```
##Prophet model###########
# Load the desired package
library(prophet)
library(dplyr)
data <- read_excel("C:/Users/User/Desktop/ARIMA--X/flu1.xlsx")
# Generate some sample time series data

time_series_data <- data.frame(time = data$Date, value = data$cases)
head(time_series_data)
# Convert the time series data to the data box format required for Prophetdf <- data.frame(ds =
as.Date(data$Date), y = data$cases)
# The time series data were split into the training and test sets
train_data <- filter(df, ds <= as.Date("2021-12-31"))
test_data <- filter(df, ds >= as.Date("2022-01-01"))

# Construct the Prophet model##########
model <- prophet(train_data)

##Build models with seasonal trends
model <- prophet (train_data,growth = "linear",
                  yearly.seasonality = TRUE, weekly.seasonality = FALSE,
                  daily.seasonality = FALSE, seasonality.mode = "multiplicative")
##Predict the data for the next 1 year, and visualize the prediction results
future <- make_future_dataframe(model, periods = 12,freq = "month")
forecast <- predict(model, future)
plot(model, forecast)

# Fit to the training set
model_fit <- predict(model)
# Predictions on the test set
future <- make_future_dataframe(model, periods = nrow(test_data))
test_predictions <- predict(model, future)

# The prediction results of the training and test sets were extracted
train_predictions <- tail(model_fit$yhat, nrow(train_data))
write.xlsx(train_predictions,    file    ="C:/Users/User/Desktop/ARIMA--X/2013-2021/prophet/The
training set was fitted to the value.xlsx")
# count MSE
mse <- mean((train_data$y - train_predictions)^2)
# count MAE
mae <- mean(abs(train_data$y - train_predictions))
# count RMSE
rmse <- sqrt(mean((train_data$y - train_predictions)^2))
# Print out the evaluation index value
```

```
print(paste("MSE:", mse))
print(paste("MAE:", mae))
print(paste("RMSE:", rmse))

data <- data.frame(observed = train_data$y, predicted = train_predictions)
# By specifying type = "all", we can calculate the confidence intervals for the MSE, MAE, and RMSE simultaneously
boot_MSE <- boot(data, calc_MSE, R = 1000)
boot_MAE <- boot(data, calc_MAE, R = 1000)
boot_RMSE <- boot(data, calc_RMSE, R = 1000)

# Output results
boot_MSE_ci <- boot.ci(boot_MSE, type = "all", conf = 0.95)
boot_MAE_ci <- boot.ci(boot_MAE, type = "all", conf = 0.95)
boot_RMSE_ci <- boot.ci(boot_RMSE, type = "all", conf = 0.95)

print("95% Confidence Interval for MSE:")
print(boot_MSE_ci)
print("95% Confidence Interval for MAE:")
print(boot_MAE_ci)
print("95% Confidence Interval for RMSE:")
print(boot_RMSE_ci)

test_predictions <- tail(test_predictions$yhat, nrow(test_data))
write.xlsx(test_predictions, file ="C:/Users/User/Desktop/ARIMA--X/2013-2021/prophet/The test set fits the value.xlsx")
# count MSE
mse1 <- mean((test_data$y - test_predictions)^2)
# count MAE
mae1 <- mean(abs(test_data$y - test_predictions))
# count RMSE
rmse1 <- sqrt(mean((test_data$y - test_predictions)^2))
# Print out the evaluation index value
print(paste("MSE:", mse1))
print(paste("MAE:", mae1))
print(paste("RMSE:", rmse1))

data <- data.frame(observed = test_data$y, predicted = test_predictions)
# By specifying type = "all", we can calculate the confidence intervals for the MSE, MAE, and RMSE simultaneously
boot_MSE <- boot(data, calc_MSE, R = 1000)
boot_MAE <- boot(data, calc_MAE, R = 1000)
boot_RMSE <- boot(data, calc_RMSE, R = 1000)
```

```
# Output results
boot_MSE_ci <- boot.ci(boot_MSE, type = "all", conf = 0.95)
boot_MAE_ci <- boot.ci(boot_MAE, type = "all", conf = 0.95)
boot_RMSE_ci <- boot.ci(boot_RMSE, type = "all", conf = 0.95)

print("95% Confidence Interval for MSE:")
print(boot_MSE_ci)
print("95% Confidence Interval for MAE:")
print(boot_MAE_ci)
print("95% Confidence Interval for RMSE:")
print(boot_RMSE_ci)

# The prediction results from the training and test sets were combined
all_predictions <- c(train_predictions, test_predictions)

# Plot of the prediction results
plot(df$ds, df$y, main="Time-series prediction of influenza onset numbers", ylab=" incidence ",
xlab=" time")
lines(df$ds, all_predictions, col="red")

#Components of the visual prediction, with mainly linear and seasonal trends
prophet_plot_components(model, forecast)

Truth<- ts(df$y, start = c(2013, 1), frequency = 12)
train_predictions<- ts(train_predictions, start = c(2013, 1), frequency = 12)
test_predictions<- ts(test_predictions, start = c(2022, 1), frequency = 12)
plot(Truth, xlim = c(2013, 2023), ylim = c(0, 2500), main = "Forecasts from Prophet ",
        ylab="Influenza cases",xlab="")
axis(1, at = seq(2013, 2022, 1), labels = seq(2013, 2022, 1))
lines(Truth,pch=1, lty=1,col = "grey", lwd = 2)    #    true value
lines(train_predictions,lty=1,col = "yellow", lwd = 2)    # The training set was fitted to the value
lines(test_predictions,lty=6,col = "yellow", lwd = 2)    # Forecast results
legend("topleft", legend = c("Actual", "Prophet fitted", "Prophet forecast"), col = c("grey",
"yellow", "yellow"),lty=c(1, 1, 6), lwd = 3)
```