ALR 2014 Caual Modelling Workshop

Example 1: Interupted time series

The impact of public transportation strikes on use of a bicycle share program in London: Interrupted time series design

Fuller D, Sahlqvist S, Cummins S, Ogilvie D. The impact of public transportation strikes on use of a bicycle share program in London: Interrupted time series design. Preventive Medicine. 2011;54(1):74â€“76.

Analysis set up

Set working directory and importing the data into R

```
options(scipen = 1, digits = 2)
opts_chunk$set(warning = FALSE, message = FALSE, error = FALSE)

setwd("/Users/DogLeg/Dropbox/Conferences/2014 ALR/Workshop Data/")
boris_data <- read.csv("http://www.walkabilly.net/Presentations/Workshops/boris_data.txt")
```

Installing necessary packages

```
install.packages("ggplot2", repos = "http://cran.rstudio.com/")
library(ggplot2)
install.packages("car", repos = "http://cran.rstudio.com/")
library(car)
```

Objectives

We want to examine the impact of 2 tube (read subway) strikes in London on the use of the Boris Bike program. The strikes happened on September 6th, 2010 and October 4th, 2010. First we need to generate the appropriate variables for the analysis. First, we need 2 dummy variables that indicate before and after each strike.

Creating necessary variables

Creating dichotomous variables to assess the immediate impact of the tube strike

Key dates that we need for the analysis are the tube strikes

1. September 6, 2010
2. October 4, 2010

The first strike happened at time 39 (September 6, 2010) so we need to create a variable that is 0 before time 39, 1 between time 39 and 65, and 0 again after time 65.

```
boris_data$strike1 <- 1
boris_data$strike1[boris_data$time < 39] <- "0"
boris_data$strike1[boris_data$time > 66] <- "0"
boris_data$strike1 <- as.numeric(boris_data$strike1)
```

The second strike happened at time 65 (October 4, 2010) so we need to create a variable that is 1 before time 65 and 0 after after time 65.

```
boris_data$strike2 <- boris_data$time - 66
boris_data$strike2 <- ifelse(boris_data$time < 67, c("0"), c("1"))
boris_data$strike2 <- as.numeric(boris_data$strike2)
```

Creating continuous variables to assess the change in impact over time

Creating a variable that is 0 before the first ban and linear after the first ban

```
boris_data$slope2 <- boris_data$time - 38
boris_data$slope2[boris_data$time < 39] <- "0"
boris_data$slope2 <- as.numeric(boris_data$slope2)
```

Creating a variable that is 0 before the second ban and linear after the second ban

```
boris_data$slope3 <- boris_data$time - 66
boris_data$slope3[boris_data$time < 67] <- "0"
boris_data$slope3 <- as.numeric(boris_data$slope3)
```

Creating a categorical variable that defines each strike period (i.e., pre, strike1 and strike 2). This is mostly to simply our life when graphing.

```
boris_data$strike[boris_data$time < 39] <- "0"
boris_data$strike[boris_data$time > 38 & boris_data$time < 67] <- "1"
boris_data$strike[boris_data$time > 66] <- "2"
```

Dataset structure

Your dataset should look like below. Notice that the dummy variables change at the correct times.

```
library(xtable)
cut_data <- boris_data[30:75, ]
print(xtable(cut_data), type = "html")
```

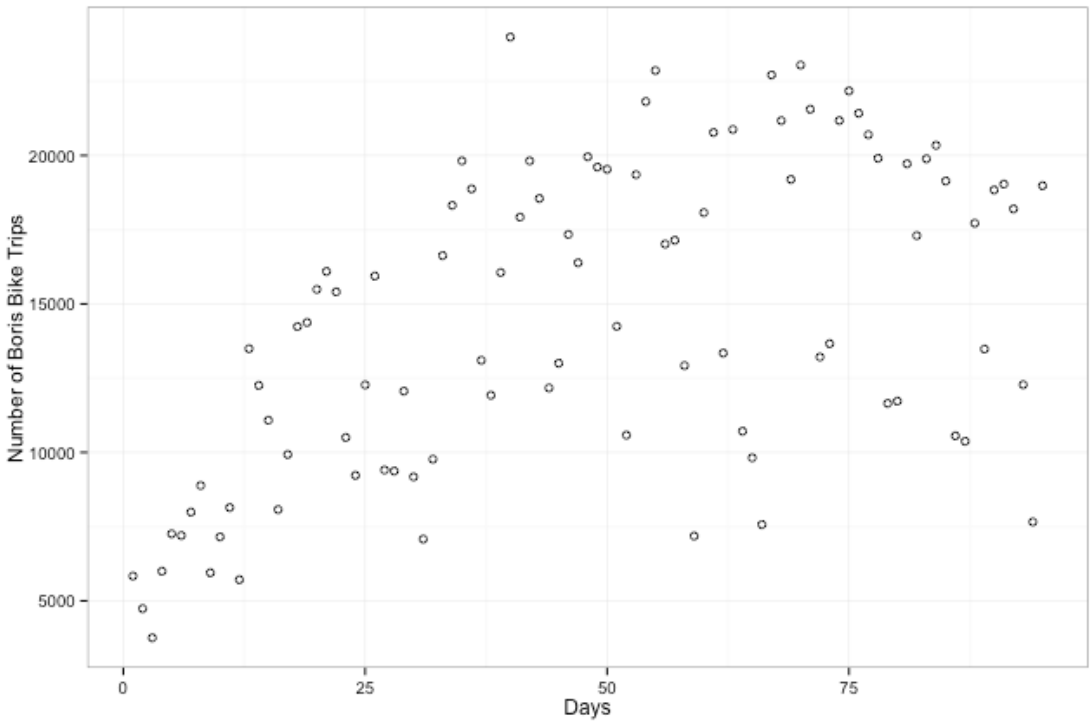
	start_date	duration_sec	t_trip	duration_min	time	time1	time2	t0_int	t1_int	t2_int	strike1	strike2	slope2	slope3	strike
30	28-Aug-10	1181.57	9176	19.69	30	0	0	1	0	0	0.00	0.00	0.00	0.00	0
31	29-Aug-10	1075.66	7079	17.93	31	0	0	1	0	0	0.00	0.00	0.00	0.00	0
32	30-Aug-10	1070.03	9771	17.83	32	0	0	1	0	0	0.00	0.00	0.00	0.00	0
33	31-Aug-10	1050.86	16628	17.51	33	0	0	1	0	0	0.00	0.00	0.00	0.00	0
34	01-Sep-10	1085.75	18320	18.10	34	0	0	1	0	0	0.00	0.00	0.00	0.00	0
35	02-Sep-10	1284.23	19818	21.40	35	0	0	1	0	0	0.00	0.00	0.00	0.00	0
36	03-Sep-10	1092.22	18873	18.20	36	0	0	1	0	0	0.00	0.00	0.00	0.00	0
37	04-Sep-10	1166.49	13101	19.44	37	0	0	1	0	0	0.00	0.00	0.00	0.00	0
38	05-Sep-10	1410.71	11924	23.51	38	0	0	1	0	0	0.00	0.00	0.00	0.00	0
39	06-Sep-10	1084.68	16058	18.08	39	1	0	0	1	0	1.00	0.00	1.00	0.00	1
40	07-Sep-10	1324.32	23988	22.07	40	2	0	0	1	0	1.00	0.00	2.00	0.00	1
41	08-Sep-10	941.46	17923	15.69	41	3	0	0	1	0	1.00	0.00	3.00	0.00	1
42	09-Sep-10	1174.07	19818	19.57	42	4	0	0	1	0	1.00	0.00	4.00	0.00	1
43	10-Sep-10	960.77	18552	16.01	43	5	0	0	1	0	1.00	0.00	5.00	0.00	1
44	11-Sep-10	1156.80	12170	19.28	44	6	0	0	1	0	1.00	0.00	6.00	0.00	1
45	12-Sep-10	1222.94	13003	20.38	45	7	0	0	1	0	1.00	0.00	7.00	0.00	1
46	13-Sep-10	1082.33	17341	18.04	46	8	0	0	1	0	1.00	0.00	8.00	0.00	1
47	14-Sep-10	887.05	16385	14.78	47	9	0	0	1	0	1.00	0.00	9.00	0.00	1

48	15-Sep-10	966.58	19956	16.11	48	10	0	0	1	0	1.00	0.00	10.00	0.00 1
49	16-Sep-10	928.89	19610	15.48	49	11	0	0	1	0	1.00	0.00	11.00	0.00 1
50	17-Sep-10	991.17	19534	16.52	50	12	0	0	1	0	1.00	0.00	12.00	0.00 1
51	18-Sep-10	1131.88	14245	18.86	51	13	0	0	1	0	1.00	0.00	13.00	0.00 1
52	19-Sep-10	1062.85	10581	17.71	52	14	0	0	1	0	1.00	0.00	14.00	0.00 1
53	20-Sep-10	952.14	19354	15.87	53	15	0	0	1	0	1.00	0.00	15.00	0.00 1
54	21-Sep-10	1027.89	21816	17.13	54	16	0	0	1	0	1.00	0.00	16.00	0.00 1
55	22-Sep-10	1092.31	22861	18.21	55	17	0	0	1	0	1.00	0.00	17.00	0.00 1
56	23-Sep-10	1013.52	17015	16.89	56	18	0	0	1	0	1.00	0.00	18.00	0.00 1
57	24-Sep-10	817.70	17145	13.63	57	19	0	0	1	0	1.00	0.00	19.00	0.00 1
58	25-Sep-10	1115.03	12925	18.58	58	20	0	0	1	0	1.00	0.00	20.00	0.00 1
59	26-Sep-10	1048.46	7184	17.47	59	21	0	0	1	0	1.00	0.00	21.00	0.00 1
60	27-Sep-10	878.24	18075	14.64	60	22	0	0	1	0	1.00	0.00	22.00	0.00 1
61	28-Sep-10	874.76	20773	14.58	61	23	0	0	1	0	1.00	0.00	23.00	0.00 1
62	29-Sep-10	797.80	13345	13.30	62	24	0	0	1	0	1.00	0.00	24.00	0.00 1
63	30-Sep-10	1038.94	20876	17.32	63	25	0	0	1	0	1.00	0.00	25.00	0.00 1
64	01-Oct-10	925.37	10714	15.42	64	26	0	0	1	0	1.00	0.00	26.00	0.00 1
65	02-Oct-10	1149.40	9818	19.16	65	27	0	0	1	0	1.00	0.00	27.00	0.00 1
66	03-Oct-10	970.92	7570	16.18	66	28	0	0	1	0	1.00	0.00	28.00	0.00 1
67	04-Oct-10	1168.89	22712	19.48	67	29	1	0	0	1	0.00	1.00	29.00	1.00 2
68	05-Oct-10	1008.68	21172	16.81	68	30	2	0	0	1	0.00	1.00	30.00	2.00 2
69	06-Oct-10	906.67	19193	15.11	69	31	3	0	0	1	0.00	1.00	31.00	3.00 2
70	07-Oct-10	904.57	23044	15.08	70	32	4	0	0	1	0.00	1.00	32.00	4.00 2
71	08-Oct-10	1007.62	21556	16.79	71	33	5	0	0	1	0.00	1.00	33.00	5.00 2
72	09-Oct-10	1064.25	13212	17.74	72	34	6	0	0	1	0.00	1.00	34.00	6.00 2
73	10-Oct-10	1272.94	13661	21.22	73	35	7	0	0	1	0.00	1.00	35.00	7.00 2
74	11-Oct-10	953.43	21176	15.89	74	36	8	0	0	1	0.00	1.00	36.00	8.00 2
75	12-Oct-10	1040.56	22173	17.34	75	37	9	0	0	1	0.00	1.00	37.00	9.00 2

Some simple scatter plots analysis

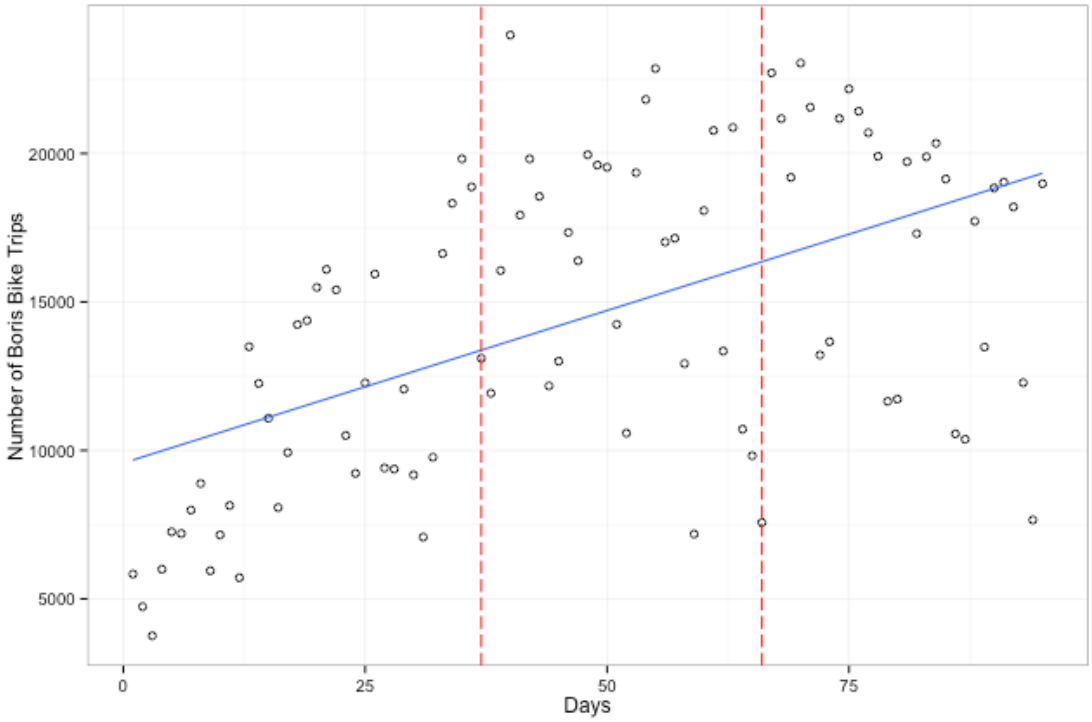
Scatter plot of the Date (x-axis) and Total number of trips (y-axis). Make sure the data is sorted by Date.

```
library(ggplot2)
ggplot(boris_data, aes(x = time, y = t_trip)) + geom_point(shape = 1) + xlab("Days") +
    ylab("Number of Boris Bike Trips") + theme_bw()
```



Scatter plot of the Date (x-axis) and Total number of trips (y-axis) adding the linear fit (blue) and dashed red lines to represent the two strikes.

```
ggplot(boris_data, aes(x = time, y = t_trip)) + geom_point(shape = 1) + stat_smooth(method = "lm",
    se = FALSE) + geom_vline(xintercept = 37, colour = "red", linetype = "longdash") +
    geom_vline(xintercept = 66, colour = "red", linetype = "longdash") + xlab("Days") +
    ylab("Number of Boris Bike Trips") + theme_bw()
```



Regression of each of the newly created variables on the total number of trips

Regression equation

$$\hat{y} = \beta_0 + \beta_1 Time_1 + \beta_2 Strike1_2 + \beta_3 Slope2_3 + \beta_4 Strike2_4 + \beta_5 Slope3_5 + \varepsilon$$

```
trip_reg <- lm(t_trip ~ time + strike1 + slope2 + strike2 + slope3, data = boris_data)
print(xtable(trip_reg), type = "html")
```

	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	5889.7269	1240.8307	4.75	0.0000
time	254.6699	55.4639	4.59	0.0000
strike1	3864.4686	1882.1411	2.05	0.0430
slope2	-465.1409	103.7797	-4.48	0.0000
strike2	11292.6829	3081.8171	3.66	0.0004
slope3	-22.2270	120.9073	-0.18	0.8546

Interpretation of the coefficients is as follows:

- Beta0 (Intercept) â€“ Value of dependent variable at baseline
- Beta1 (time1) - Trend prior to 1st intervention implementation
- Beta2 (strike1) - Difference between the last point prior and first point post the 1st intervention implementation
- Beta3 (slope2) â€“ Change in trend from pre to post 1st implementation
- Beta4 (strike2) - Difference between last point in 1st implementation period and first point in 2nd implementation period
- Beta5 (slope3) â€“ Change in trend from 1st to 2nd implementation periods

Getting predicted values from the regression and mergning them to the "boris_data"" data set

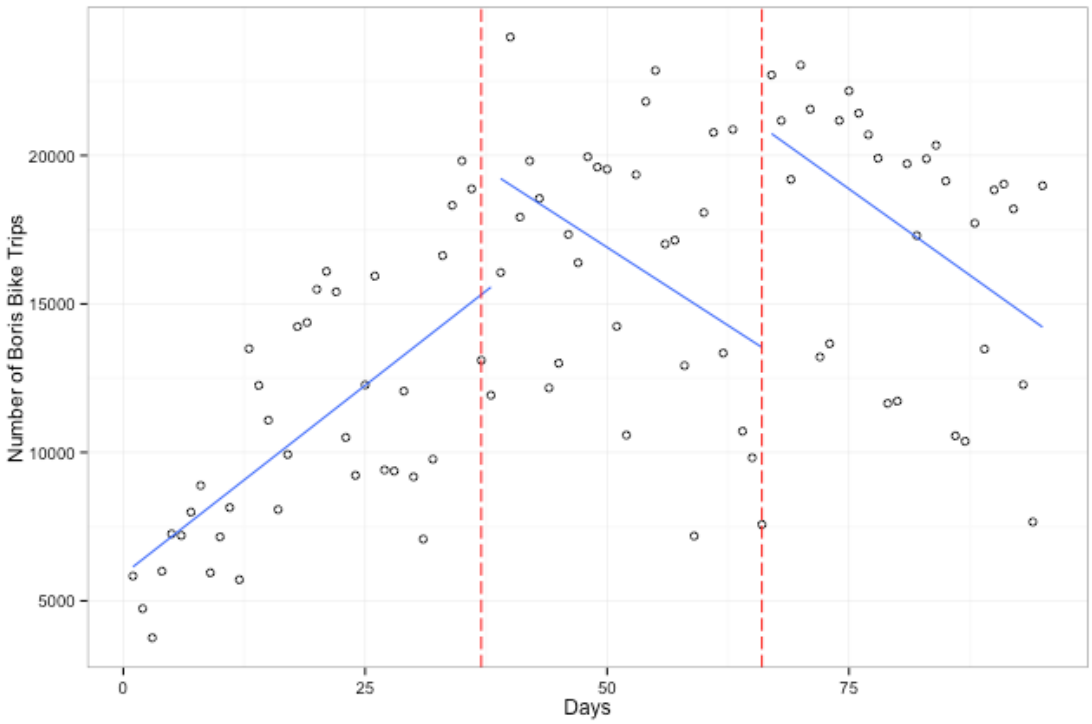
```
trip_hat <- fitted(trip_reg)  # predicted values
boris_data$trip_hat <- fitted(trip_reg)
```

Getting residual values from the regression and mergning them to the "boris_data"" data set

```
trip_resid <- residuals(trip_reg)  # residuals
boris_data$trip_resid <- (trip_resid)
```

Graph the regression plot

```
ggplot(boris_data, aes(x = time, y = t_trip)) + geom_point(shape = 1) + geom_smooth(aes(group = strike),
    method = "lm", se = FALSE) + geom_vline(xintercept = 37, colour = "red",
    linetype = "longdash") + geom_vline(xintercept = 66, colour = "red", linetype = "longdash") +
    xlab("Days") + ylab("Number of Boris Bike Trips") + theme_bw()
```

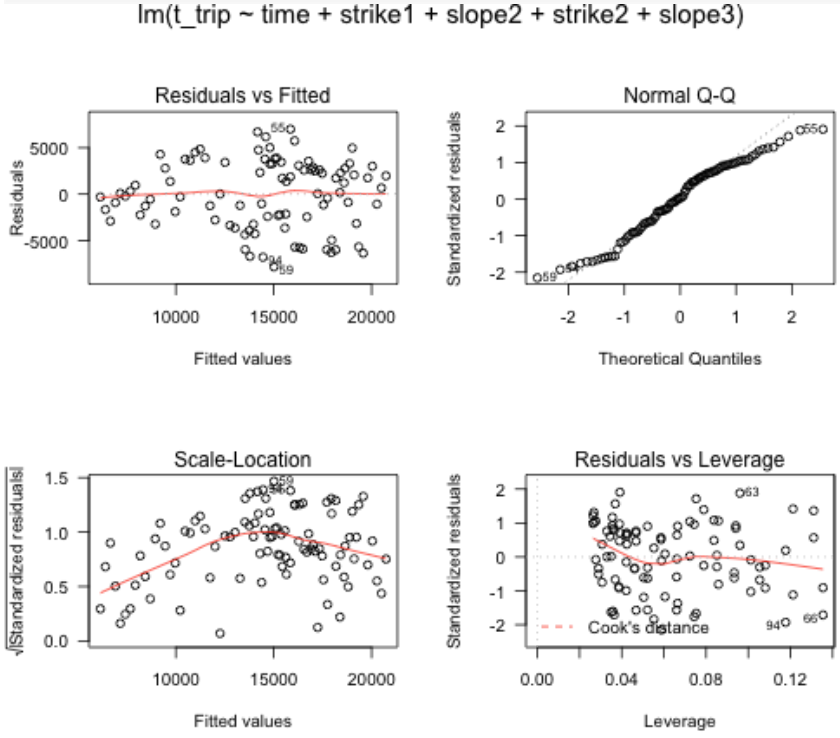


Notes. The plot is a new linear model for each strike period. It is mostly for descriptive purposes. Including CI's in this graph is incorrect as they will be based on the actual model but rather the model used in the graph. That's why I've set the "se=FALSE."

Model fit diagnostics

Fitted residuals, Normal Q-Q plot

```
opar <- par(mfrow = c(2, 2), oma = c(0, 0, 1.1, 0))
plot(trip_reg, las = 1)
```



The Durbin-Watson statistic is a simple numerical method for checking serial dependence.

If the Durbinâ€“Watson statistic is substantially less than 2, there is evidence of positive serial correlation. As a rough rule of thumb, if Durbinâ€“Watson is less than 1.0, there may be cause for alarm. Small values of d indicate successive error terms are, on average, close in value to one another, or positively correlated.Wiki

```
library(car)
durbinWatsonTest(trip_reg)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1            0.28           1.4       0
##  Alternative hypothesis: rho != 0
```

Additional notes

1. We could easily add non linear terms (e.g., quadratic) to the models by squaring or cubing the time, slope2 and slope 3 variables. This might help with model fit.
2. My guess is that there is likely a weekend/weekday autocorrelation. We could add a weekend/weekday dummy variable, which would likely improve model fit.
3. If serial autocorrelation continus to be a problem (after adding weekend/weekday variable) we could add Auto-Regressive Integrated Moving Average (ARIMA) components to the model.
4. There is a package designed for interrupted time series regression in R. I have purposefully not shown this package because it is better to understand the model running the analysis the "manual" way.

   ```
   install.packages("segmented", repos = "http://cran.rstudio.com/")
   library(segmented)
   ```

The end