real

Joshua

```
rm(list=ls())
library('knitr')
```

```
## Warning: package 'knitr' was built under R version 3.5.2
```

```
library('kableExtra')
```

```
## Warning: package 'kableExtra' was built under R version 3.5.2
```

```
library('forecast')
```

```
## Warning: package 'forecast' was built under R version 3.5.2
```

```
library('smooth')
```

```
## Warning: package 'smooth' was built under R version 3.5.2
```

```
## Loading required package: greybox
```

```
## Warning: package 'greybox' was built under R version 3.5.2
```

```
## Package "greybox", v0.5.8 loaded.
```

```
## This is package "smooth", v2.5.5
```

```
library('beanplot')
library('pastecs')
library('scales')
library('rbenchmark')
library('ggplot2')
library('readxl')
```

```
## Warning: package 'readxl' was built under R version 3.5.2
```

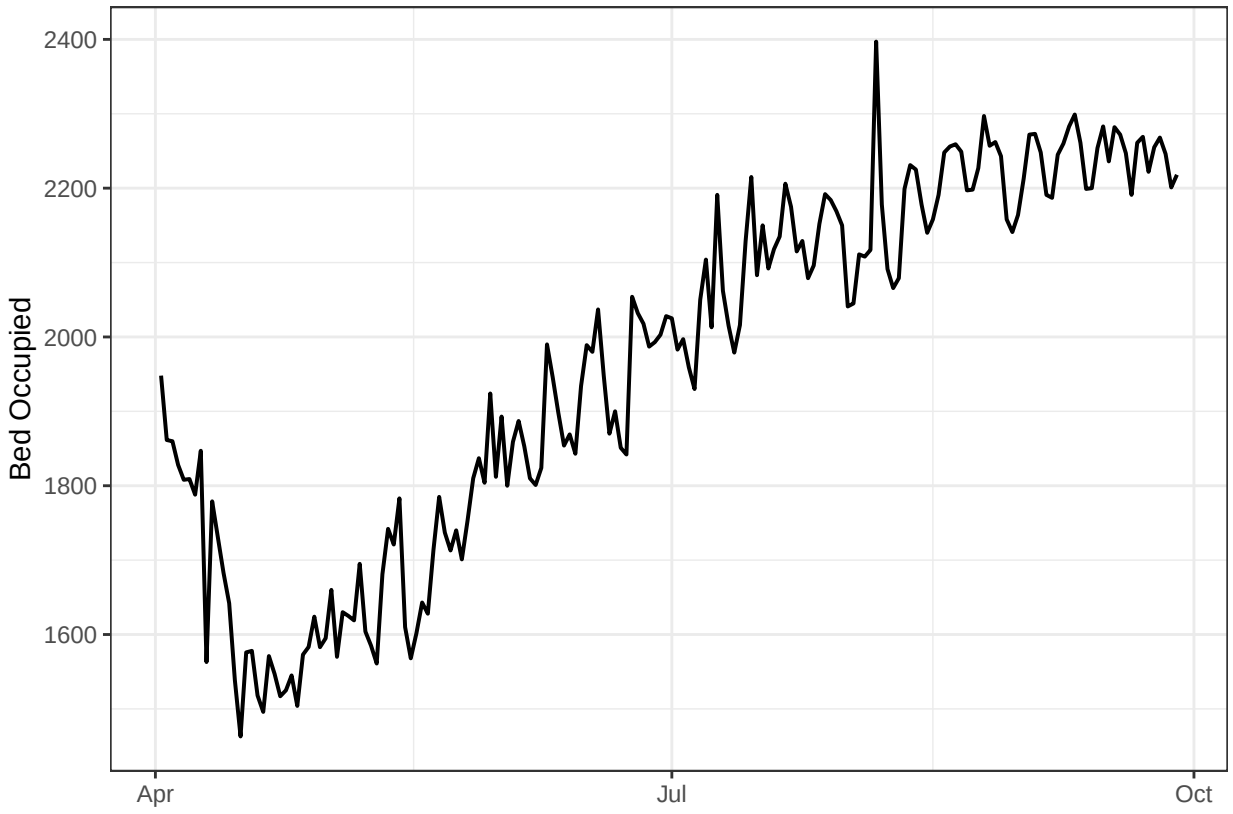
```
source<-read_excel('Covid.xlsx',2,range = cell_rows(151),col_names = FALSE)
```

```
## New names:
## * `` -> ...1
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * ... and 181 more problems
```

```
time_series<-ts(as.numeric(source[4:183]),frequency = 7, start = c(1, 5))
time_series<-tsclean(time_series)
horizon<-100
n<-80
```

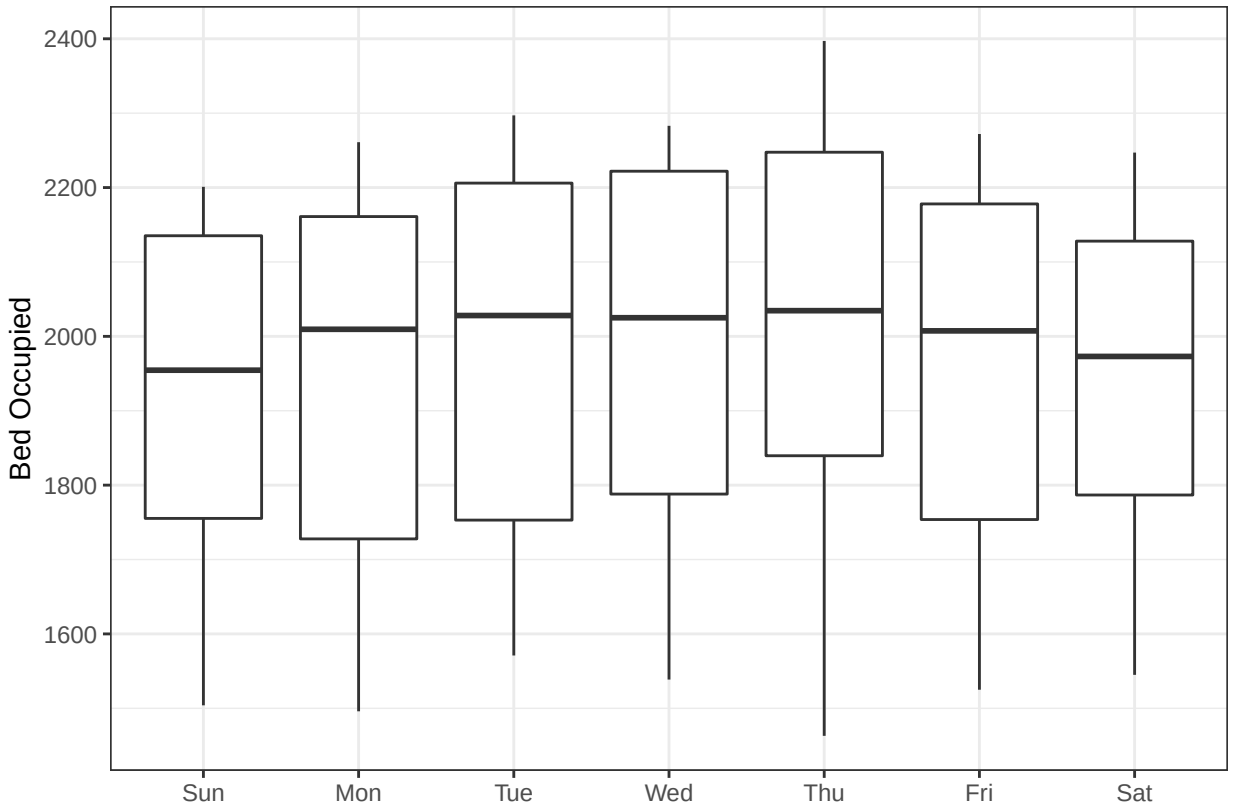
```
ts_plot_season <- function(x = x) {
season <- cycle(x)
```

```
season.factor <- factor(season)
ggplot() +
  geom_boxplot(mapping = aes(x = season.factor,
                             y = x)) +
  labs(x = "", y =  "Bed Occupied") +
  scale_x_discrete(labels=c("1" = "Sun", "2" = "Mon", "3" = "Tue", "4" = "Wed", "5" = "Thu", "6" = "Fri
  theme_bw()
}

data <- data.frame(
  day = as.Date("2020-04-02") + 0:179,
  value = as.numeric(time_series)
)
plot <- ggplot(data, aes(x=day, y=value)) +
  geom_line(size = 0.7) +
  xlab("")
plot + labs(x = "", y =  "Bed Occupied") + theme_bw()
```

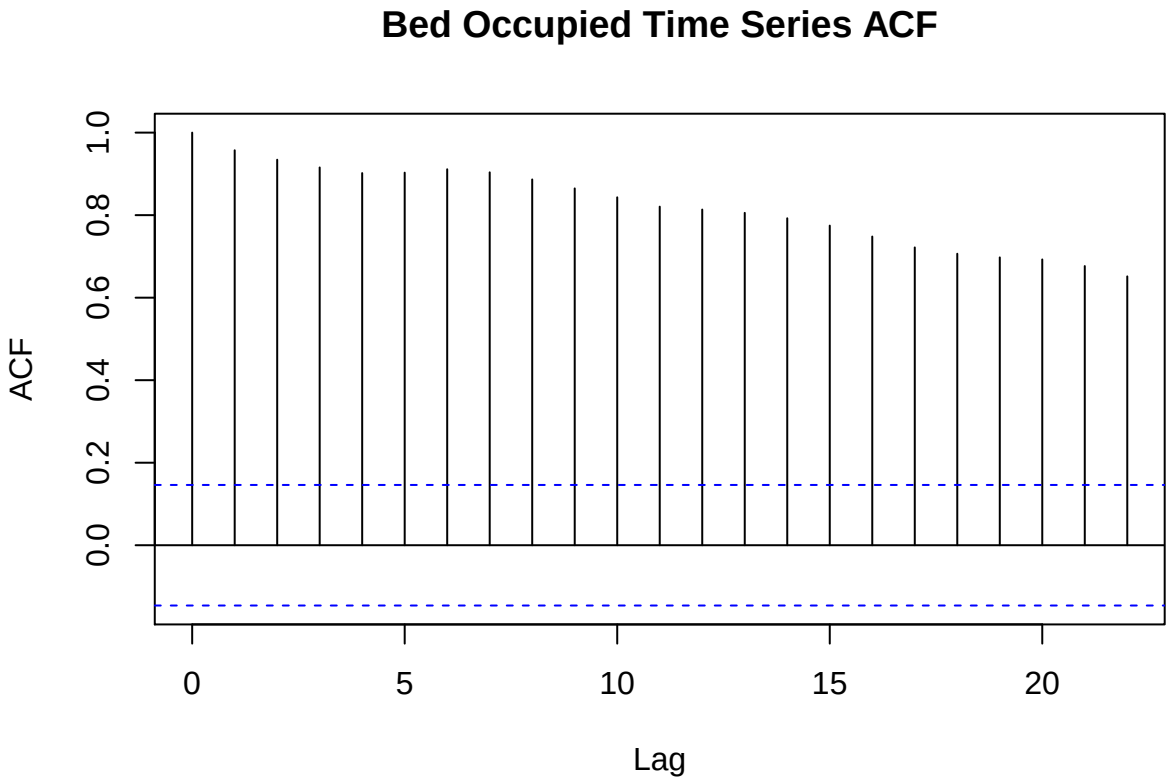



```
ts_plot_season(time_series)
```

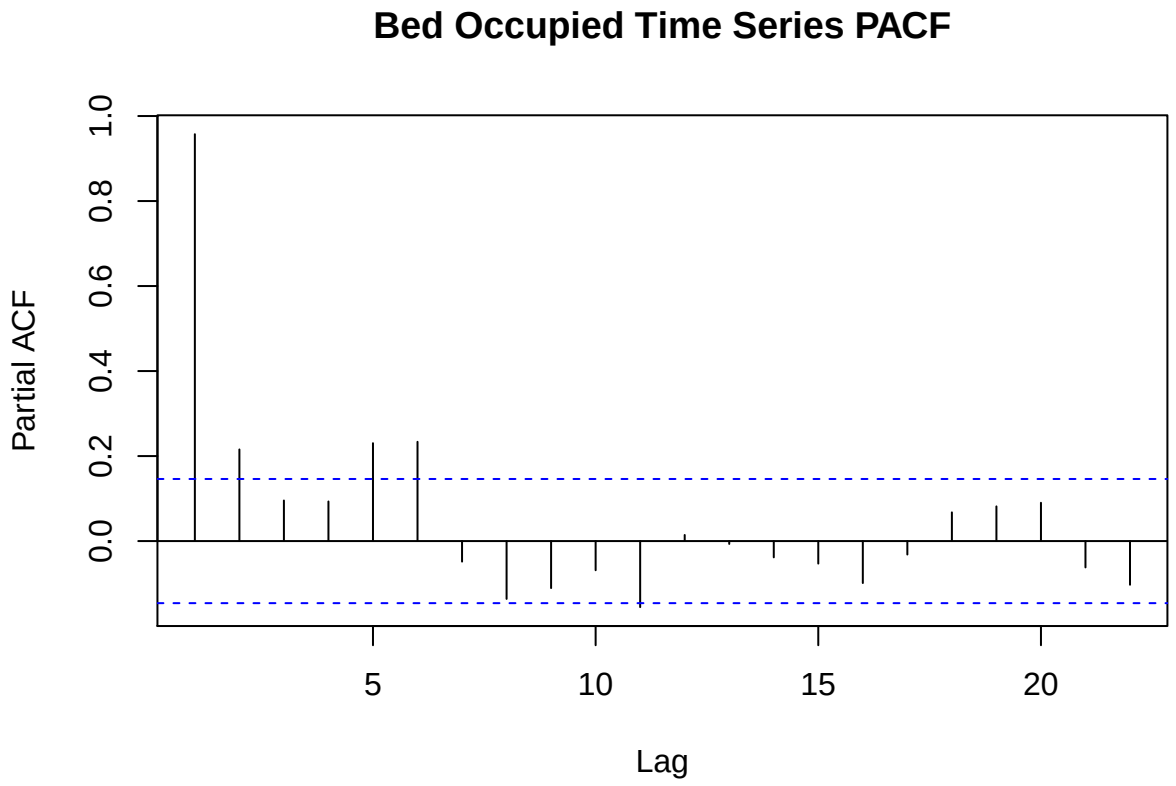
```
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
```




```
ts_acf<-Acf(time_series,plot = FALSE)
ts_pacf<-Pacf(time_series,plot = FALSE)
plot(ts_acf, main = "Bed Occupied Time Series ACF")
```



```r
plot(ts_pacf, main = "Bed Occupied Time Series PACF")
```




```r
cost_function<-function(order,demand){
  u<-runif(100,0,15)
  salvage<-mean((rep((order-demand),100)>=u)*u+(rep((order-demand),100)<u)*rep((order-demand),100))
  over<-10*(order-demand)-4*salvage
  under<-(demand-order)^2
  cost<-(order>=demand)*over+(order<demand)*under
  return(cost)
}
```

```r
art_demand<-rnorm(10000,700,50)
mini_function<-function(par,list_demand){
  record<-c()
  for (i in list_demand) {
    co<-cost_function(par,i)
    record<-c(record,co)
  }
  return(mean(record))
}
quan<-mean(optim(par = 700, fn = mini_function,method = 'L-BFGS-B', list_demand = art_demand)$par>art_d
```

```r
mini_cost_1<-function(data,par){
  order<-par[1]+par[2]*data[,2]
  total_cost<-sum(apply(cbind(order,data[,1]),1,function(x) cost_function(x[1],x[2])))
  return(total_cost)
}

dj_record_1<-c()
```

```
imeo_record_1<-c()
for (i in 1:n){
  train<-ts(time_series[i:(horizon+i-1)]/3,frequency = 7)
  L1_train=lag(train,-1)
  set_train=cbind(train,L1_train)
  colnames(set_train)<-c('train','L1')
  set_train=ts(set_train[8:horizon,])
  dj_fit<-ssarima(train,frequency=7,orders=list(ar=c(1,0)),lags =c(1,7),constant=T)
  dj_value<-forecast(dj_fit,1,interval="parametric",level=quan*2-1)$upper
  dj_record_1<-c(dj_record_1,dj_value)
  coe<-lm(train ~., data=set_train)$coefficients
  imeo_fit<-optim(par = coe, fn = mini_cost_1,method = 'L-BFGS-B', data = set_train)$par
  imeo_value<-imeo_fit[1]+imeo_fit[2]*train[horizon]
  imeo_record_1<-c(imeo_record_1,imeo_value)
}
```

```
mini_cost_2<-function(data,par){
  order<-par[1]+par[2]*data[,2]+par[3]*data[,3]
  total_cost<-sum(apply(cbind(order,data[,1]),1,function(x) cost_function(x[1],x[2])))
  return(total_cost)
}

dj_record_2<-c()
imeo_record_2<-c()
for (i in 1:n){
  train<-ts(time_series[i:(horizon+i-1)]/3,frequency = 7)
  L1_train=lag(train,-1)
  L2_train=lag(train,-2)
  set_train=cbind(train,L1_train,L2_train)
  colnames(set_train)<-c('train','L1','L2')
  set_train=ts(set_train[8:horizon,])
  dj_fit<-ssarima(train,frequency=7,orders=list(ar=c(2,0)),lags =c(1,7),constant=T)
  dj_value<-forecast(dj_fit,1,interval="parametric",level=quan*2-1)$upper
  dj_record_2<-c(dj_record_2,dj_value)
  coe<-lm(train ~., data=set_train)$coefficients
  imeo_fit<-optim(par = coe, fn = mini_cost_2,method = 'L-BFGS-B', data = set_train)$par
  imeo_value<-imeo_fit[1]+imeo_fit[2]*train[horizon]+imeo_fit[3]*train[horizon-1]
  imeo_record_2<-c(imeo_record_2,imeo_value)
}
```

```
mini_cost_3<-function(data,par){
  order<-par[1]+par[2]*data[,2]+par[3]*data[,3]
  total_cost<-sum(apply(cbind(order,data[,1]),1,function(x) cost_function(x[1],x[2])))
  return(total_cost)
}

dj_record_3<-c()
imeo_record_3<-c()
for (i in 1:n){
  train<-ts(time_series[i:(horizon+i-1)]/3,frequency = 7)
  L1_train=lag(train,-1)
  L7_train=lag(train,-7)
  set_train=cbind(train,L1_train,L7_train)
  colnames(set_train)<-c('train','L1','L7')
```

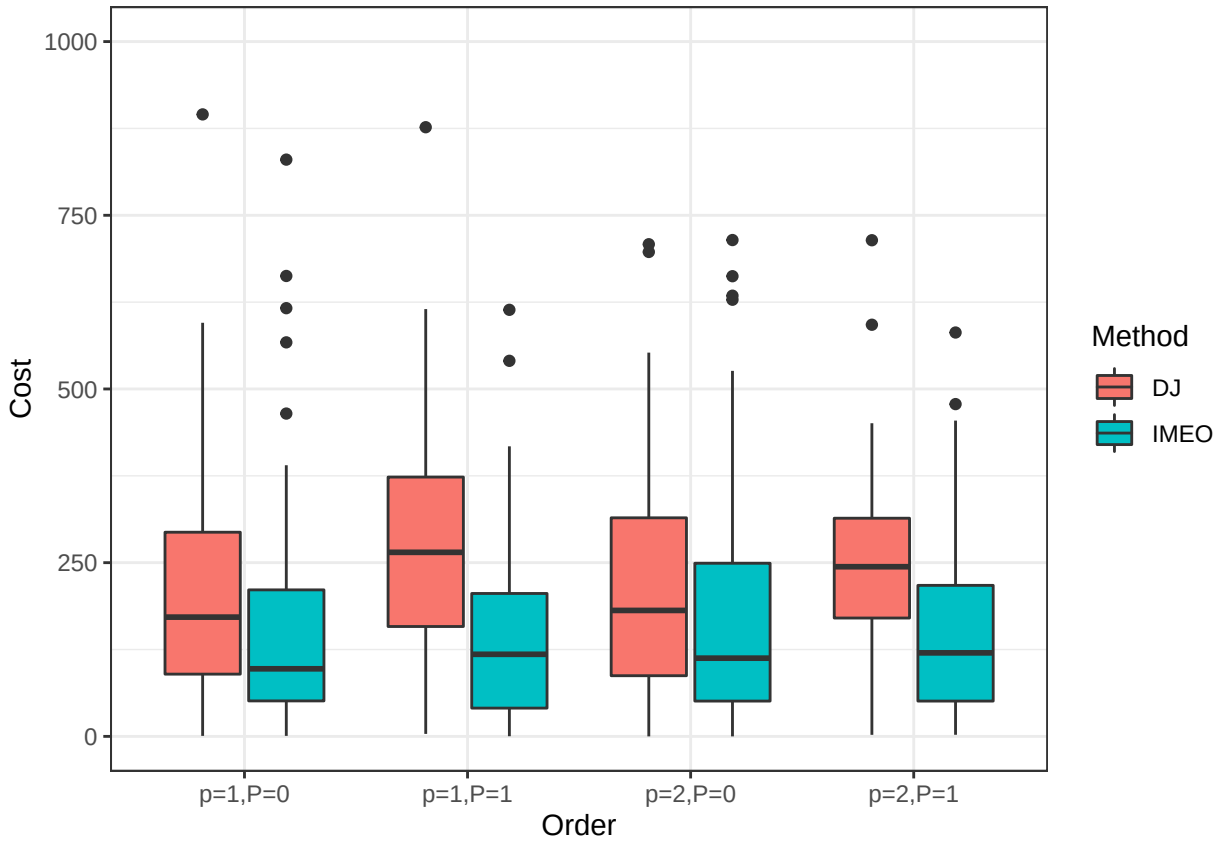
```
  set_train=ts(set_train[8:horizon,])
  dj_fit<-ssarima(train,frequency=7,orders=list(ar=c(1,1)),lags =c(1,7),constant=T)
  dj_value<-forecast(dj_fit,1,interval="parametric",level=quan*2-1)$upper
  dj_record_3<-c(dj_record_3,dj_value)
  coe<-lm(train ~., data=set_train)$coefficients
  imeo_fit<-optim(par = coe, fn = mini_cost_3,method = 'L-BFGS-B', data = set_train)$par
  imeo_value<-imeo_fit[1]+imeo_fit[2]*train[horizon]+imeo_fit[3]*train[horizon-6]
  imeo_record_3<-c(imeo_record_3,imeo_value)
}
```

```
mini_cost_4<-function(data,par){
  order<-par[1]+par[2]*data[,2]+par[3]*data[,3]+par[4]*data[,4]
  total_cost<-sum(apply(cbind(order,data[,1]),1,function(x) cost_function(x[1],x[2])))
  return(total_cost)
}

dj_record_4<-c()
imeo_record_4<-c()
for (i in 1:n){
  train<-ts(time_series[i:(horizon+i-1)]/3,frequency = 7)
  L1_train=lag(train,-1)
  L2_train=lag(train,-2)
  L7_train=lag(train,-7)
  set_train=cbind(train,L1_train,L2_train,L7_train)
  colnames(set_train)<-c('train','L1','L2','L7')
  set_train=ts(set_train[8:horizon,])
  dj_fit<-ssarima(train,frequency=7,orders=list(ar=c(2,1)),lags =c(1,7),constant=T)
  dj_value<-forecast(dj_fit,1,interval="parametric",level=quan*2-1)$upper
  dj_record_4<-c(dj_record_4,dj_value)
  coe<-lm(train ~., data=set_train)$coefficients
  imeo_fit<-optim(par = coe, fn = mini_cost_4,method = 'L-BFGS-B', data = set_train)$par
  imeo_value<-imeo_fit[1]+imeo_fit[2]*train[horizon]+imeo_fit[3]*train[horizon-1]+imeo_fit[4]*train[hor
  imeo_record_4<-c(imeo_record_4,imeo_value)
}
```

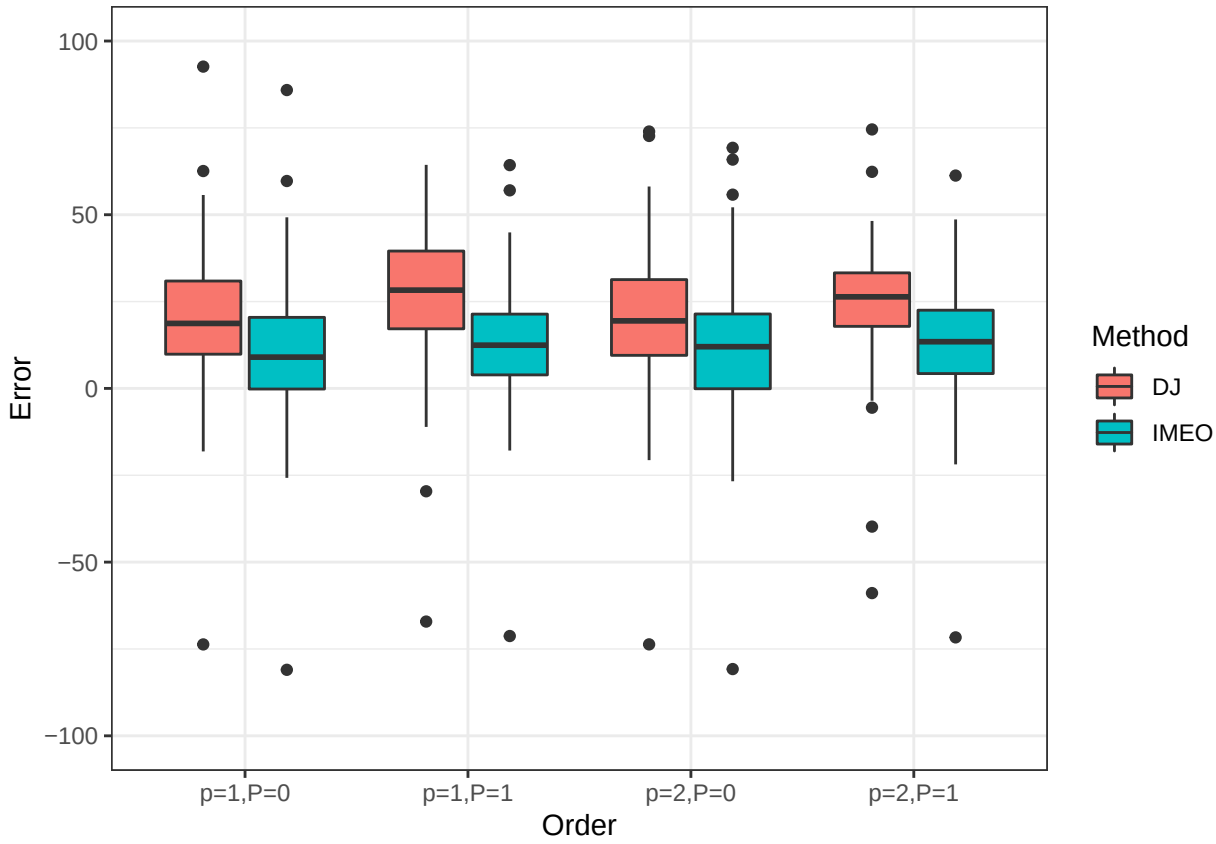
```
test_series<-time_series[(horizon+1):(horizon+n)]/3

return_function<-function(forecast_list,real_list){
  cost<-apply(cbind(forecast_list,real_list),1,function(x) cost_function(x[1],x[2]))
  error<-forecast_list-real_list
  list<-list(cost,error)
  return(list)
}


dj_result_1<-return_function(dj_record_1,test_series)
dj_result_2<-return_function(dj_record_2,test_series)
dj_result_3<-return_function(dj_record_3,test_series)
dj_result_4<-return_function(dj_record_4,test_series)
imeo_result_1<-return_function(imeo_record_1,test_series)
imeo_result_2<-return_function(imeo_record_2,test_series)
imeo_result_3<-return_function(imeo_record_3,test_series)
imeo_result_4<-return_function(imeo_record_4,test_series)
method_app<-c(rep('DJ',4*n),rep('IMEO',4*n))
```

```
lag_length<-c(rep('p=1,P=0',n),rep('p=2,P=0',n),rep('p=1,P=1',n),rep('p=2,P=1',n),
              rep('p=1,P=0',n),rep('p=2,P=0',n),rep('p=1,P=1',n),rep('p=2,P=1',n))
cost_record<-c(dj_result_1[[1]],dj_result_2[[1]],dj_result_3[[1]],dj_result_4[[1]],
               imeo_result_1[[1]],imeo_result_2[[1]],imeo_result_3[[1]],imeo_result_4[[1]])
error_record<-c(dj_result_1[[2]],dj_result_2[[2]],dj_result_3[[2]],dj_result_4[[2]],
               imeo_result_1[[2]],imeo_result_2[[2]],imeo_result_3[[2]],imeo_result_4[[2]])
df<-data.frame(Method=method_app,lag_length=lag_length,cost_record=cost_record,error_record=error_recor
df$lag_length <- as.factor(df$lag_length)
```

```
ggplot(df, aes(x=lag_length, y=cost_record, fill=Method)) +
  coord_cartesian(ylim = c(0, 1000)) +
  geom_boxplot(aes(middle = mean(cost_record))) + theme_bw() +
  labs(x = "Order", y =  "Cost")
```




```
ggplot(df, aes(x=lag_length, y=error_record, fill=Method)) +
  coord_cartesian(ylim = c(-100, 100)) +
  geom_boxplot(aes(middle = mean(error_record))) + theme_bw() +
  labs(x = "Order", y =  "Error")
```


100
50
0
−50
−100
Error
p=1,P=0
p=1,P=1
p=2,P=0
p=2,P=1
Order
Method
DJ
IMEO