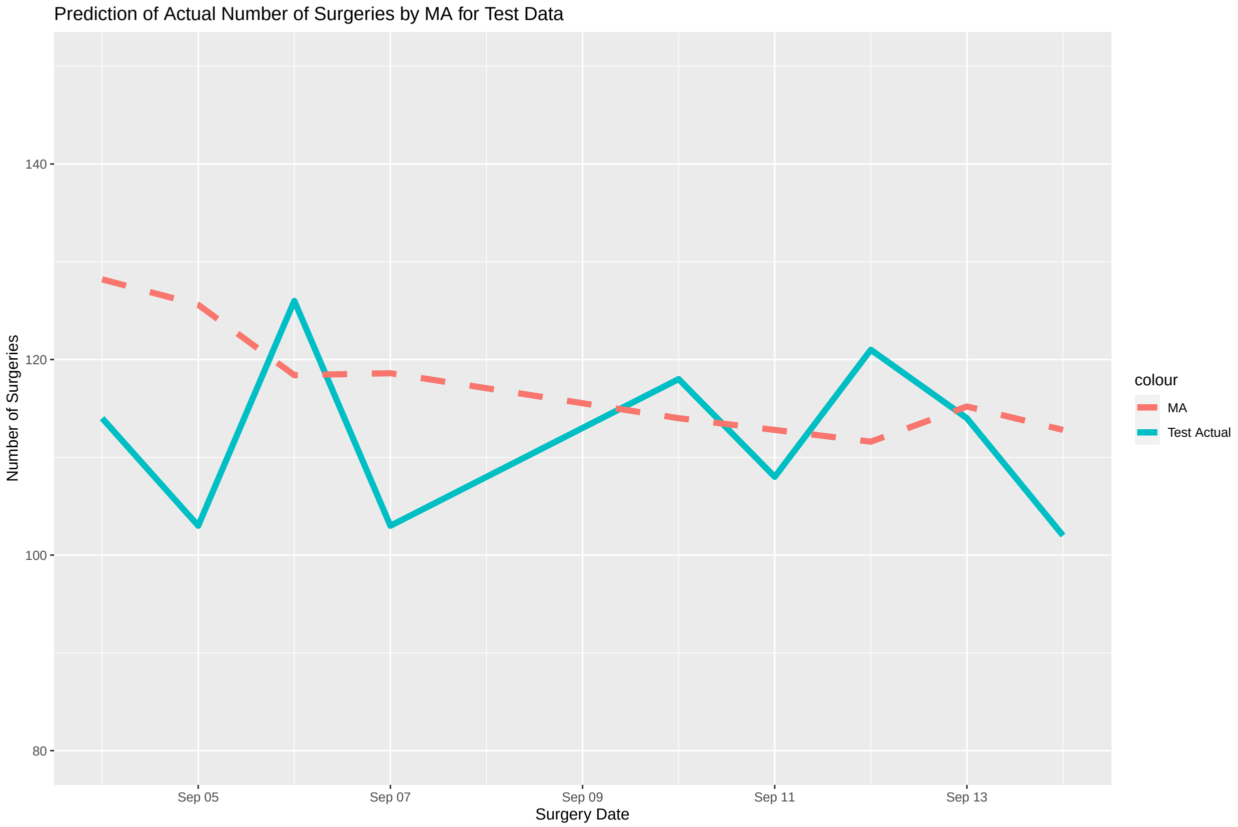
```
df_model2 %>%
  na.omit() %>%
  mutate(MSE_test2=as.numeric(MSE_test2)) %>%
  ggplot(aes(x=DOW, y=MSE_test2)) +
  geom_point(size=2) +
  labs(x="Day", y="MSE of Test Data", title="MSE of Test Data for Model 2") +
  theme(axis.text.x=element_text(angle=90))
```



```
df_model2 %>%
  na.omit() %>%
  mutate(MSE_test2=as.numeric(R22)) %>%
  ggplot(aes(x=DOW, y=R22)) +
  geom_point(size=2) +
  labs(x="Day", y="R-squared", title="R-squared of Model 2") +
  theme(axis.text.x=element_text(angle=90))
```

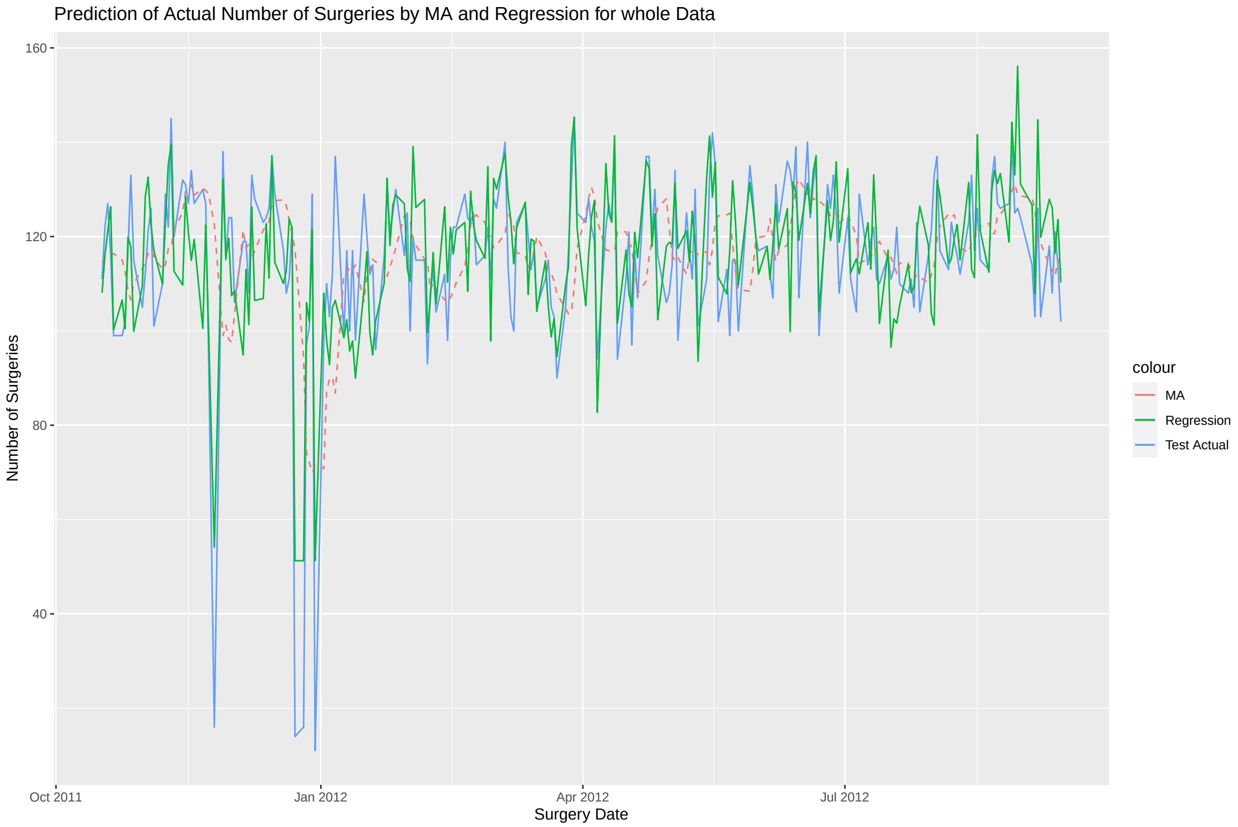


```
n_MA <- 5

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 %>%
  ggplot(aes(x=SurgDate)) +
  geom_line(aes(y=Actual, color="Test Actual"), linewidth=2) +
  geom_line(aes(y=MA, color="MA"), linetype="dashed", linewidth=2) +
  scale_y_continuous(limits=c(80,150)) +
  labs(x="Surgery Date", y="Number of Surgeries",
       title="Prediction of Actual Number of Surgeries by MA for Test Data")
```



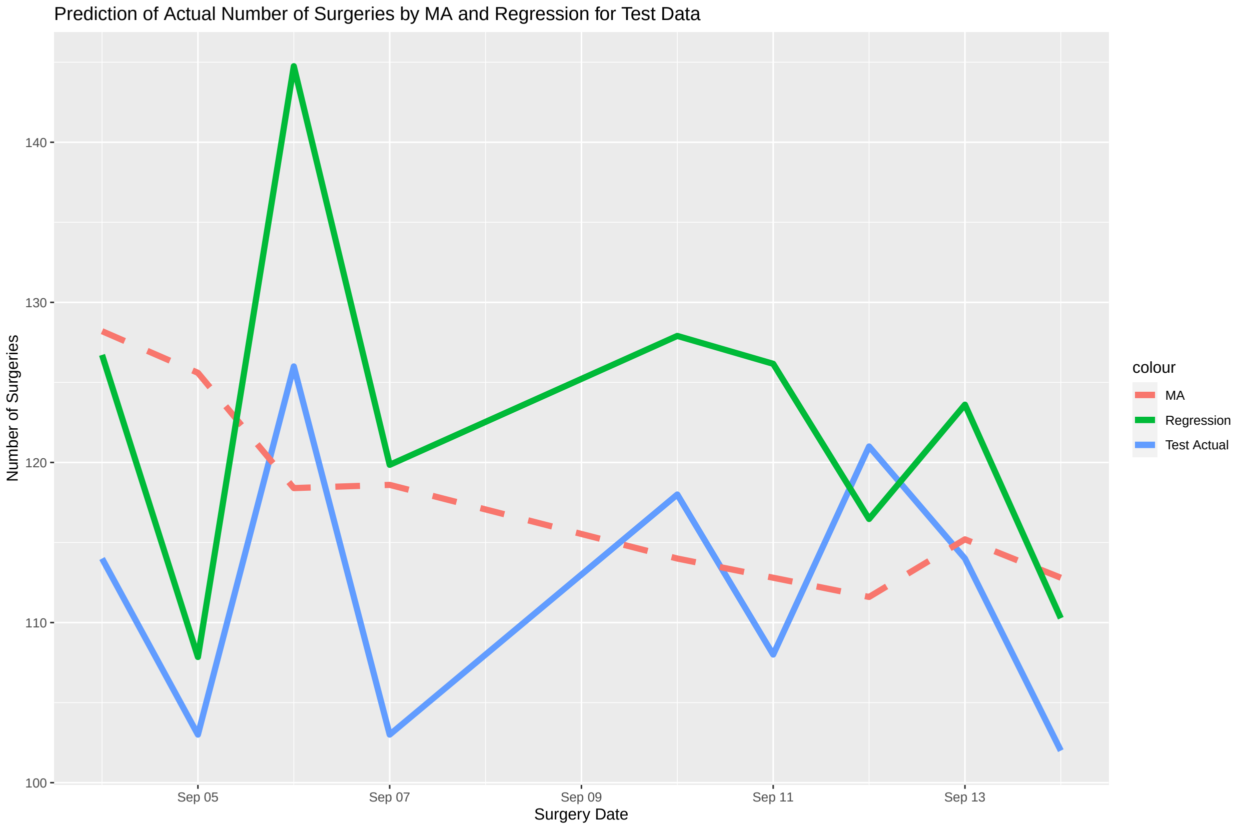
```
df_inc$Model1 <- predict(model1, df_inc)
n_inc <- nrow(df_inc)

df_inc %>%
  ggplot(aes(x=SurgDate)) +
  geom_line(aes(y=Actual, color="Test Actual"), linewidth=0.5) +
  geom_line(aes(y=MA, color="MA"), linetype="dashed", linewidth=0.5) +
  geom_line(aes(y=Model1, color="Regression"), linetyep="dotdash", linewidth=0.5) +
  labs(x="Surgery Date", y="Number of Surgeries",
       title="Prediction of Actual Number of Surgeries by MA and Regression for whole Data")
```

```
## Warning in geom_line(aes(y = Model1, color = "Regression"), linetyep =
## "dotdash", : Ignoring unknown parameters: ‘linetyep‘
```



```
df_test$Model1 <- predict(model1, df_test)
n_test <- nrow(df_test)

df_test %>%
  ggplot(aes(x=SurgDate)) +
  geom_line(aes(y=Actual, color="Test Actual"), linewidth=2) +
  geom_line(aes(y=MA, color="MA"), linetype="dashed", linewidth=2) +
  geom_line(aes(y=Model1, color="Regression"), linetyep="dotdash", linewidth=2) +
  labs(x="Surgery Date", y="Number of Surgeries",
       title="Prediction of Actual Number of Surgeries by MA and Regression for Test Data")
```

```
## Warning in geom_line(aes(y = Model1, color = "Regression"), linetyep =
## "dotdash", : Ignoring unknown parameters: ‘linetyep‘
```



```
MSE_model1 <- (sum((df_test$Actual - df_test$Model1)^2)/
               n_test) %>% round(3)
MSE_MA <- (sum((df_test$Actual - df_test$MA)^2)/
               n_test) %>% round(3)
```