Assignment 5

Solutions

Mark Deming

4/20/2025

Overview

In this assignment, you will replicate and interpret an interrupted time-series (ITS) analysis using data on acute coronary events (ACEs) in Sicily, Italy. The dataset focuses on hospitalizations among individuals aged 0–69, from 2002 to 2006. This assignment is based on Lopez-Bernal et al. (2017): https://github.com/gasparrini/2017_lopezbernal_IJE_codedata/tree/master.

In January 2005, Italy enacted a national ban on smoking in indoor public spaces. You will evaluate the impact of this intervention using both linear and Poisson regression approaches.

The dataset is available at the course GitHub page. It includes the following variables:

- `year`: calendar year
- `month`: calendar month
- `aces`: number of ACE hospitalizations
- `time`: monthly time index (1 to 59)
- `smokban`: smoking ban indicator (0 before Jan 2005, 1 after)
- `pop`: raw population
- `stdpop`: age-standardized population in person-years

Part 1: Preparing and Visualizing the Data

Q1. Load the `sicily.csv` dataset. You will also require the following packages:

- `here`
- `readr`
- `tidyverse`
- `lmtest`

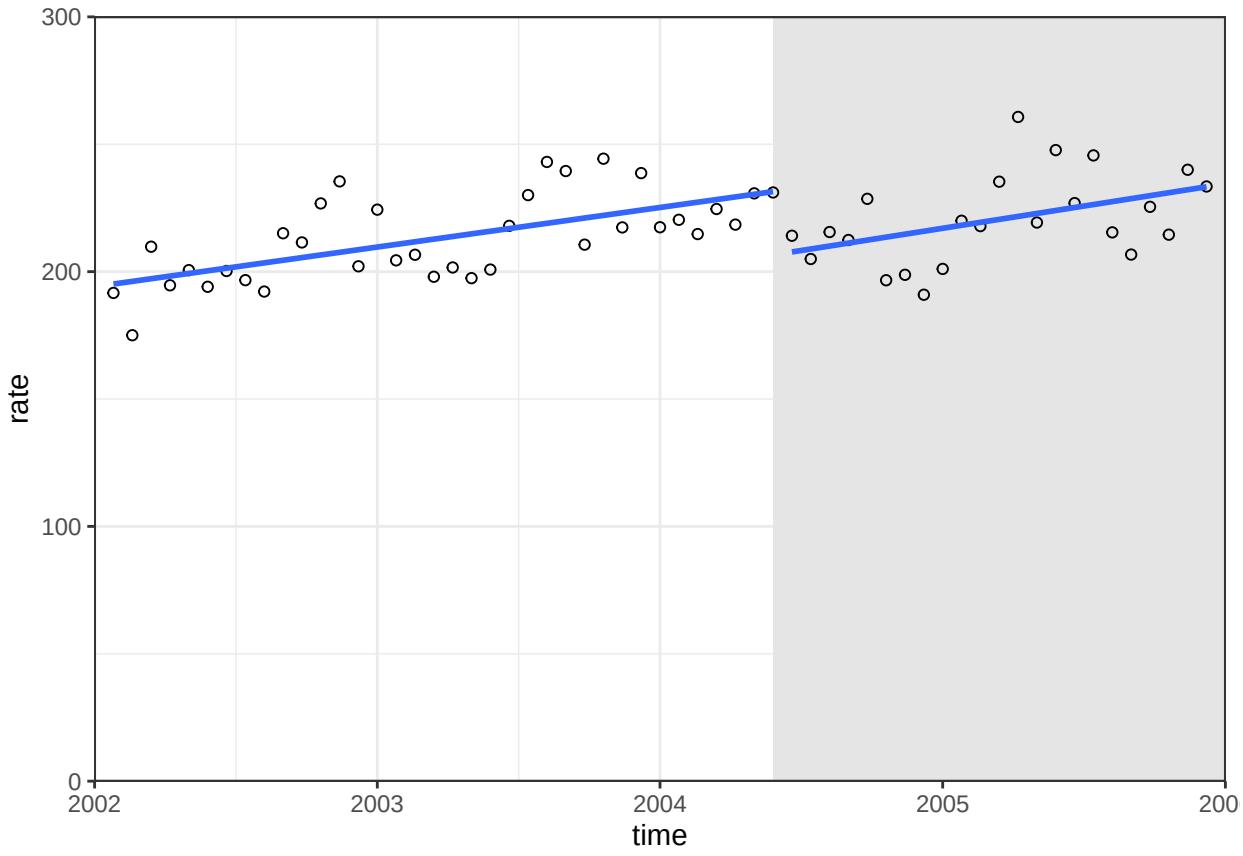
Q2. `aces` counts the number of ACEs per month. Below, calculate a new variable `rate` that is the age-standardized rate of ACEs per 100,000 person-years. **Note:** To convert the rate to per 100,000, you will need to multiply by 10^5.

```
data$rate <- with(data, aces / stdpop * 10 ^ 5)
```

Q3. Use `ggplot` to create a scatterplot of the ACEs rate over time. You should try to replicate this image at GitHub: https://github.com/jmdeming/independent_study/blob/main/assignments/assignment5/figure_question3.png

```
ggplot(data, aes(x = time, y = rate)) +
  geom_rect(aes(xmin = 36, xmax = 60, ymin = 0, ymax = 300), fill = "gray90") +
  geom_point(shape = 1) +
  geom_smooth(data = subset(data, time < 37), method = "lm", se = F) +
  geom_smooth(data = subset(data, time > 36), method = "lm", se = F) +
  scale_x_continuous(limit = c(0, 60),
                     breaks = seq(0, 60, 15),
                     labels = seq(2002, 2006, 1),
                     expand = c(0, 0)) +
  scale_y_continuous(limit = c(0, 300),
                     expand = c(0, 0)) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Part 2: OLS Regression

In this section, you will estimate the impact of the smoking ban using OLS regression, with the standardized rate as the dependent variable.

Q4. Below, run an OLS regression. The dependent variable is `rate`. The independent variables are `smokban` and `time`. Use `coeftest` to print the output of your regression.

```r
ols_model <- lm(rate ~ smokban + time, data = data)
coeftest(ols_model)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 193.62236    4.41565 43.8492 < 2.2e-16 ***
## smokban     -24.09330    7.05001 -3.4175  0.001184 **
## time          1.06214    0.20191  5.2605 2.339e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q5. Examine the results of your OLS regression above. In 1-2 sentences below, interpret the coefficient for `smokban`.

Part 3: Addressing Seasonality

Examining your scatterplot above, you may notice regular seasonal patterns in the data. One way to address this is by differencing the data by 12 months.

Q6. Create a new variable `rate_diff12` that is the current `rate` minus the `rate` 12 months prior.

```r
data$rate_diff12 <- data$rate - lag(data$rate, 12)
```

Q7. Run the OLS regression on the deaseasoned variable `rate_diff12`.

```r
ols_model_diff <- lm(rate_diff12 ~ smokban + time, data = data)
coeftest(ols_model_diff)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.92323    9.22978 -0.9668 0.338935
## smokban     -30.27148    9.53233 -3.1757 0.002731 **
## time          0.87381    0.35129  2.4874 0.016729 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q8. Compare the results to your earlier OLS regression. What changed? Write your answer in 1-2 sentences below.

Part 4: Poisson Regression

Poisson regression is a type of regression for dependent variables that are **count variables** – like `aces`. In this section, you will use Poisson regression to model the *number* of ACEs in Sicily.

In addition to `smokban` and `time`, the authors include `log(stdpop)` as an **offset variable** in their Poisson regression model.

An **offset variable** adjusts for differences in population size, allowing us to model rates (e.g., ACEs per person) rather than just counts. In this case, we are scaling the number of ACEs by the size of the population at risk. To include an offset variable in a regression model, we place the variable inside the `offset()` function.

When we include an offset in a model using the offset() function, R:

- Does not estimate a coefficient for that variable,
- But does use it to adjust the model so that counts are interpreted as rates.

In short, adding offset(log(stdpop)) tells the model: "The number of events should go up if the population is bigger—so account for that when estimating effects."

Q9. Run a Poisson regression using `aces` as the dependent variable and `log(stdpop)` as an offset. Include `time` and `smokban` as independent variables. You may need to do a quick google search for how to run a poisson model in `glm()`.

```
#Poisson with the standardised population as an offset
poisson_model <- glm(aces ~ smokban + time + offset(log(stdpop)), family = poisson, data)
coeftest(poisson_model)
```

```
##
## z test of coefficients:
##
##                Estimate  Std. Error   z value  Pr(>|z|)
## (Intercept) -6.24315091  0.01118935 -557.9546 < 2.2e-16 ***
## smokban     -0.11162844  0.01722477   -6.4807  9.13e-11 ***
## time         0.00494499  0.00049923    9.9052 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Part 5: Predictions and Counterfactual

You will now use your Poisson regression model to generate predicted values for the ACE hospitalization rate over time, both with and without the smoking ban. Then, you will visualize these predictions alongside the actual data.

Q10. Below, create a new dataset for generating predictions.

Here, we want to:

- Generate predicted rates using the Poisson model,
- Create a smooth line. So, instead of 1 point per month (like in the original dataset), we'll use 10 points per month, for a total of 600 time points (60 months $\times$ 10),
- Include the relevant variables: `time`, `smokban`, and `stdpop`.

What your new dataset should include:

1. A variable time that ranges from 1.0 to 60.0 in 0.1 increments (i.e., 600 evenly spaced points). **Hint:** Use 1:600 / 10 to generate this.
2. A variable `smokban` that is 0 for all months before the smoking ban (i.e., before month 37), 1 for all months from 37 onward. **Hint:** You'll need to assign 0 for the first 360 rows and 1 for the remaining 240 rows (36 months $\times$ 10 = 360 points).
3. A variable `stdpop` that is set to the mean of `data$stdpop` for all rows. **Hint:** Use `mean(data$stdpop)` to compute this.

4. Optionally, include a `month` variable (this won't be used in predictions, but helps keep structure consistent).

```
# create a new dataframe with 0.1 time units to improve the graph
datanew <- data.frame(stdpop = mean(data$stdpop),
                      smokban = rep(c(0, 1), c(360, 240)),
                      time = 1:600/10,
                      month = rep(1:120/10, 5))
```

Q11. Use `predict()` to calculate the expected number of `aces` events at each time point in your new data above. Then, convert those counts into rates per 100,000 by dividing by the population and multiplying by 10^5.

```
# We generate predicted values based on the model in order to create a plot
pred1 <- predict(poisson_model, type = "response", datanew) / mean(data$stdpop) * 10^5
```

Q12. Generate counterfactual predictions (no smoking ban):

- Create a new dataset identical to the one above, but set `smokban = 0` for all rows.
- Use your model to generate predicted ACEs counts under this "no intervention" scenario.
- Convert the predicted counts to rates.

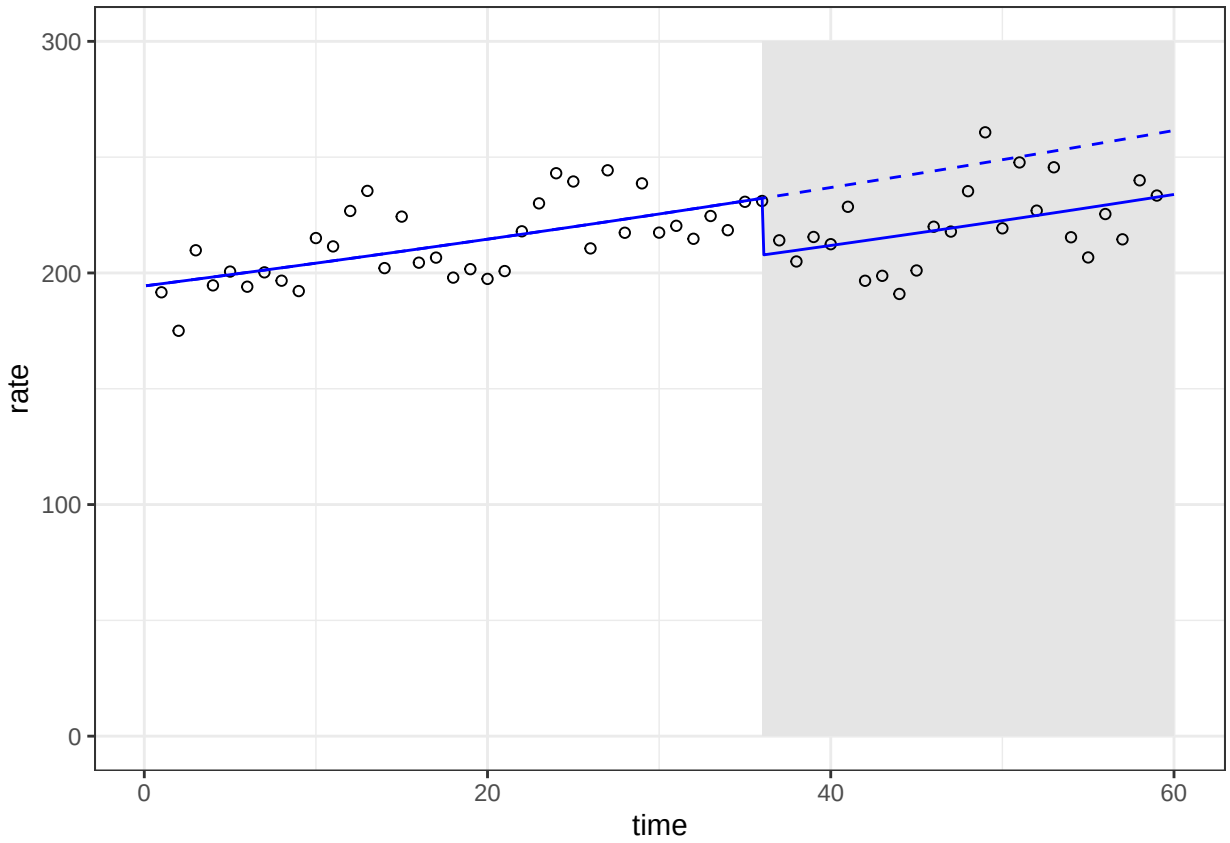
```
# to plot the counterfactual scenario we create a data frame as if smokban
#   (the intervention) was never being implemented
datanew2 <- data.frame(stdpop = mean(data$stdpop),
                       smokban = 0,
                       time = 1:600/10,
                       month = rep(1:120/10,5))

# generate predictions under the counterfactual scenario and add it to the plot
pred1b <- predict(poisson_model , datanew2 , type = "response") / mean(data$stdpop) * 10^5
```

Q13. Plot the actual and counterfactual scenarios:

- Plot the observed ACEs rate (`data$rate`) as points.
- Plot the predicted values from your model (`pred1`) as a solid line.
- Plot the predicted values from your counterfactual model (`pred1b`) as a dashed line.
- Shade the post-intervention period.
- Label axes clearly.

```
ggplot() +
  geom_rect(aes(xmin = 36, xmax = 60, ymin = 0, ymax = 300), fill = "gray90") +
  geom_point(data = data , aes(x = time, y = rate), shape = 1) +
  geom_line(data = datanew, aes(x = time, y = pred1), color = "blue") +
  geom_line(data = datanew2, aes(x = time, y = pred1b), color = "blue", linetype = "dashed") +
  theme_bw()
```


300
200
100
0
rate
0
20
40
60
time