Semi-Mechanistic Modeling and Machine Learning Models for Forecasting Measles Incidence

Wyatt Madden

Introduction

In this tutorial, we will forecast measles incidence using both the semi-mechanistic TSIR model and the LASSO machine learning model. We will employ the seminal England and Wales pre-vaccination bi-weekly measles dataset (a description of which can be found here).

This tutorial employs methods developed in [this paper] (https://doi.org/10.1371/journal.pcbi.1010251)), by Lau et al..

Data

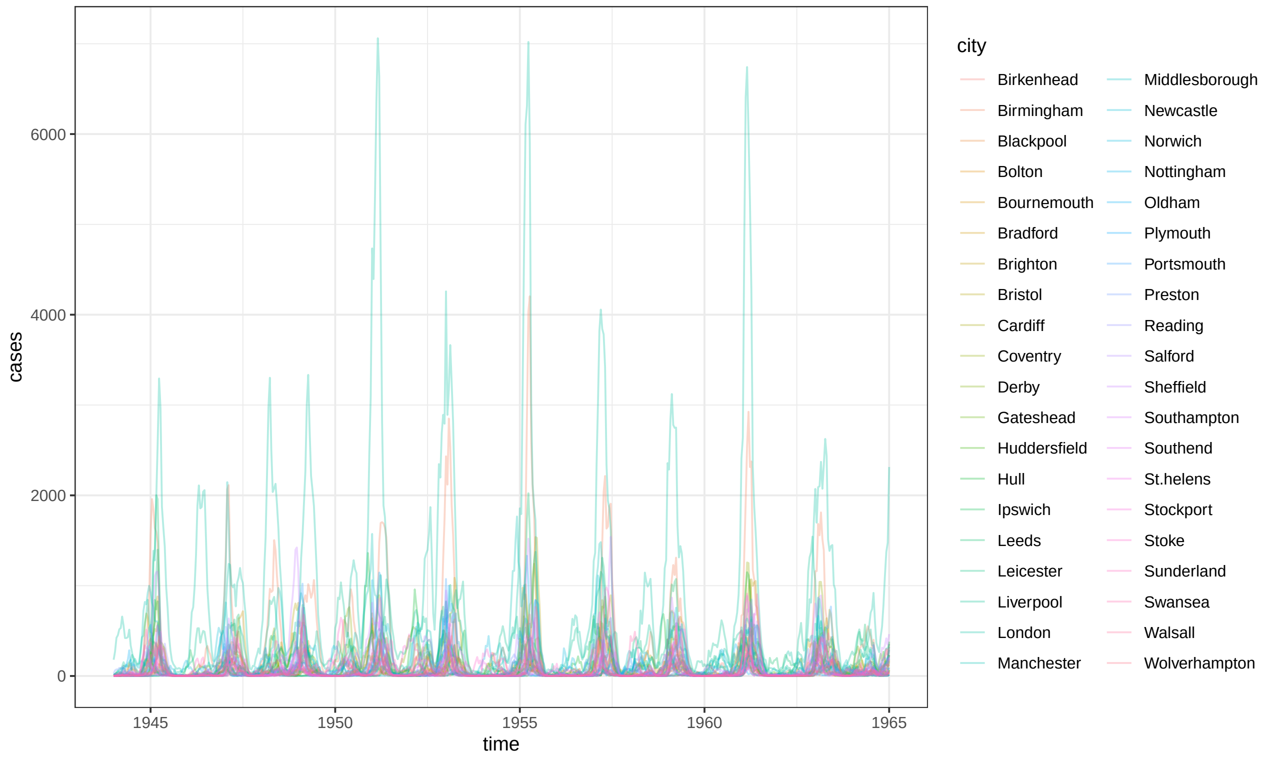
Next, we read-in and inspect the measles dataset.

```
url_loc <- paste0("https://raw.githubusercontent.com/WyattGMadden/",
                  "intro_to_ml_for_id_emory/main/data/england_and_wales_measles/measles.csv")
measles <- read_csv(url_loc)
head(measles)
```

```
## # A tibble: 6 x 7
##    time cases births     pop city         lat   lon
##   <dbl> <dbl>  <dbl>   <dbl> <chr>      <dbl> <dbl>
## 1 1944      1  106.  118626. Birkenhead  53.4 -3.04
## 2 1944.     0  104.  118458. Birkenhead  53.4 -3.04
## 3 1944.     0  103.  118319. Birkenhead  53.4 -3.04
## 4 1944.     0  101.  118233. Birkenhead  53.4 -3.04
## 5 1944.     1  100.  118165. Birkenhead  53.4 -3.04
## 6 1944.     1   99.1 118096. Birkenhead  53.4 -3.04
```

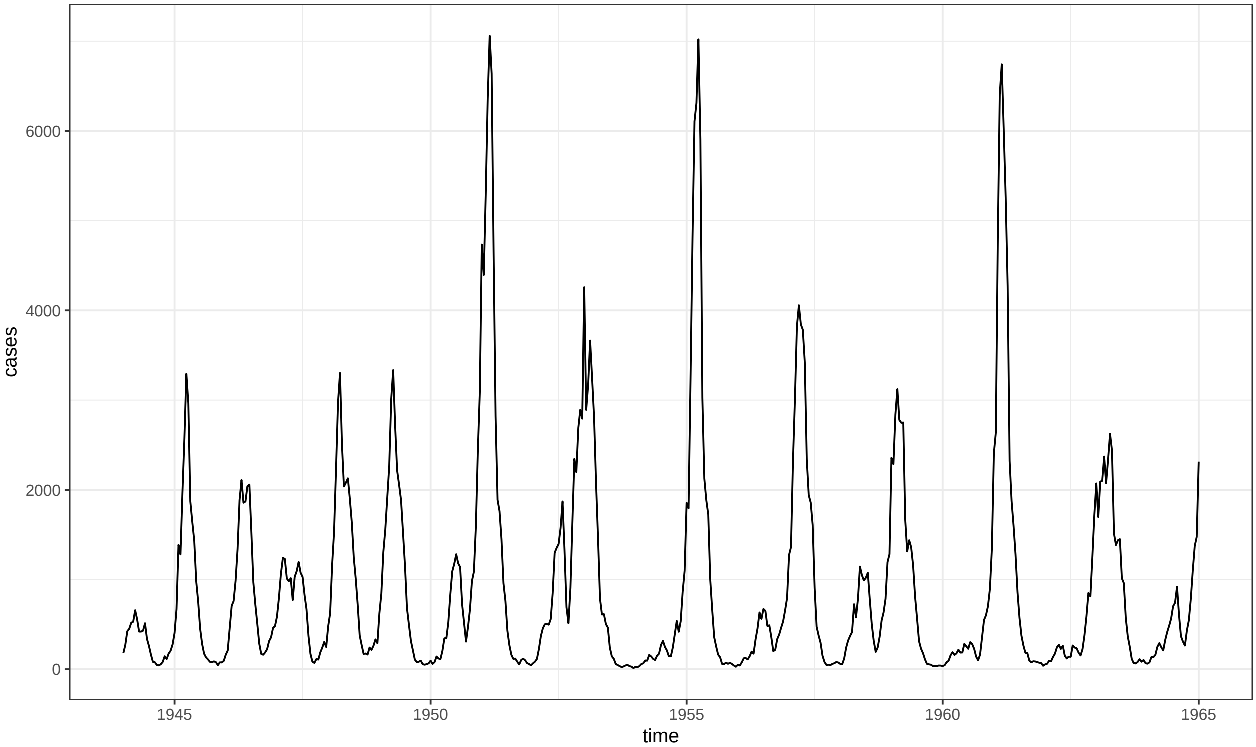
This dataset contains measles incidence in forty cities from 1944-1965. Let us plot all the data prior to fitting models.

```
measles |>
    ggplot(aes(x = time, y = cases, color = city)) +
    geom_line(alpha = 0.3)
```



For now, we will focus on just London measles incidence.

```
london_measles <- measles |>
    filter(city == "London")

london_measles |>
    ggplot(aes(x = time, y = cases)) +
    geom_line()
```

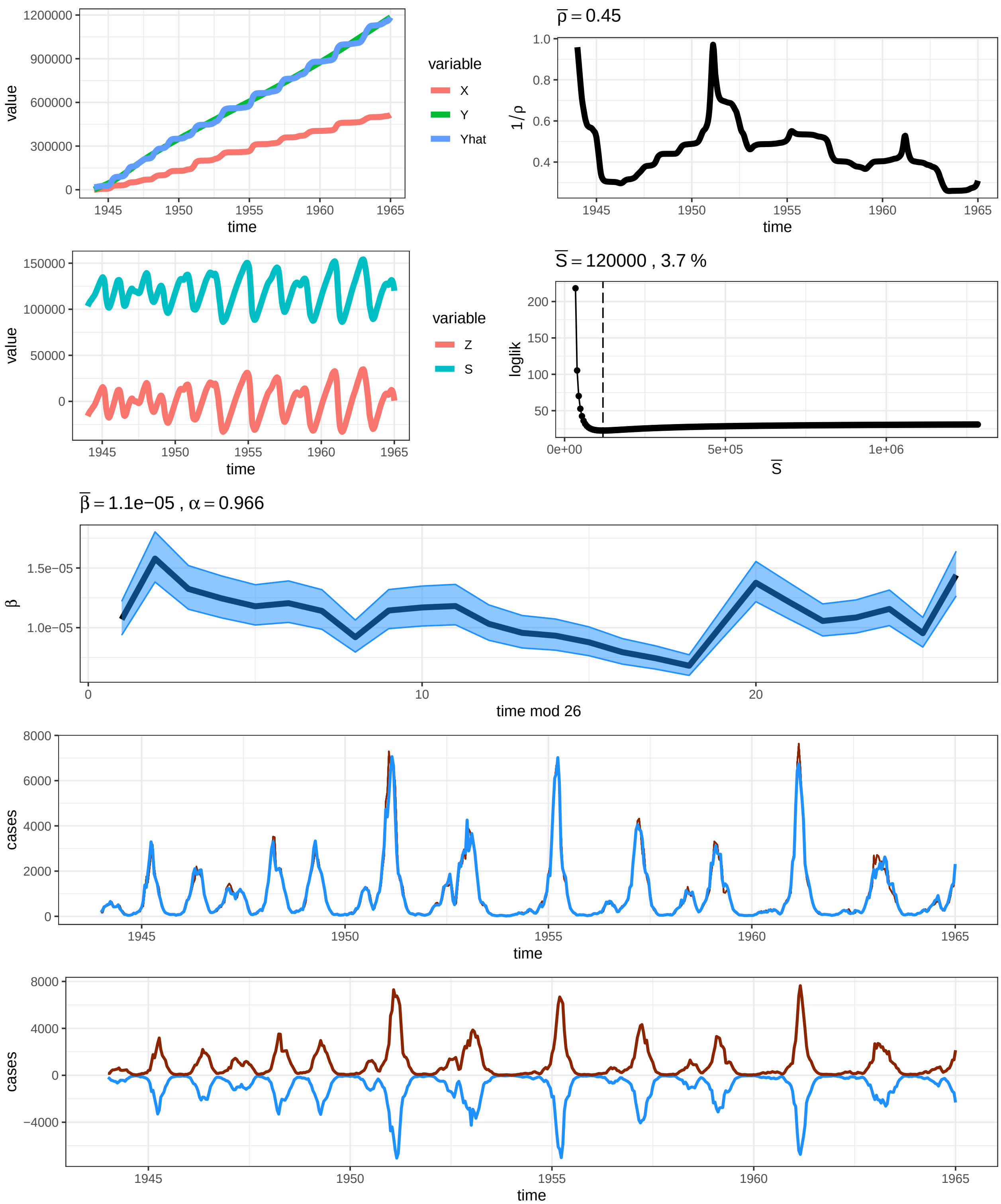


TSIR

Next, we fit the TSIR model to the full London measles dataset. A paper accompanying the tsiR package can be found here.

```
london_tsir <- runtsir(
    data = london_measles,
    IP = 2,
    xreg = 'cumcases',
    regtype = 'gaussian',
    alpha = NULL,
    sbar = NULL,
    family = 'gaussian',
    link = 'identity',
    method = 'deterministic',
    pred = 'step-ahead'
)

plotres(london_tsir)
```

```
## Warning in geom_line(data = dat$res, aes_string(x = "time", y = "cases", : Ignoring unknown aestheti
```



Exercises

1. Read the help file of the `runtsir` function (`?runtsir`) and explore different fitting options. Compare the summary plots.
2. Fit the tsir model in measles incidence for a different city.

Next, we will generate one-step-ahead forecasts using the TSIR model. We will use a non-seasonal beta estimation to simplify forecasting. We again fit the TSIR model:

```
london_tsir_est <- runtsir(
    data = london_measles,
    IP = 2,
    xreg = 'cumcases',
    regtype = 'gaussian',
    alpha = NULL,
    sbar = NULL,
    family = 'gaussian',
    seasonality = "none",
    link = 'identity',
    method = 'deterministic',
    pred = 'step-ahead',
    nsim = 1
)
```

The TSIR updating equations are described as follows:

$$S_{t+1} = B_{t+1} - S_t - I_{t+1}$$

$$E[I_{t+1}] = \beta S_t I_t^{\alpha}$$

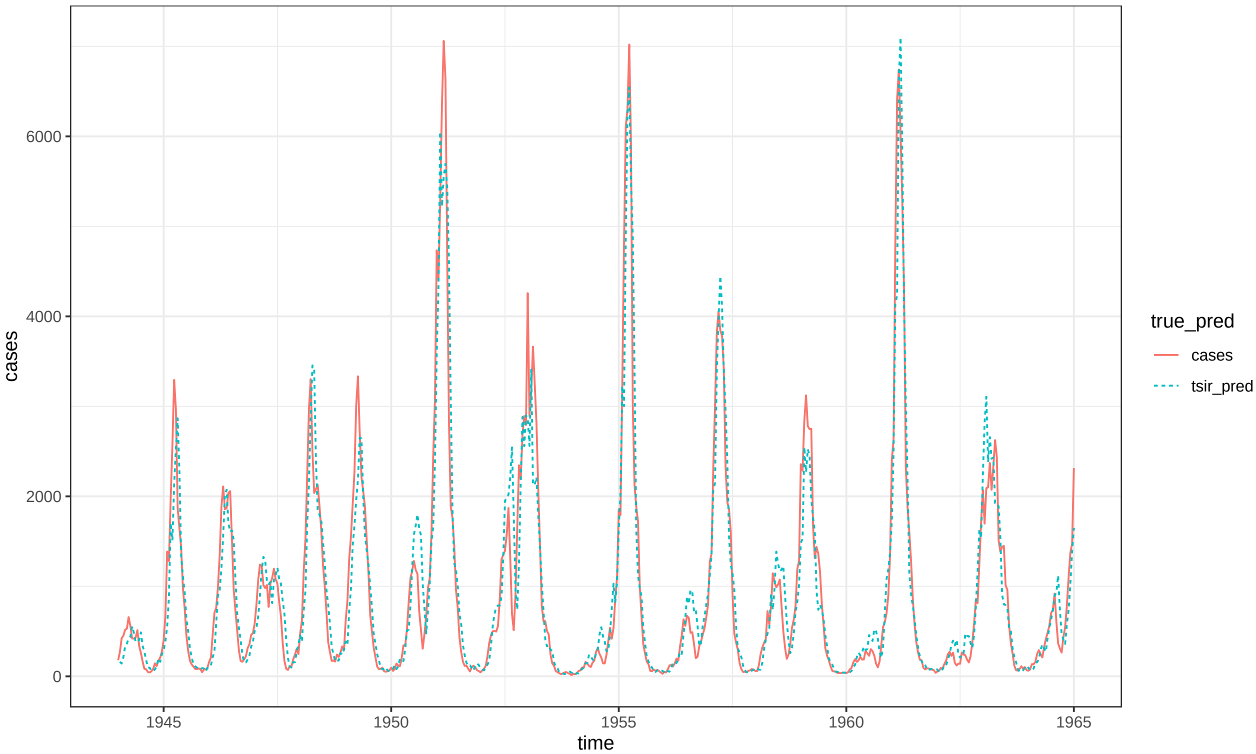
where S are the reconstructed susceptibles provided by the TSIR model, B are the births, I is the true case number ($I \times$ reporting rate $=$ incidence), and β and α are estimated parameters.

We first find the values of interest from the model output:

```
# Values of interest from model output
beta_est <- unique(london_tsir_est$beta)
alpha_est <- london_tsir_est$alpha
rho_est <- london_tsir_est$rho
S_est <- london_tsir_est$simS[, "mean"]
incidence_est <- london_tsir_est$res$mean
I_est <- incidence_est * rho_est

# Empty vectors for forecasts
S_one_step_ahead <- rep(NA, nrow(london_measles))
I_one_step_ahead <- rep(NA, nrow(london_measles))
cases_one_step_ahead <- rep(NA, nrow(london_measles))

# One-year-lagged births vector
births_for_pred <- lag(london_measles$births, 26)

# Calculate one-step-ahead forecasts
for (i in 2:nrow(london_measles)) {
    I_one_step_ahead[i] <- beta_est * S_est[i - 1] * (I_est[i - 1])^alpha_est
    cases_one_step_ahead[i] <- I_one_step_ahead[i] / rho_est[i - 1]
    S_one_step_ahead[i] <- births_for_pred[i] + S_est[i - 1] - I_one_step_ahead[i]
}

london_tsir_pred <- london_measles
london_tsir_pred$tsir_pred <- cases_one_step_ahead
```

```r
london_tsir_pred |>
    pivot_longer(c("cases", "tsir_pred"), names_to = "true_pred", values_to = "cases") |>
    ggplot(aes(x = time, y = cases, color = true_pred, linetype = true_pred)) +
    geom_line()
```

```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```



Exercises

1. Try forecast two (or more) steps ahead.
2. Attempt to forecast on a city other than London.
3. How might you account for seasonal beta terms in the forecast?

LASSO

Next, we fit the LASSO machine learning model to the same London measles data. The LASSO is a flexible regression model that is often more performant than mechanistic/semi-mechanistic models. Rather than making iterative step-ahead predictions, we instead include lagged measles incidence values as features. We fit the regression model on a training data set (the first 70% of the data in this case), and assess performance on the remainder of the data.

The model is written as follows:

$$log(E(I_{i,t+k})) = \eta + \sum_{J=1}^{T_{lag}} \Psi_J log(I_{i,t-J+1}) + \gamma log(\bar{B}_{i,(t-T_{lag}):t} + 1)$$

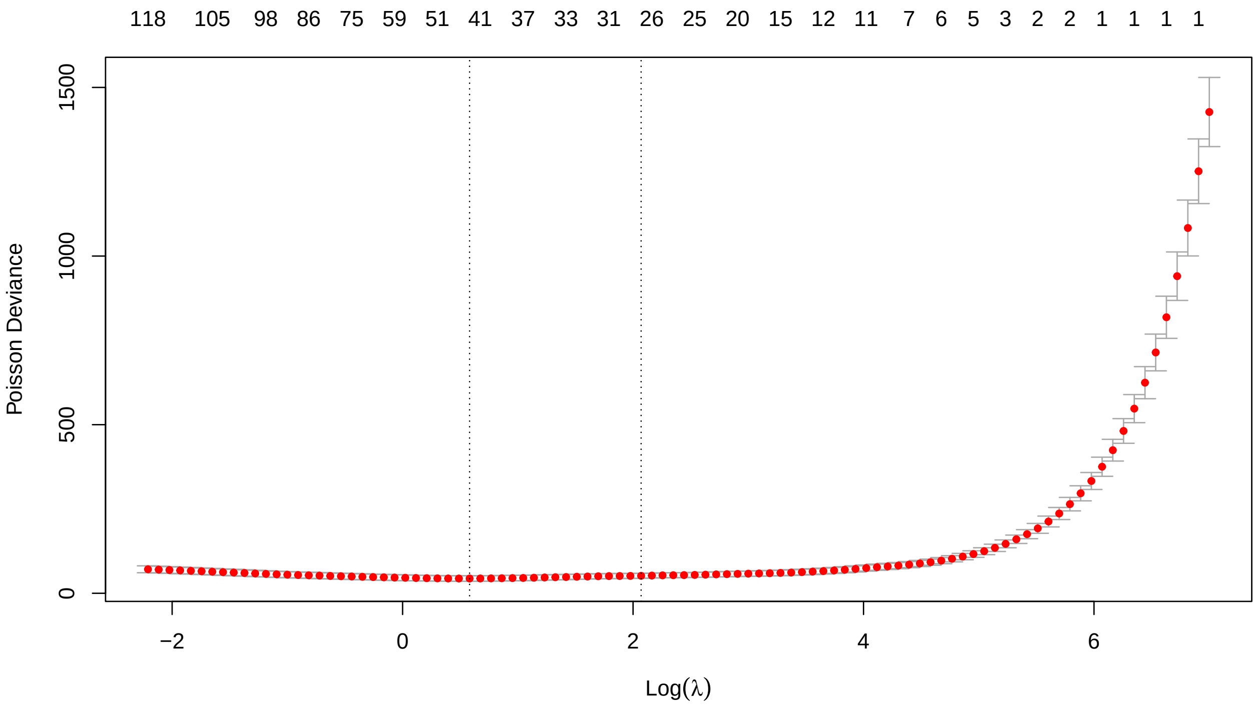
Here we are regressing lagged log-incidence (with lags ranging from k steps prior to T_{lag} steps prior) and the mean births over the lags, on the current-step log-incidence. Further explanation can be found here.

We start by setting T_{lag} to 130, creating the lag-incidence features and the mean birth feature.

```r
# Initialize data and T_lag
lasso_data <- london_measles
T_lag <- 130

lasso_data$log_cases <- log(lasso_data$cases)

# Get case lags
for (i in 1:T_lag) {
    lasso_data[, paste0("log_cases_lag", i)] <- lag(lasso_data$log_cases, i)
}

# Get mean births
lasso_data$log_mean_births_lag <- rep(NA, nrow(lasso_data))
for (i in (T_lag + 1):nrow(lasso_data)) {
    lasso_data$log_mean_births_lag[i] <- log(mean(lasso_data$births[(i - T_lag):(i - 1)]) + 1)
}

# Remove first 130 data rows with NA lag incidence values
lasso_data_full <- lasso_data[!is.na(lasso_data$log_mean_births_lag), ]

# Divide our dataset into training and testing
train_set_proportion <- 0.7
train_set_size <- round(nrow(lasso_data_full) * train_set_proportion)
lasso_data_train <- lasso_data_full[1:train_set_size, ]
lasso_data_test <- lasso_data_full[(train_set_size + 1):nrow(lasso_data_full), ]

Y_train <- lasso_data_train$cases
X_col_names <- c(grep("log_cases_lag", names(lasso_data), value = T), "log_mean_births_lag")
X_train <- as.matrix(lasso_data_train[, X_col_names])
```

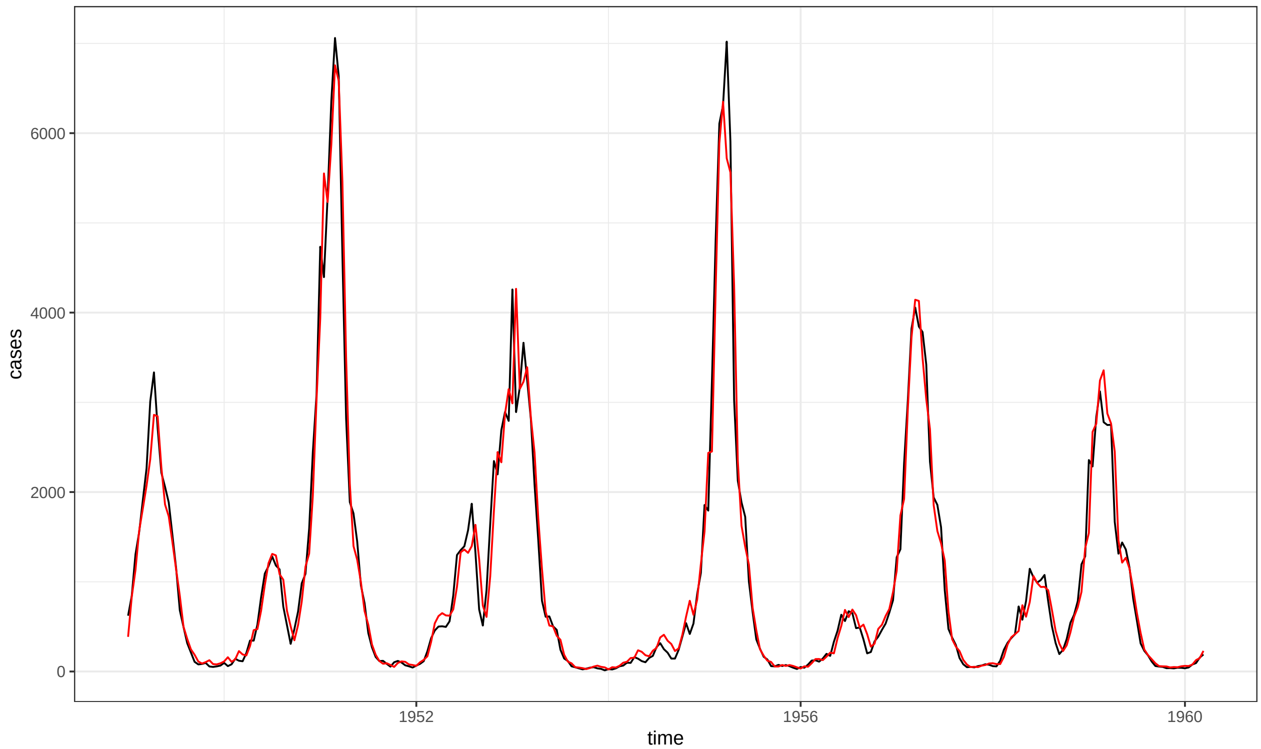
Lasso uses a penalized L1 loss term in addition to the standard MSE loss: $\lambda \sum_{k=1}^{p} |\theta_k|$, where λ is a hyperparameter. Here we select λ using cross-validation:

```r
# Estimate lambda
cv.lasso.oneahead <- cv.glmnet(
    X_train,
    Y_train,
    alpha = 1,
    lower.limits = -Inf,
    family = "poisson",
    intercept = T
)

plot(cv.lasso.oneahead)
```



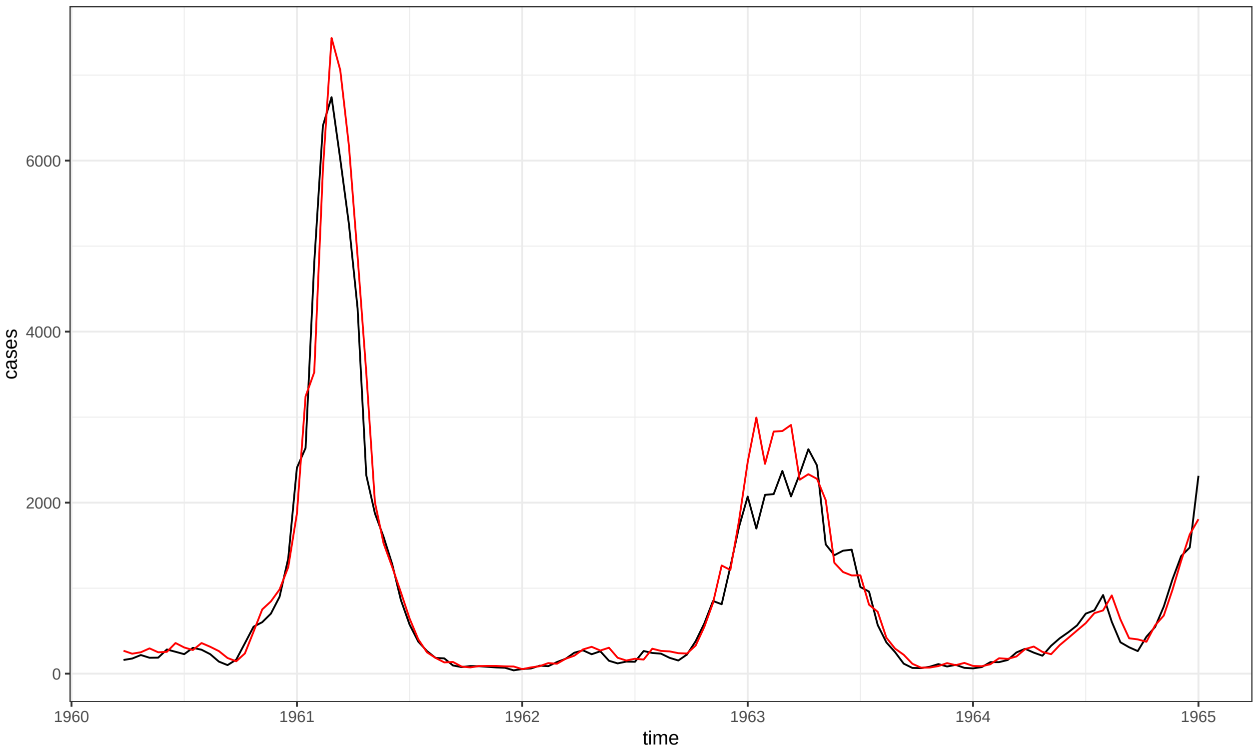
Now we are ready to fit the LASSO on the training dataset.

```
model.oneahead <- glmnet(
    X_train,
    Y_train,
    alpha = 1,
    family = "poisson",
    lambda = cv.lasso.oneahead$lambda.1se
)

# Make predictions on the training set
train_pred <- predict(
    model.oneahead,
    newx = X_train,
    type = "response"
)

lasso_data_train$pred <- train_pred

lasso_data_train |>
    ggplot(aes(x = time)) +
    geom_line(aes(y = cases)) +
    geom_line(aes(y = pred), colour = "red")
```



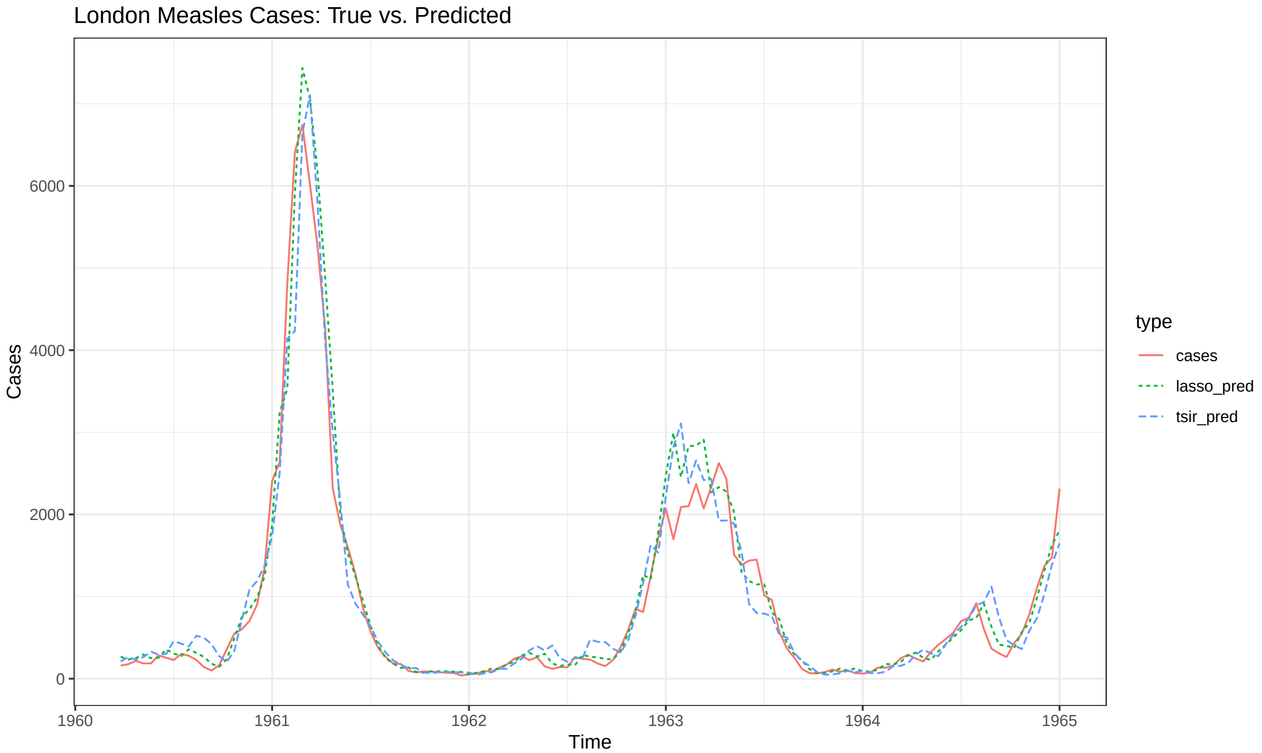
LASSO appears to forecast one-step ahead very accurately on the training dataset. Is performance similarly high on the test dataset?

```
X_test <- as.matrix(lasso_data_test[, X_col_names])

test_pred <- predict(
    model.oneahead,
    newx = X_test,
    type = "response"
)

lasso_data_test$lasso_pred <- test_pred

lasso_data_test |>
    ggplot(aes(x = time)) +
    geom_line(aes(y = cases)) +
    geom_line(aes(y = lasso_pred), colour = "red")
```



Finally, let us compare TSIR predictions with LASSO predictions.

```
tsir_pred <- london_tsir_pred[london_tsir_pred$time %in% lasso_data_test$time, ]
lasso_data_test$tsir_pred <- tsir_pred$tsir_pred

lasso_data_test |>
    pivot_longer(c("cases", "tsir_pred", "lasso_pred"),
                 names_to = "type",
                 values_to = "value") |>
    ggplot(aes(x = time, y = value, color = type, linetype = type)) +
    geom_line() +
    labs(title = "London Measles Cases: True vs. Predicted",
         x = "Time",
         y = "Cases")
```




```
tsir_mse <- mean((lasso_data_test$cases - lasso_data_test$tsir_pred)^2)
lasso_mse <- mean((lasso_data_test$cases - lasso_data_test$lasso_pred)^2)

print(paste0("TSIR MSE: ", round(tsir_mse, 2)))
```

```
## [1] "TSIR MSE: 125248.97"
```

```
print(paste0("LASSO MSE: ", round(lasso_mse, 2)))
```

```
## [1] "LASSO MSE: 95277.97"
```

Exercises

1. Mean Absolute Error (MAE) may be a more appropriate metric due to large positive incidence values. Calculate MAE for the TSIR and LASSO test predictions.
2. Experiment with different T_{lag} values. Is the LASSO still better than TSIR for small T_{lag} values.
3. Fit the LASSO model on a different city and/or for different step-ahead forecasts.