Photo credit: Dave DiCello

17-803 Empirical Methods

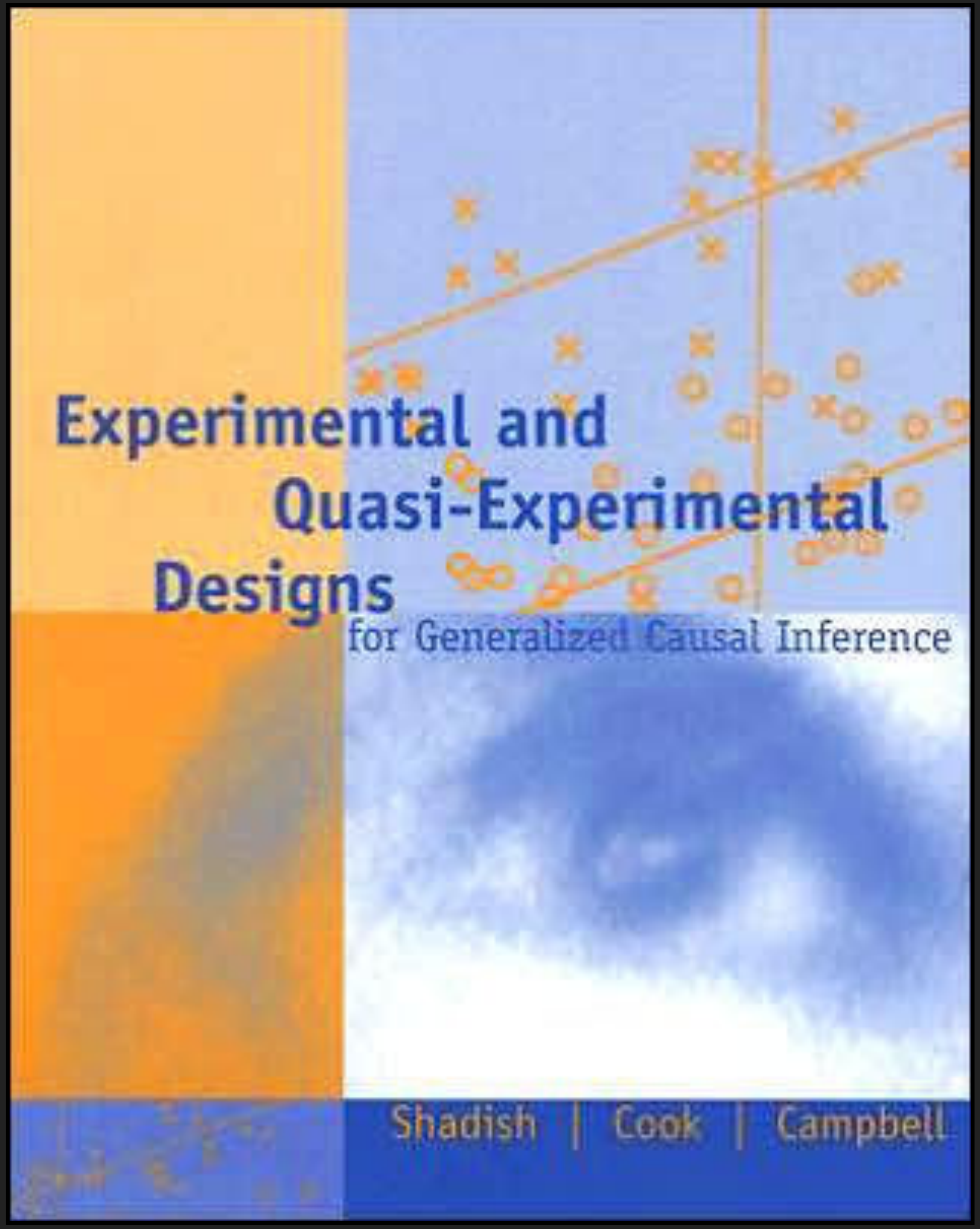
Bogdan Vasilescu, S3D

Time Series Analysis

Tuesday, November 8, 2022

Plan for Today

- Galton families leftovers (see last lecture slides)
- Time series analysis (seasonality/trend decomposition)
- Segmented regression of interrupted time series data



Time Series Analysis

Intro to time series analysis

Beer production in Australia

```
#install.packages("fpp")
library(fpp)
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```
## Loading required package: lmtest
```

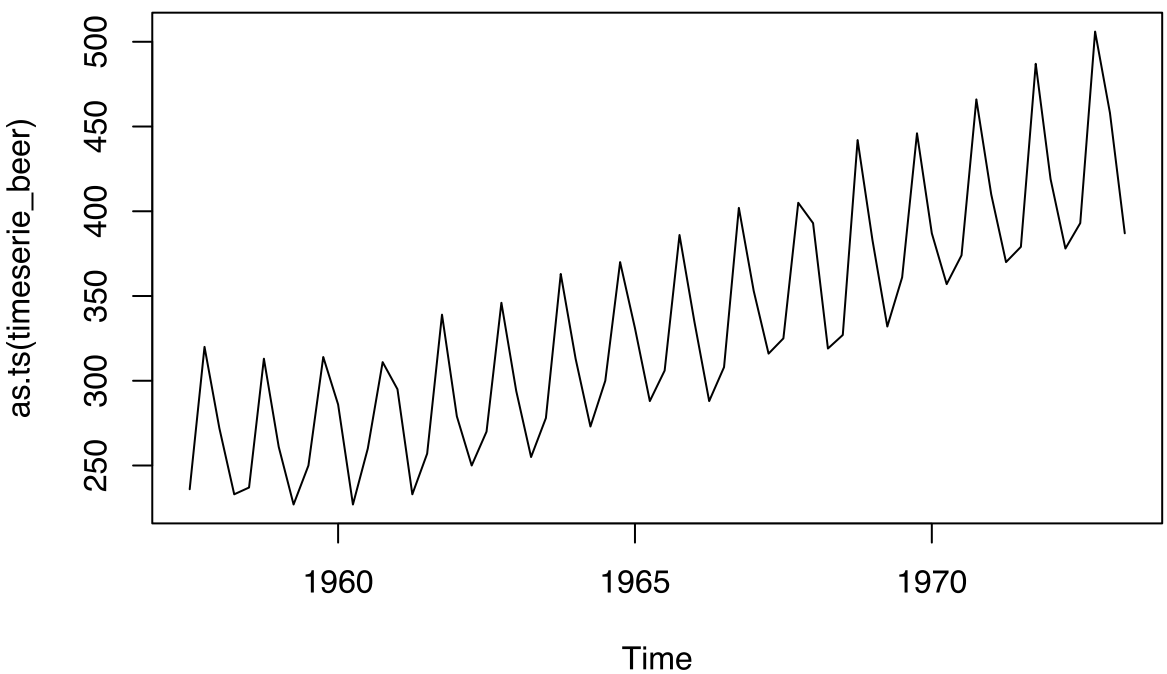
```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: tseries
```

```
data(ausbeer)
timeserie_beer = tail(head(ausbeer, 17*4+2),17*4-4)
plot(as.ts(timeserie_beer))
```

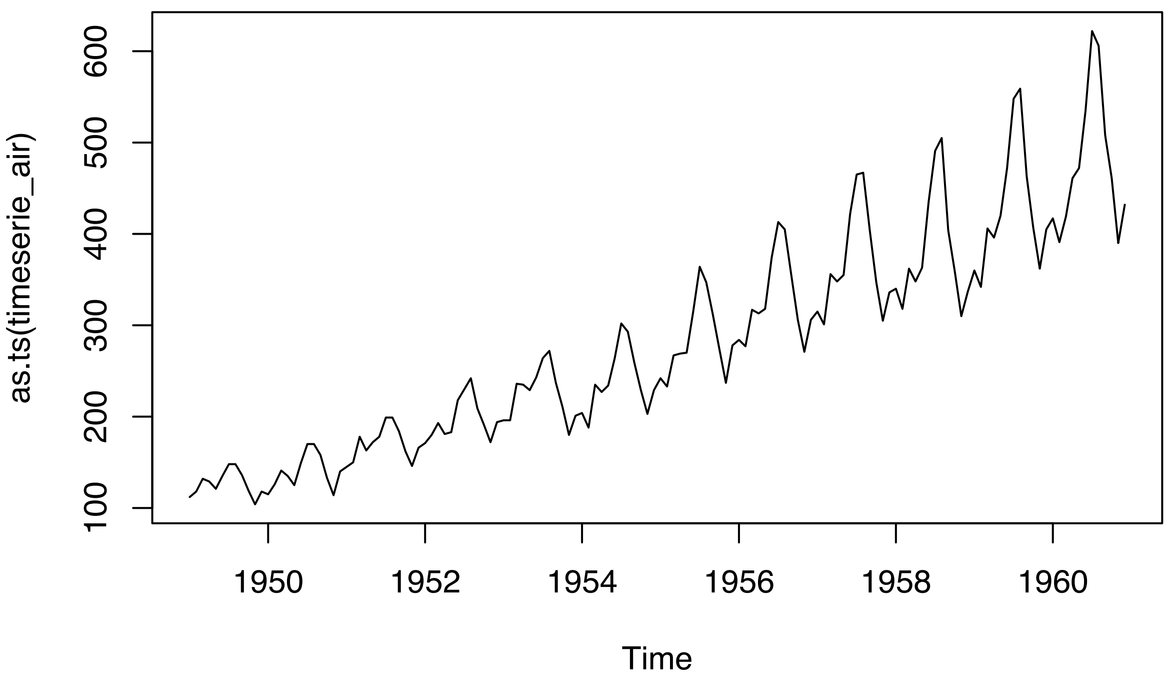


Monthly airline passengers

```
#install.packages("Ecdat")
library(Ecdat)
```

```
## Loading required package: Ecfun

##
## Attaching package: 'Ecfun'

## The following object is masked from 'package:forecast':
##
##     BoxCox

## The following object is masked from 'package:base':
##
##     sign

##
## Attaching package: 'Ecdat'

## The following object is masked from 'package:datasets':
##
##     Orange
```

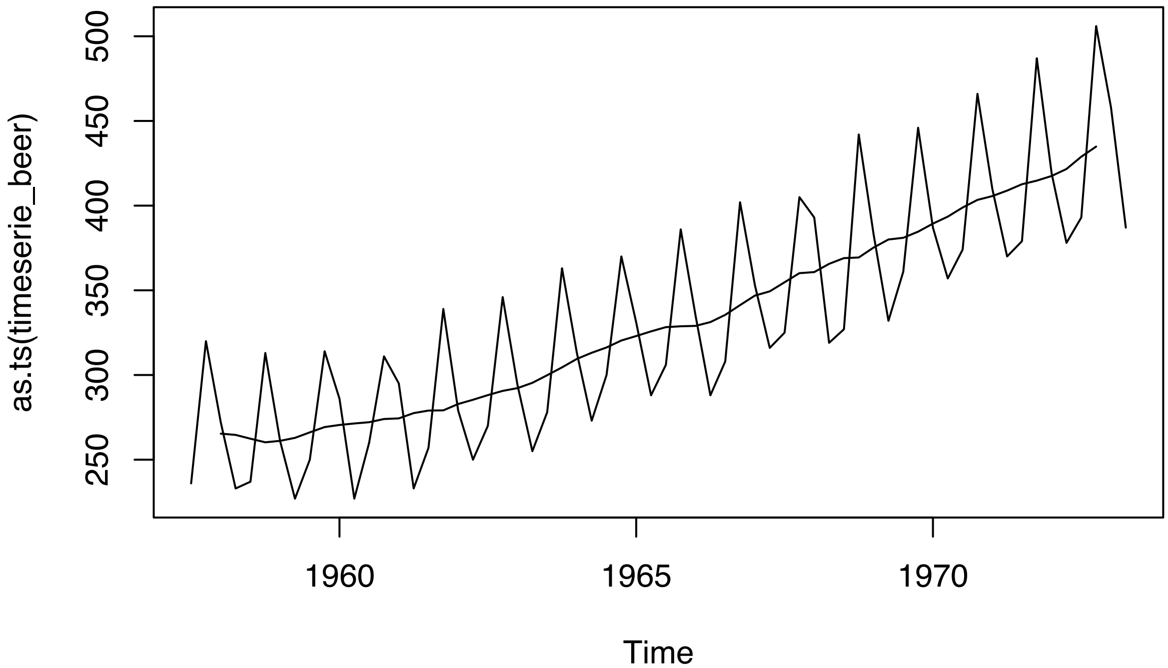
```
data(AirPassengers)
timeserie_air = AirPassengers
plot(as.ts(timeserie_air))
```



Detect trend

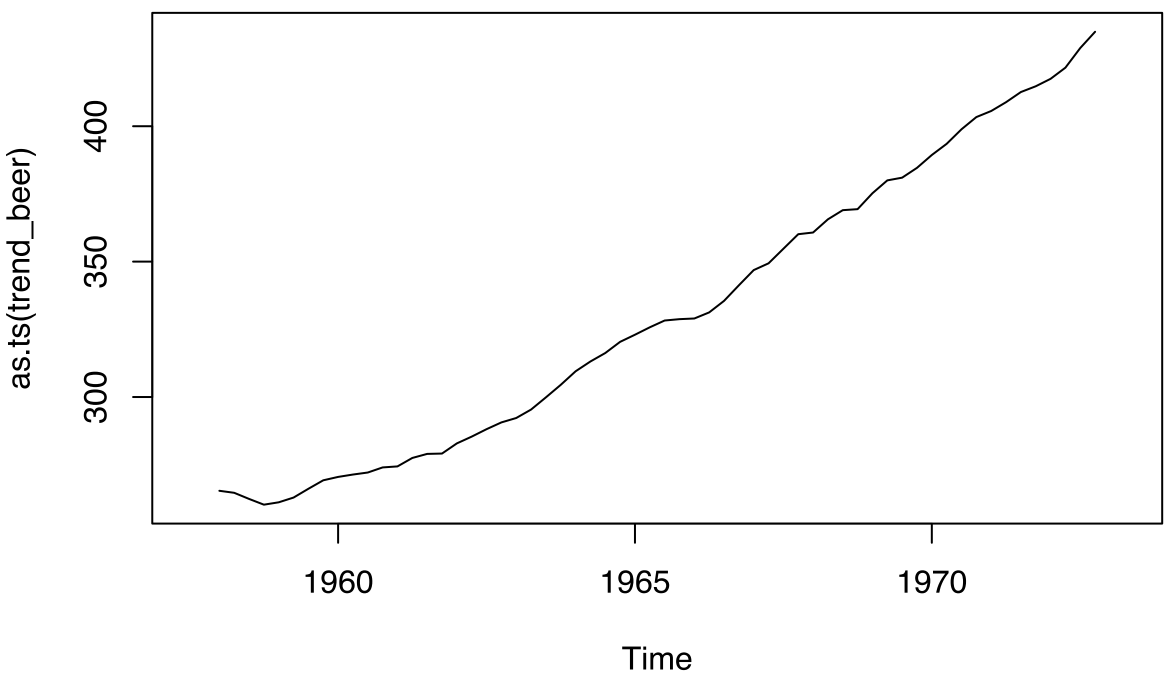
```
#install.packages("forecast")
library(forecast)
trend_beer = ma(timeserie_beer, order = 4, centre = T)
```

```
plot(as.ts(timeserie_beer))
lines(trend_beer)
```

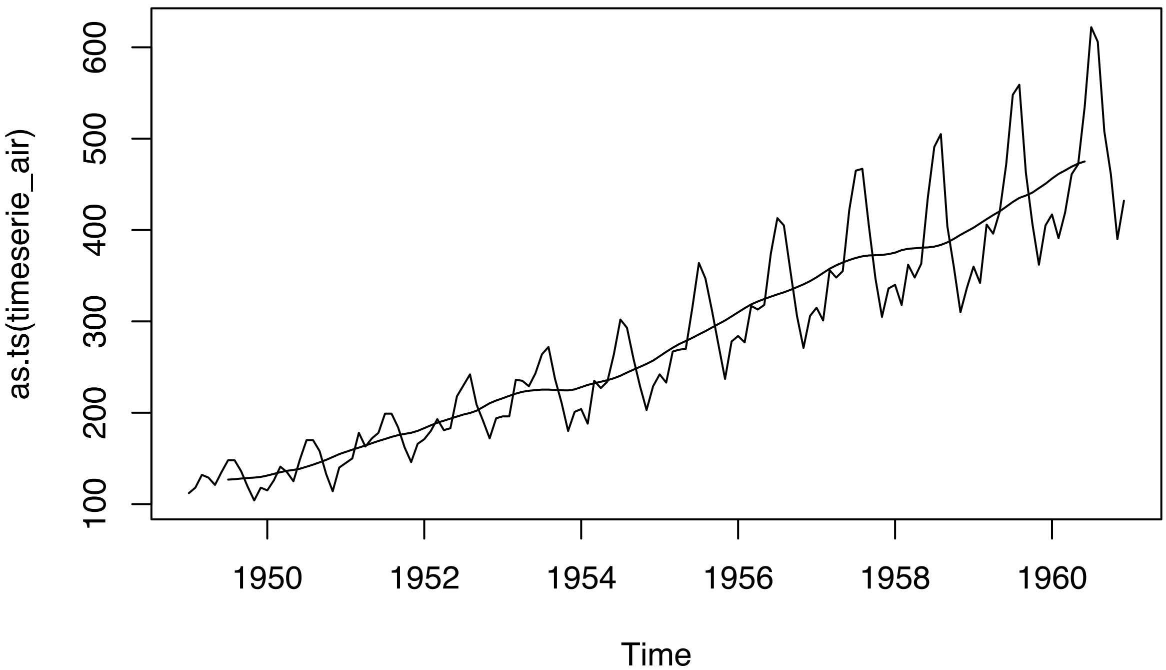


Beer

```
plot(as.ts(trend_beer))
```

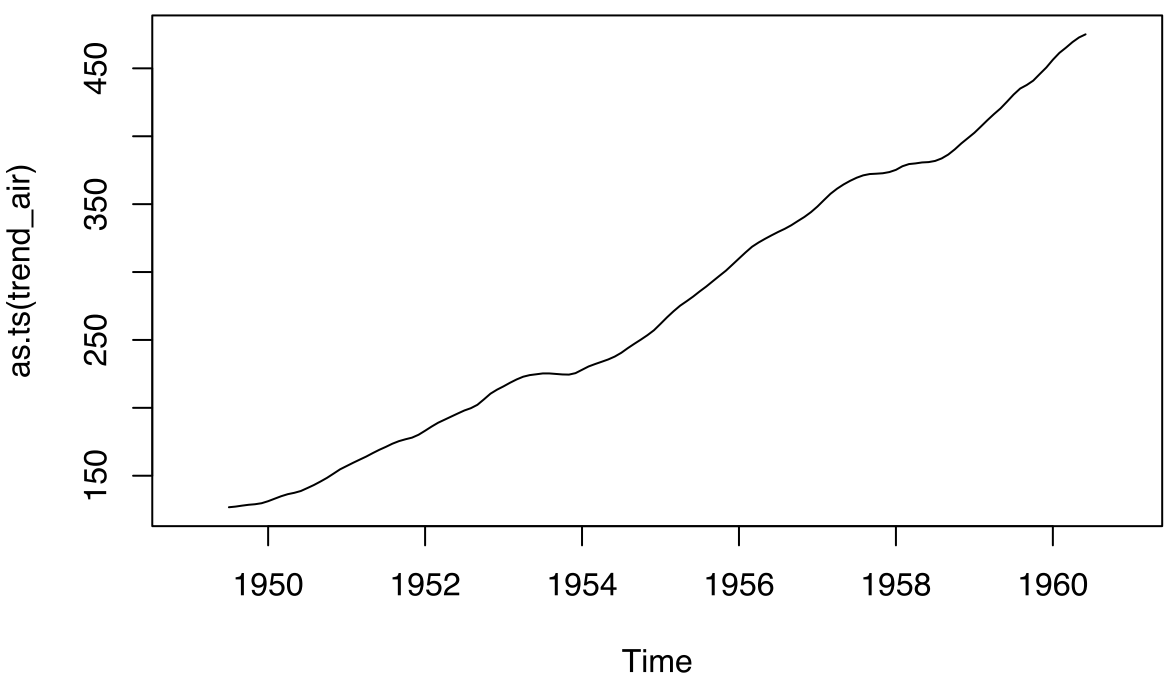



```
trend_air = ma(timeserie_air, order = 12, centre = T)
plot(as.ts(timeserie_air))
lines(trend_air)
```



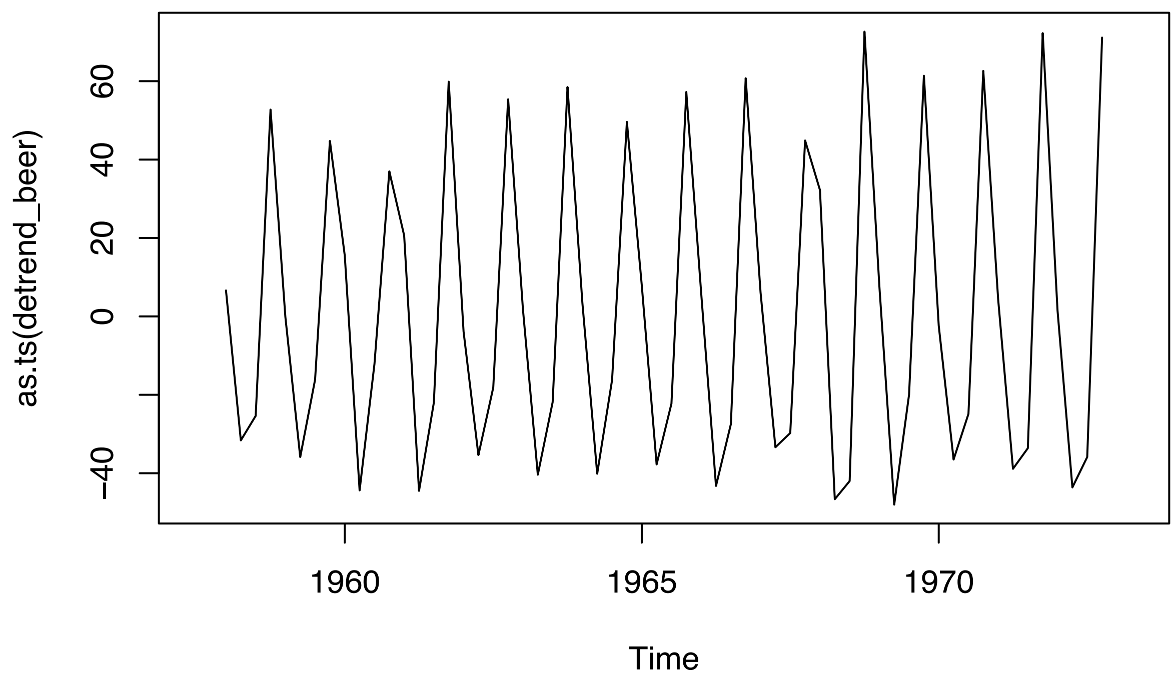
Airlines

```
plot(as.ts(trend_air))
```



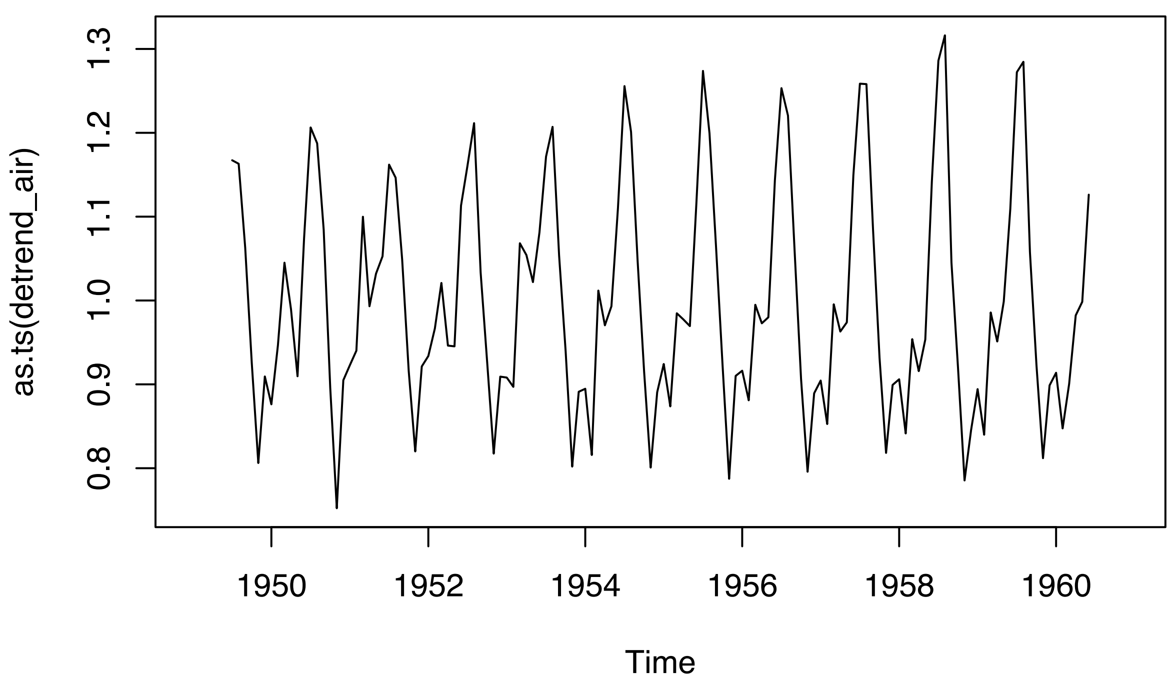
Detrend

```
detrend_beer = timeserie_beer - trend_beer
plot(as.ts(detrend_beer))
```



Beer

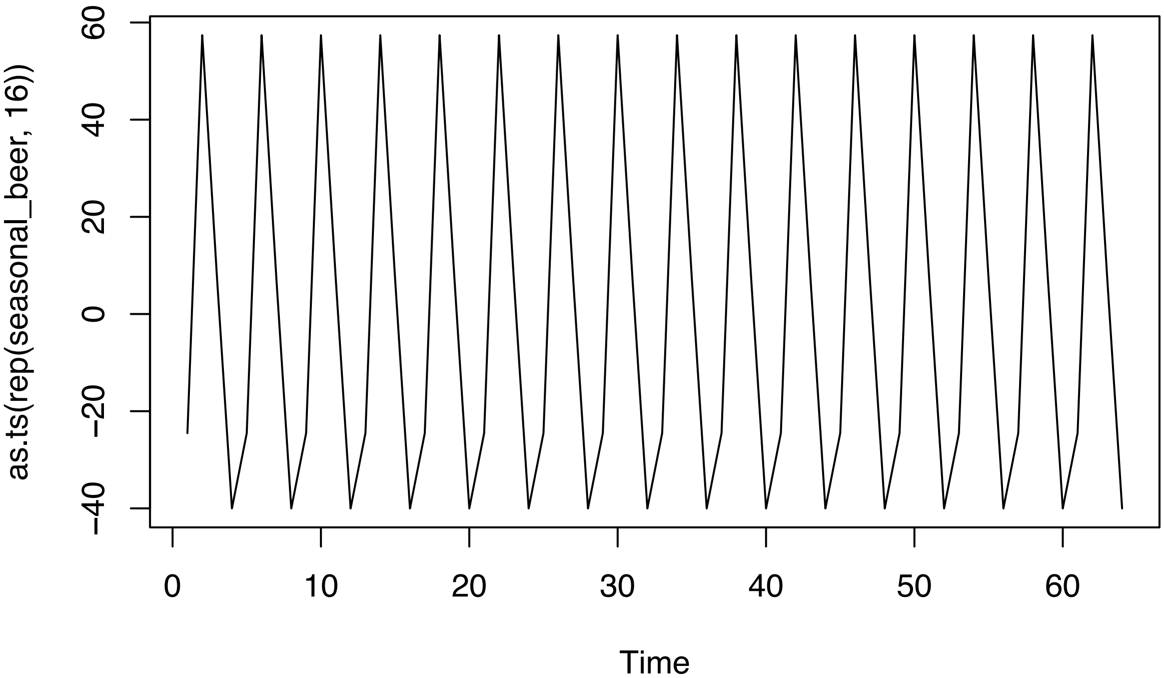
```
detrend_air = timeserie_air / trend_air
plot(as.ts(detrend_air))
```



Air

Seasonality

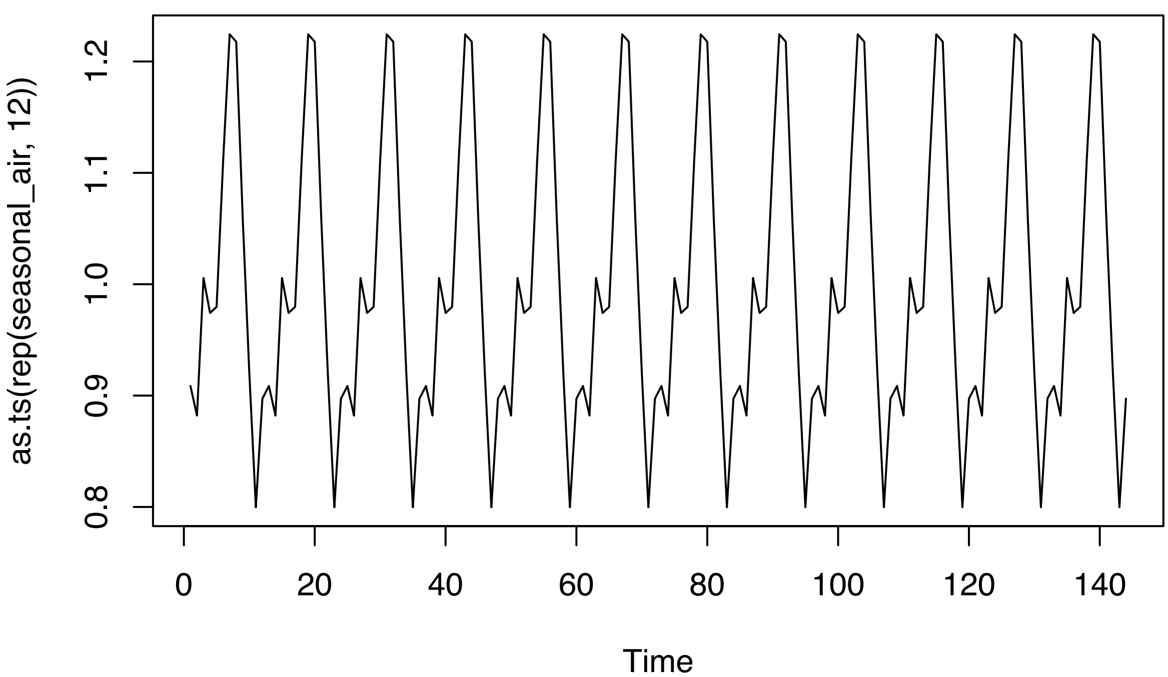
```
m_beer = t(matrix(data = detrend_beer, nrow = 4))
seasonal_beer = colMeans(m_beer, na.rm = T)
plot(as.ts(rep(seasonal_beer,16)))
```



Beer

Air

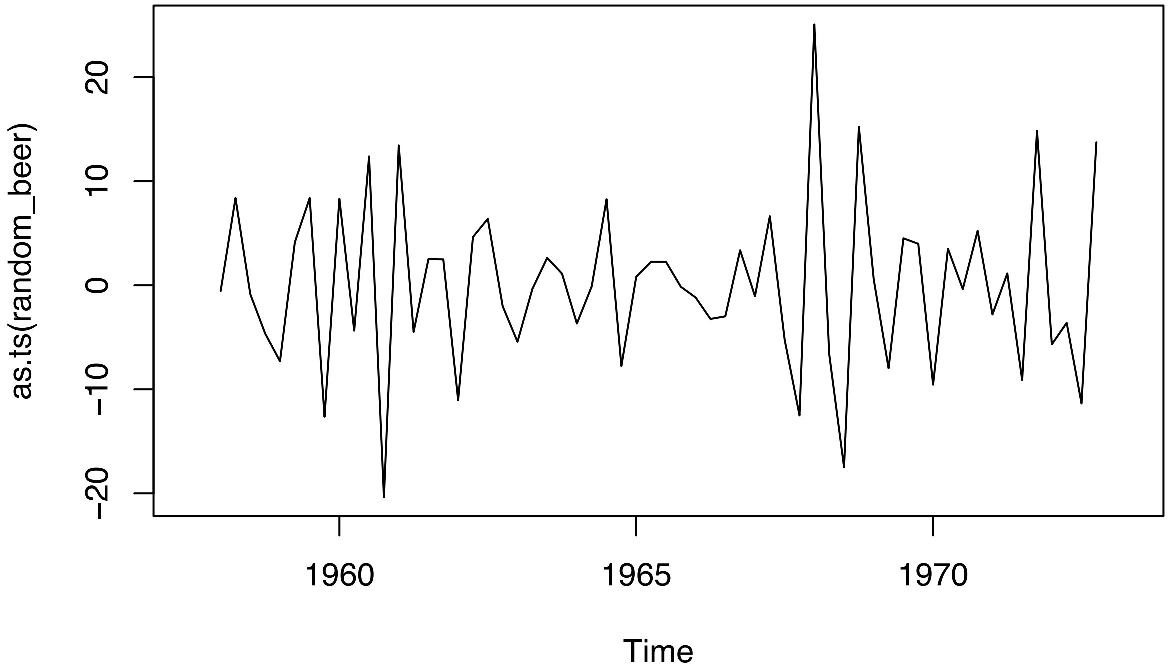
```
m_air = t(matrix(data = detrend_air, nrow = 12))
seasonal_air = colMeans(m_air, na.rm = T)
plot(as.ts(rep(seasonal_air,12)))
```



Random

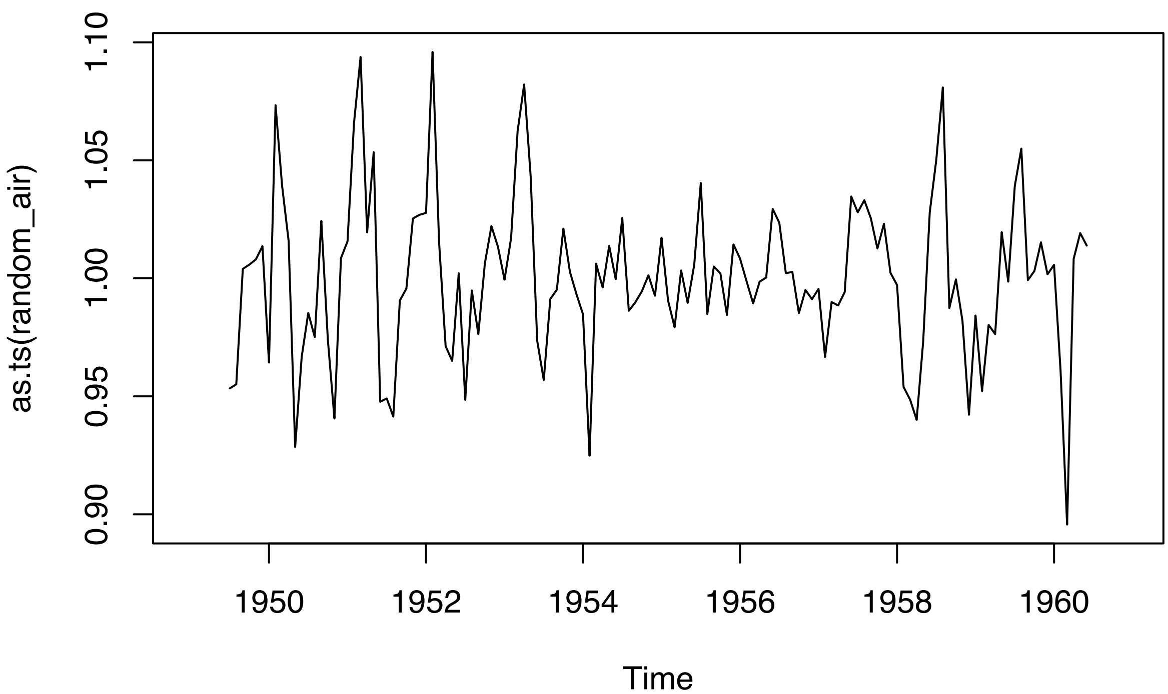
```
random_beer = timeserie_beer - trend_beer - seasonal_beer
```

```
plot(as.ts(random_beer))
```



Beer

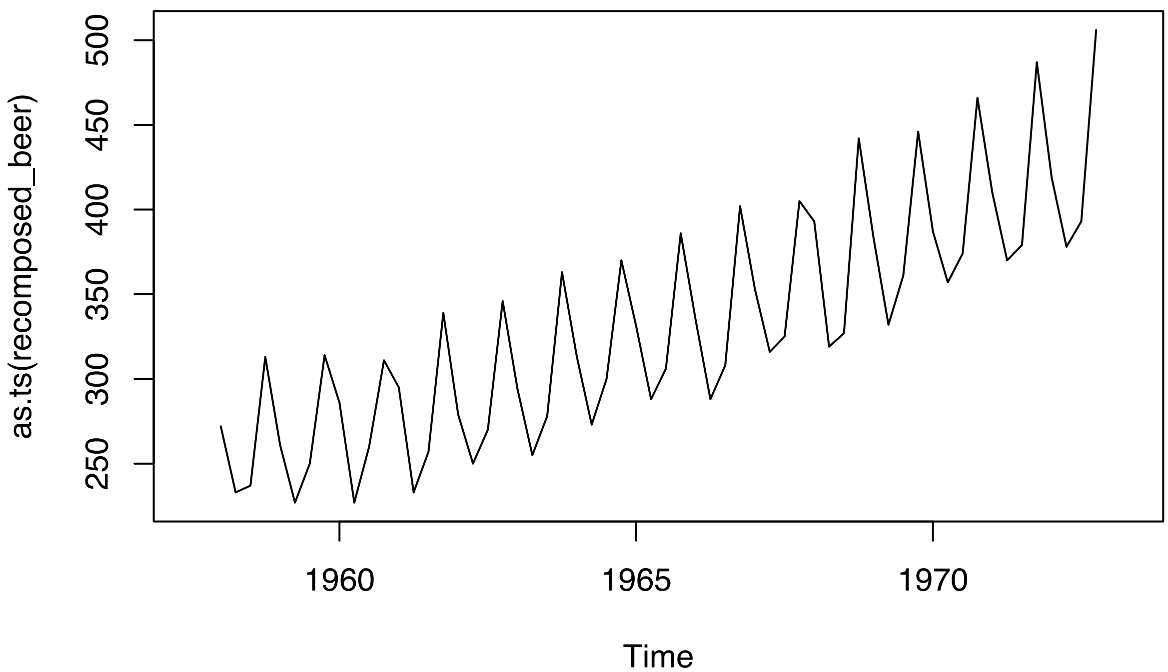
```
random_air = timeserie_air / (trend_air * seasonal_air)
plot(as.ts(random_air))
```



Air

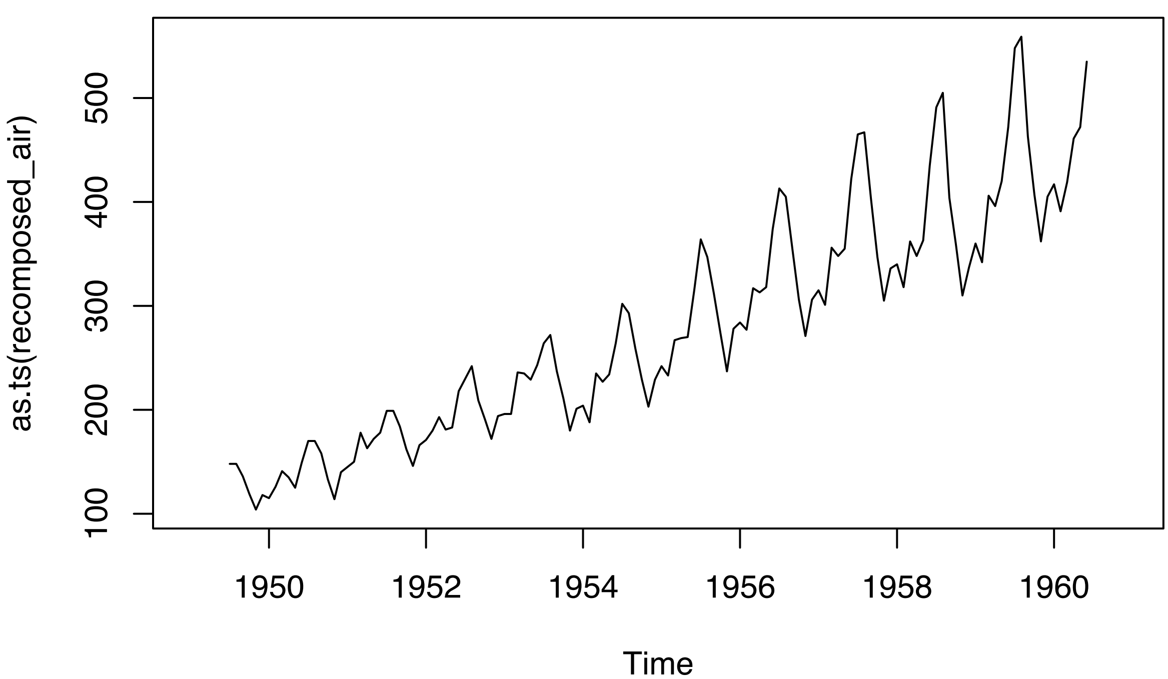
Reconstruct

```
recomposed_beer = trend_beer+seasonal_beer+random_beer
plot(as.ts(recomposed_beer))
```



Beer

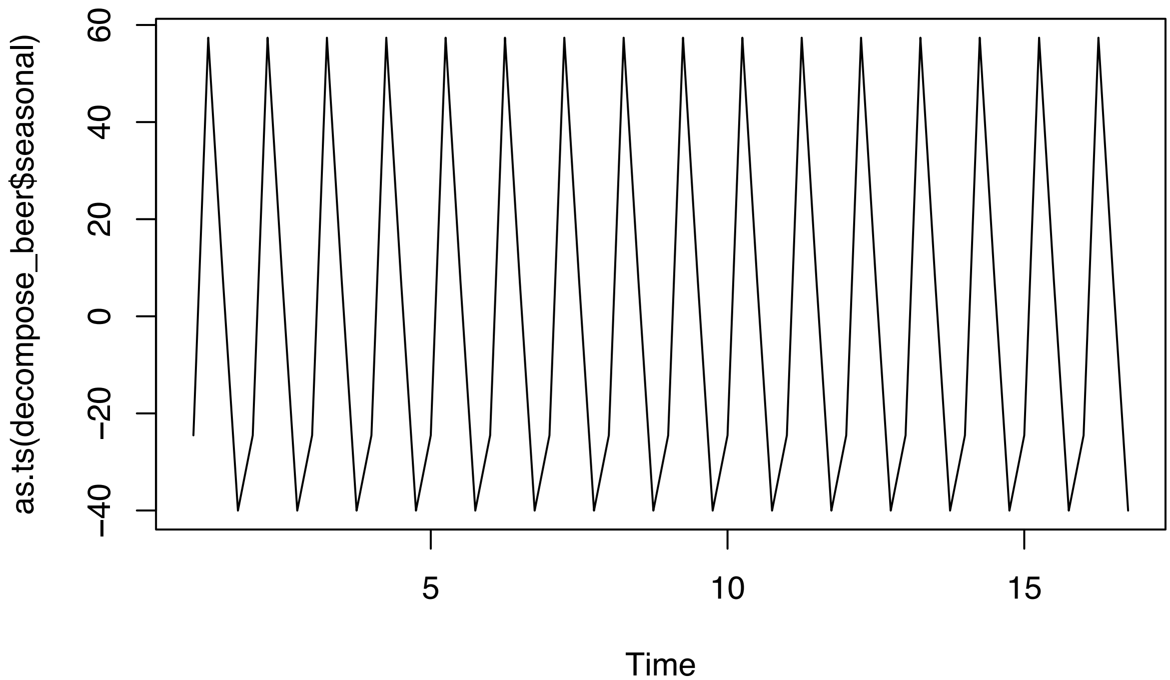
```
recomposed_air = trend_air*seasonal_air*random_air
plot(as.ts(recomposed_air))
```



Air

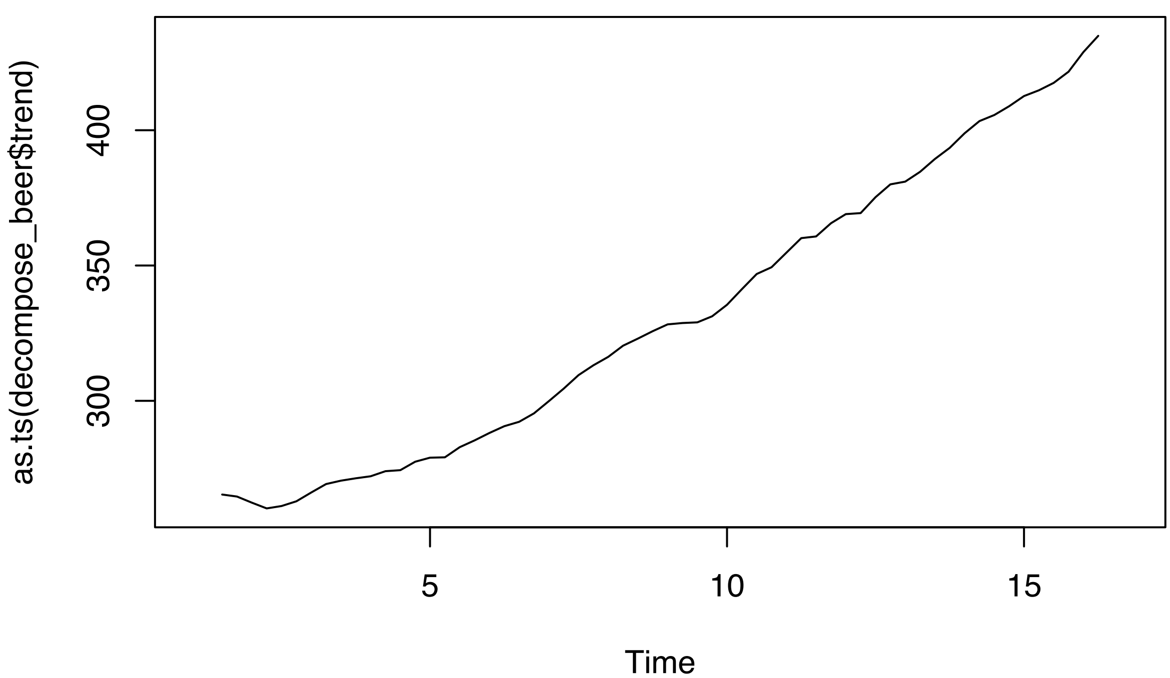
With TS

```
ts_beer = ts(timeserie_beer, frequency = 4)
decompose_beer = decompose(ts_beer, "additive")

plot(as.ts(decompose_beer$seasonal))
```

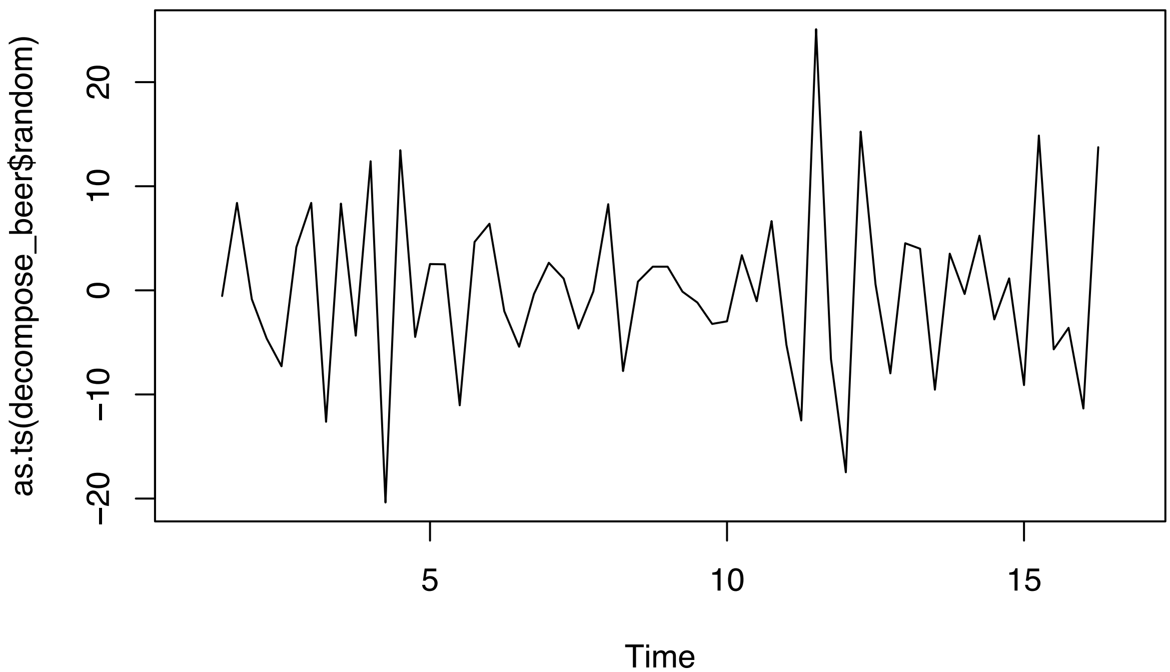


Beer

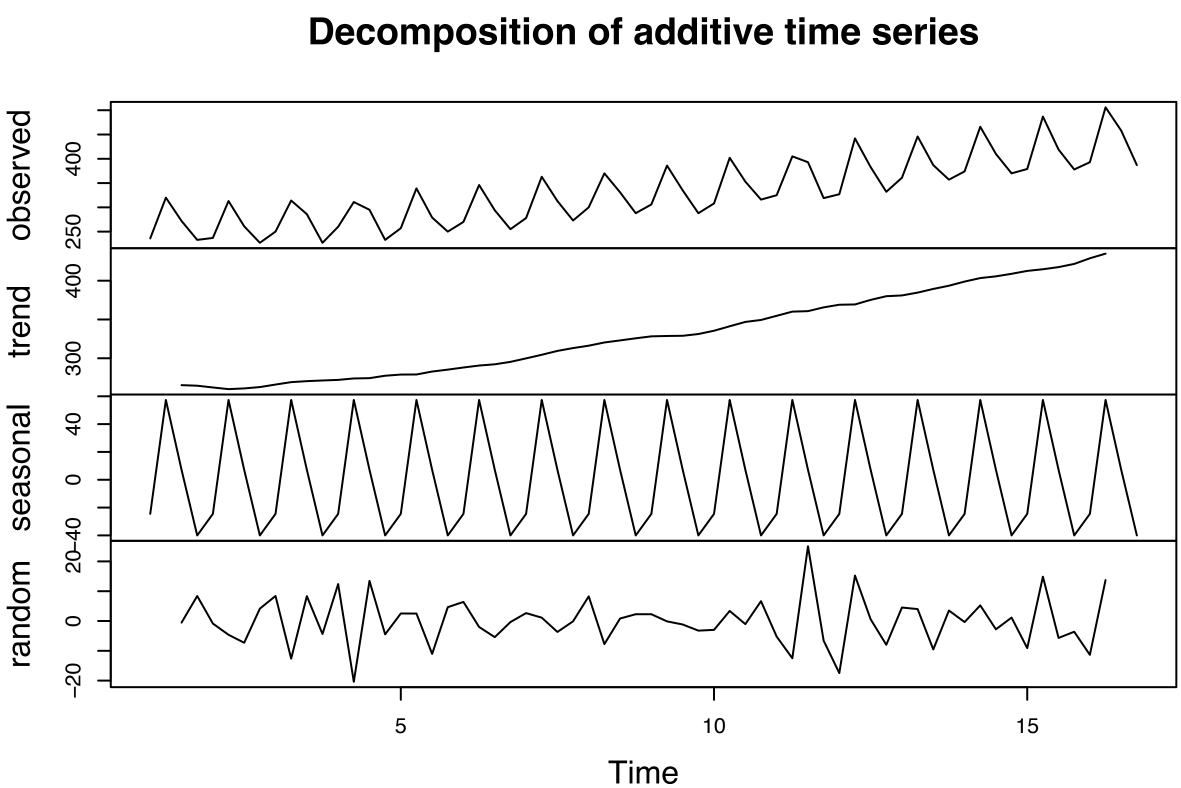
```
plot(as.ts(decompose_beer$trend))
```



```
plot(as.ts(decompose_beer$random))
```

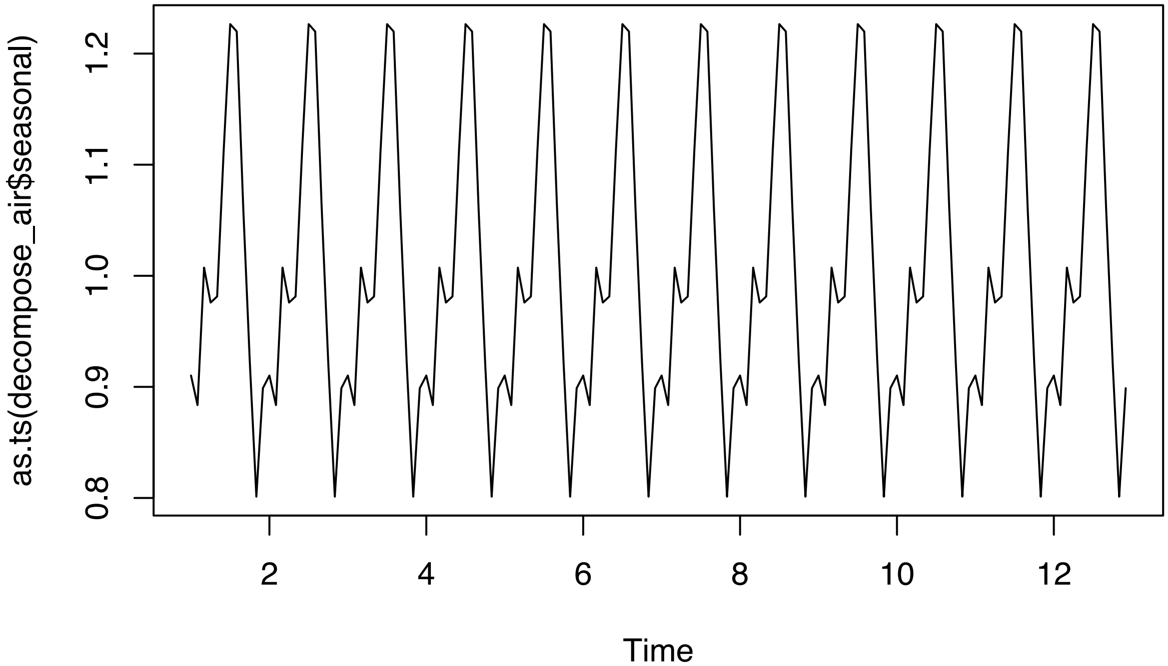



```
plot(decompose_beer)
```



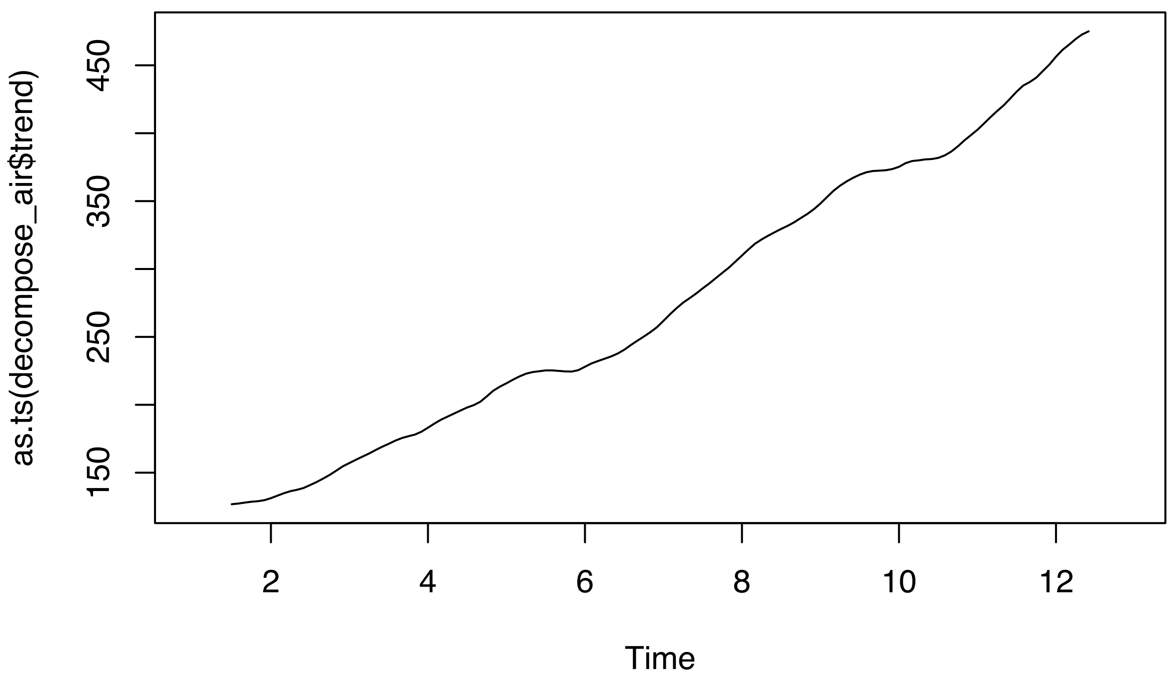

```
ts_air = ts(timeserie_air, frequency = 12)
decompose_air = decompose(ts_air, "multiplicative")
```

```
plot(as.ts(decompose_air$seasonal))
```

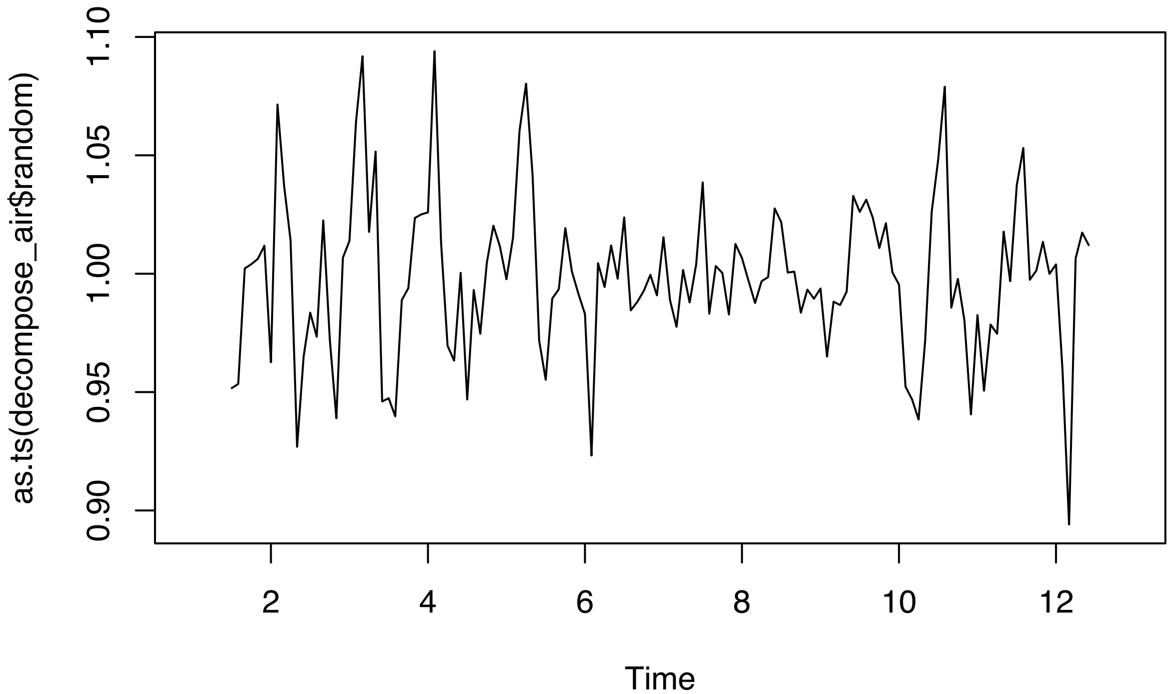


Air

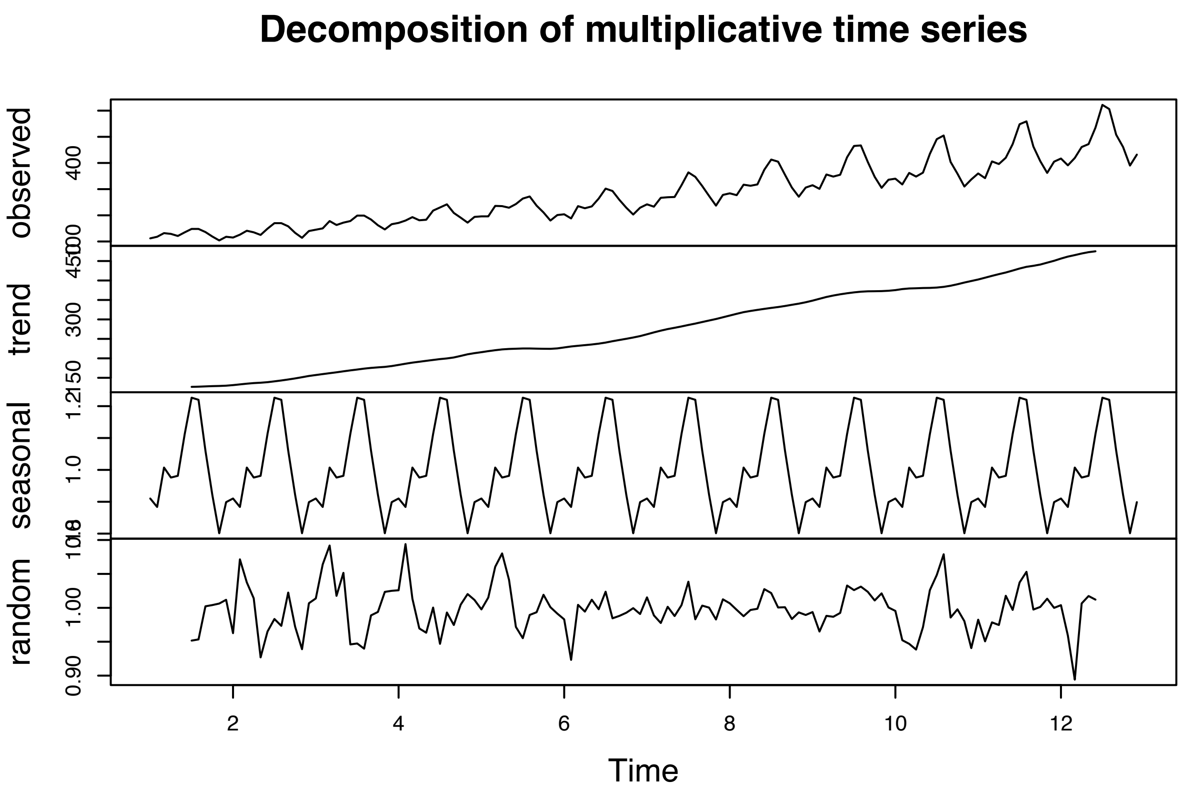
```
plot(as.ts(decompose_air$trend))
```




```
plot(as.ts(decompose_air$random))
```



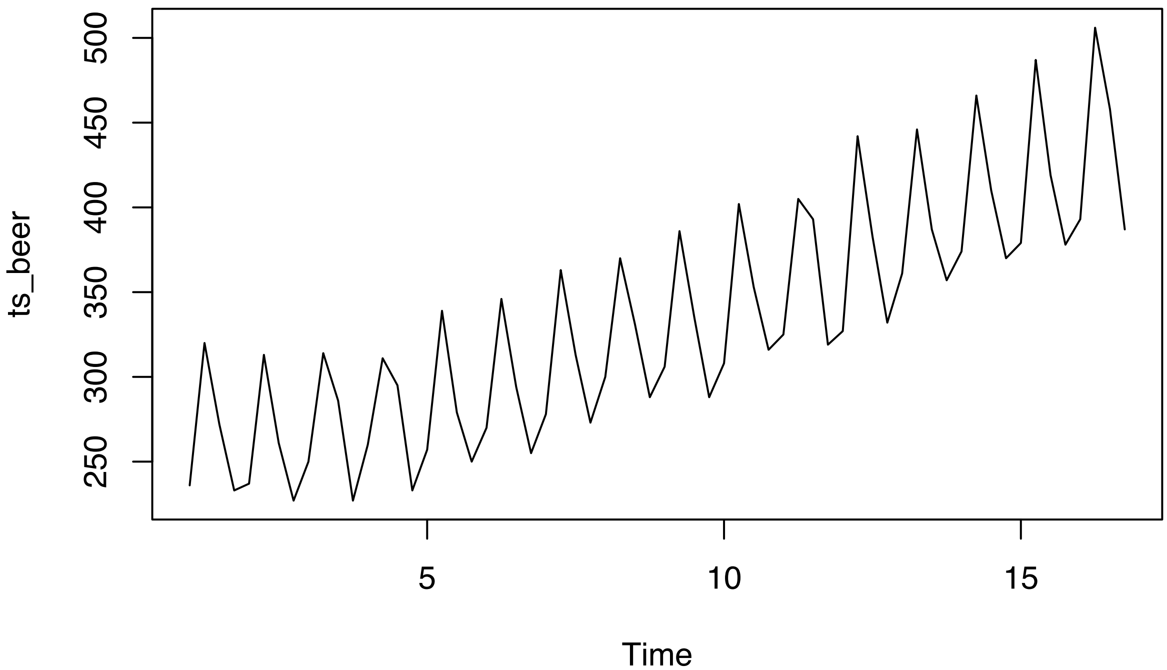

```
plot(decompose_air)
```



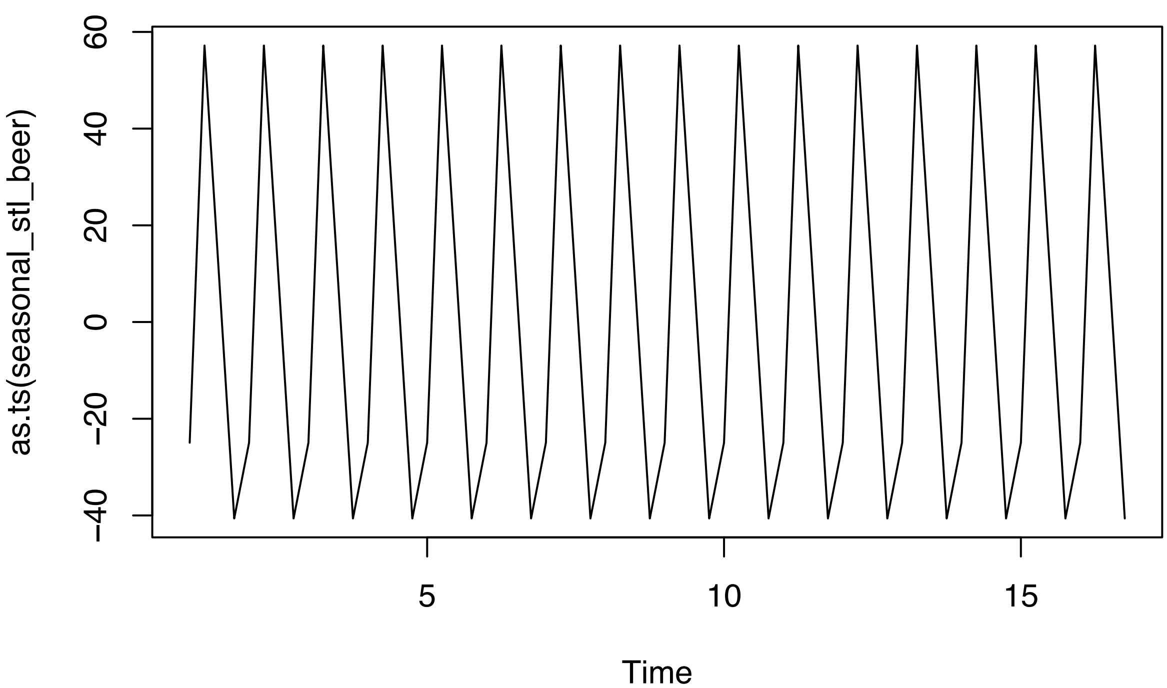
STL

```
ts_beer = ts(timeserie_beer, frequency = 4)
stl_beer = stl(ts_beer, "periodic")
```

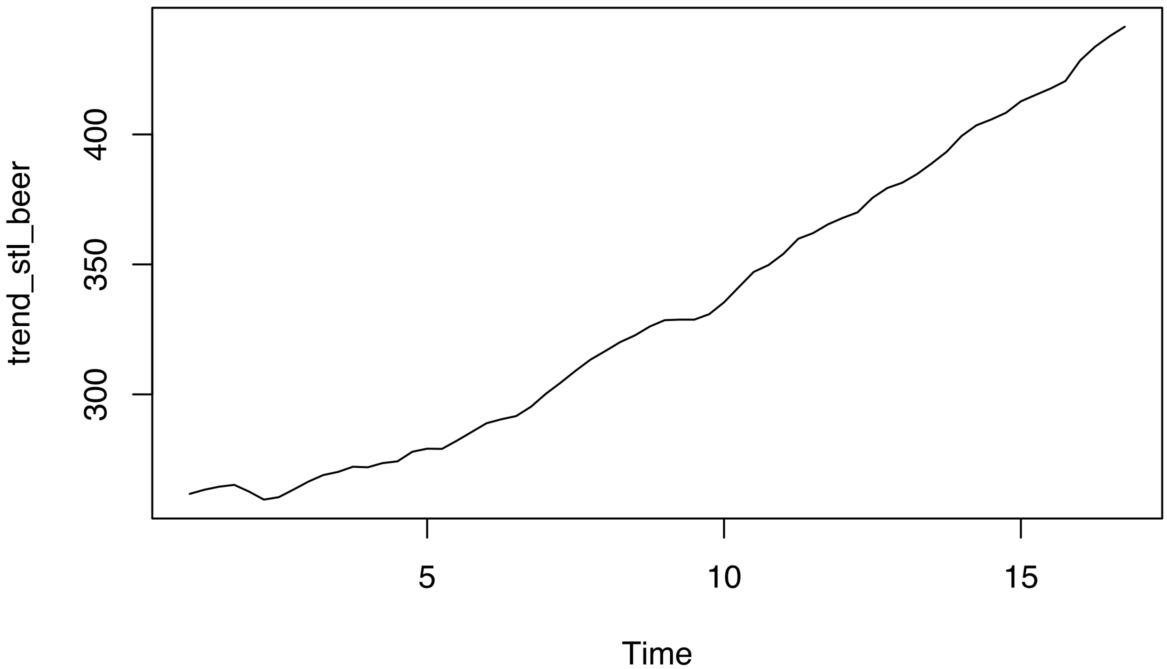
```r
seasonal_stl_beer   <- stl_beer$time.series[,1]
trend_stl_beer     <- stl_beer$time.series[,2]
random_stl_beer  <- stl_beer$time.series[,3]

plot(ts_beer)
```



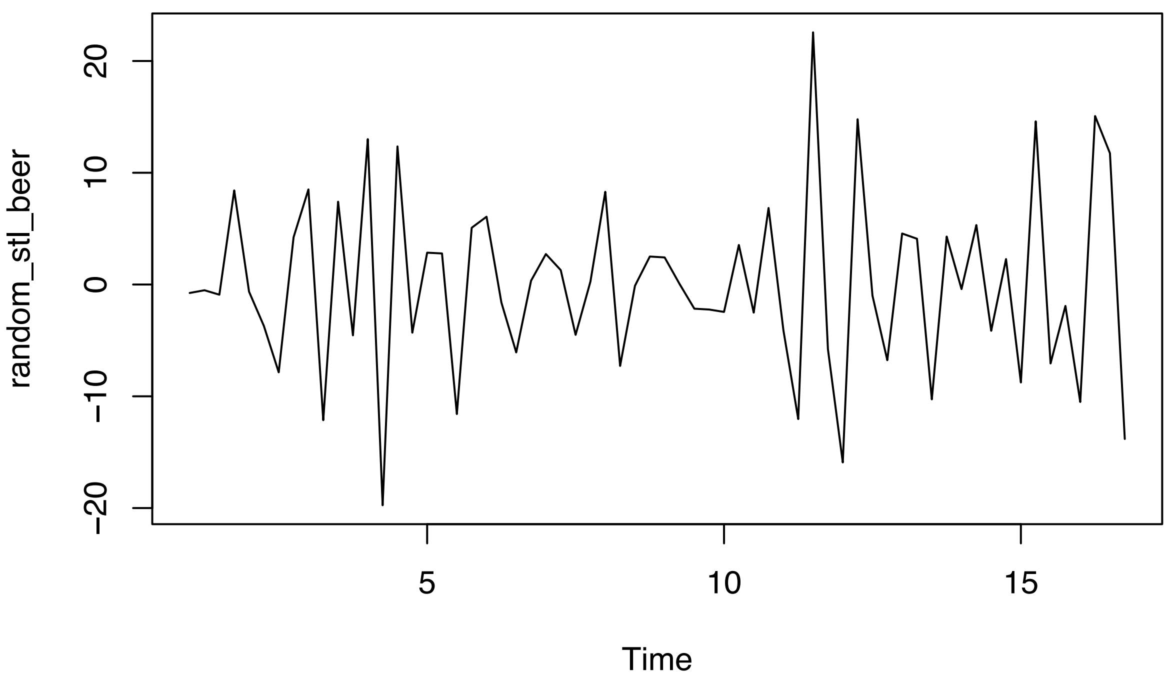

```r
plot(as.ts(seasonal_stl_beer))
```



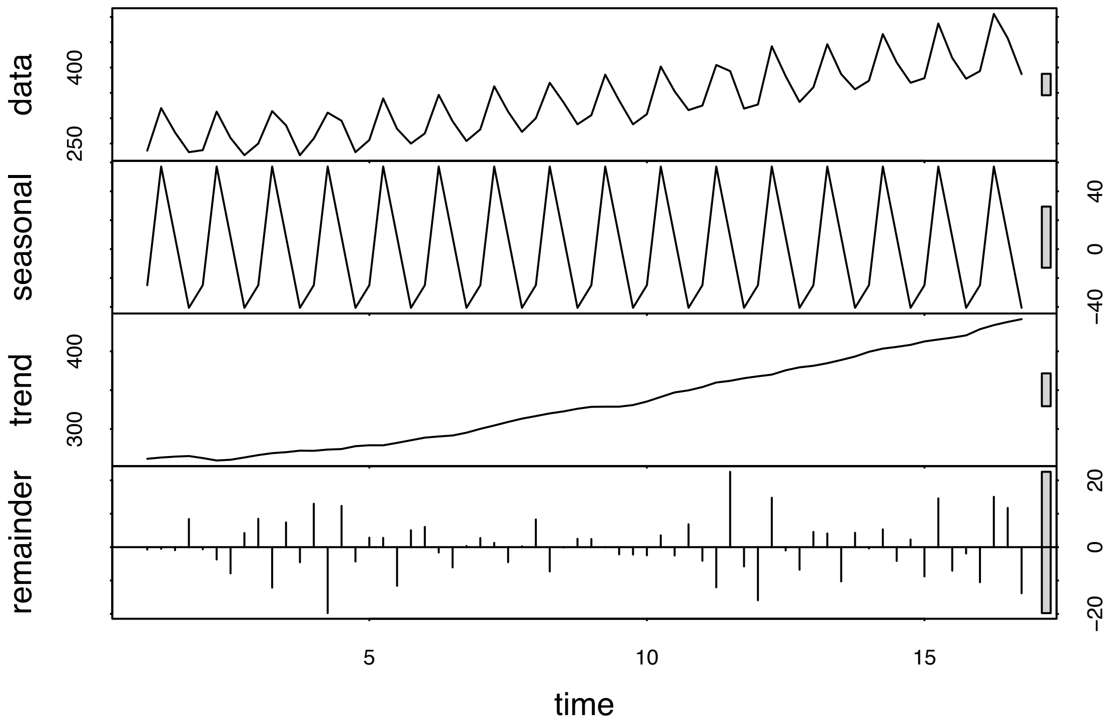
```
plot(trend_stl_beer)
```




```
plot(random_stl_beer)
```

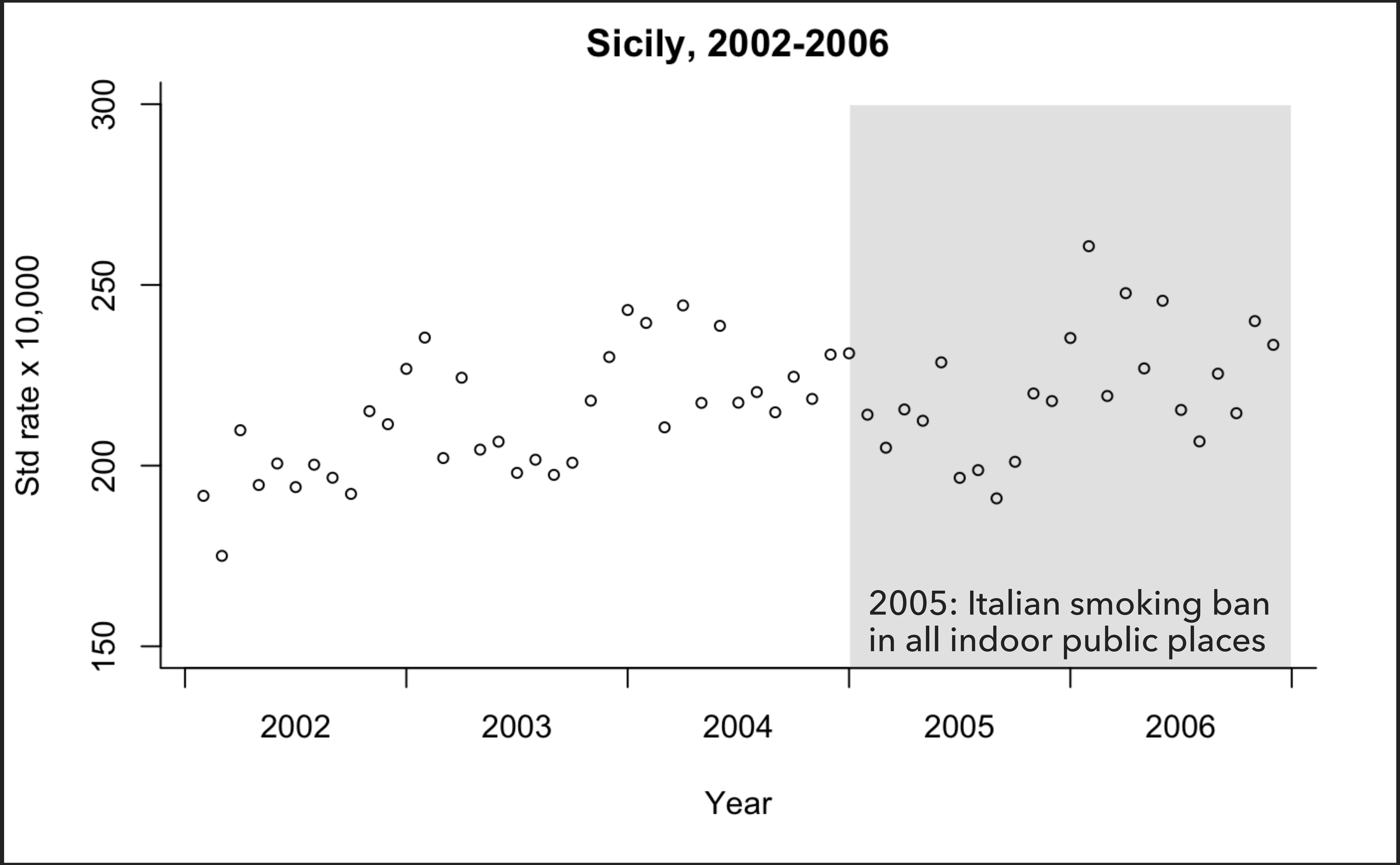



```
plot(stl_beer)
```

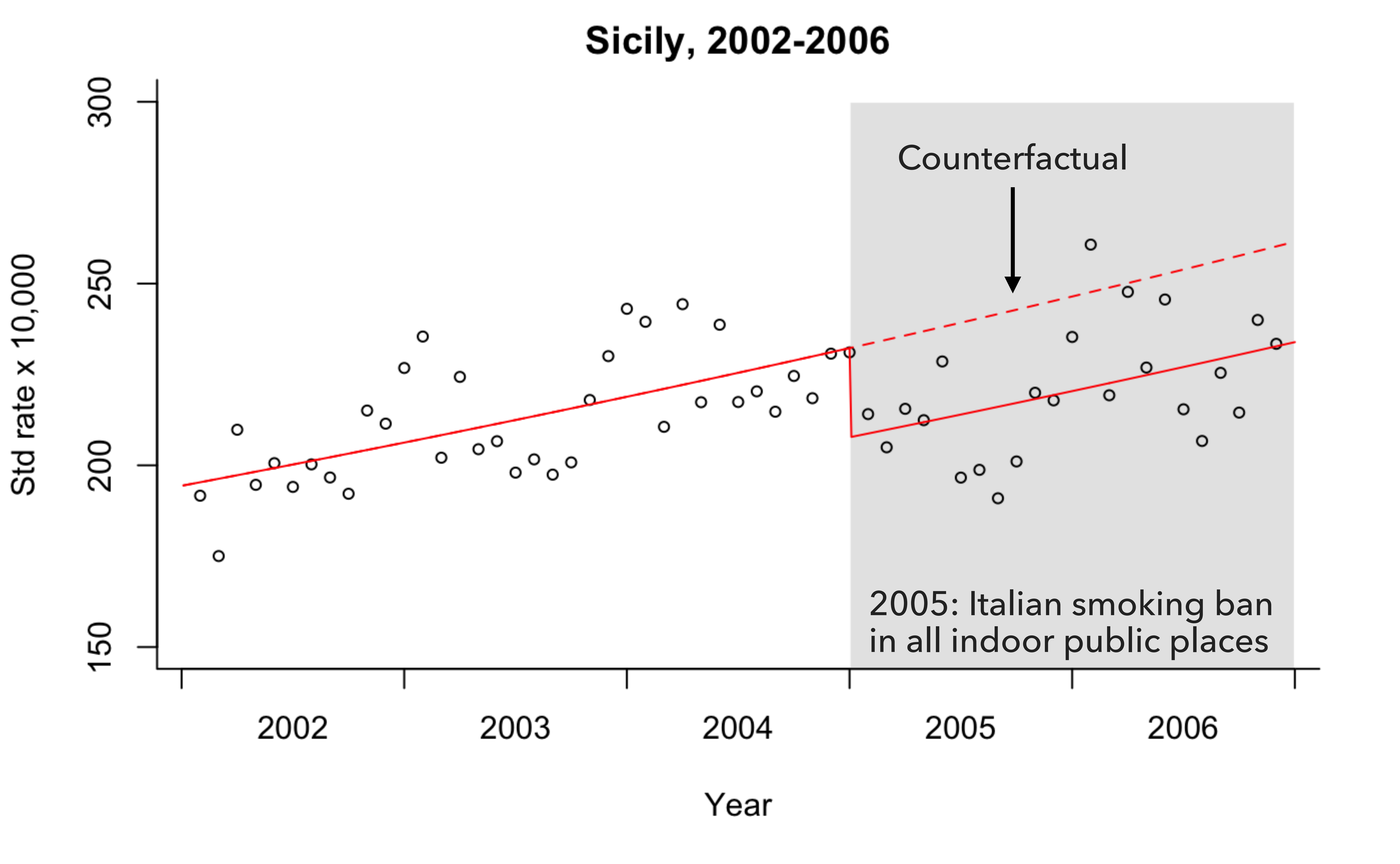

data
250
400
seasonal
40
0
−40
trend
300
400
remainder
20
0
−20
5
10
15
time

Interrupted Time Series Analysis

Hospital Admissions for Acute Coronary Events



Hospital Admissions for Acute Coronary Events

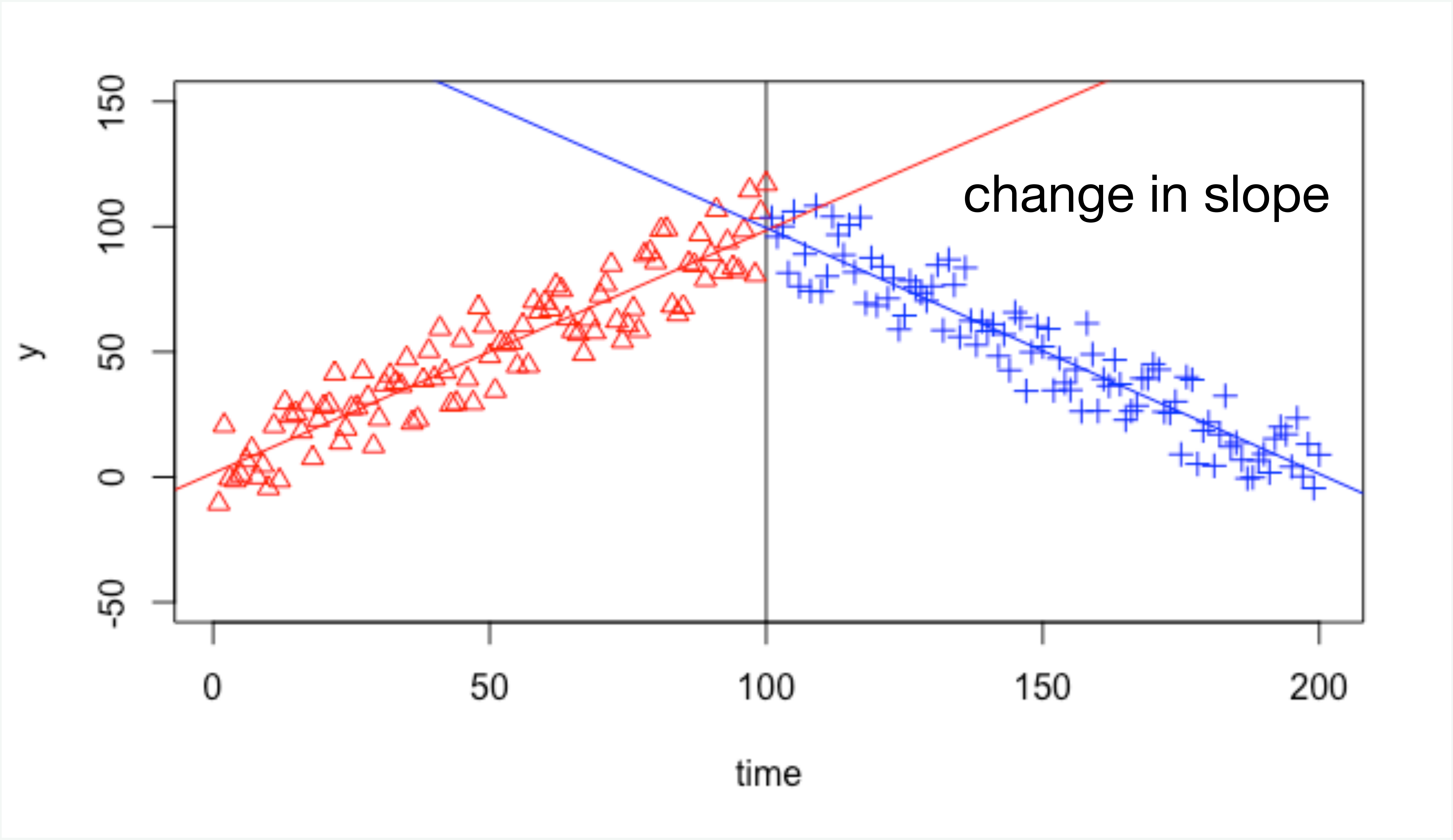


Interrupted Time Series Design

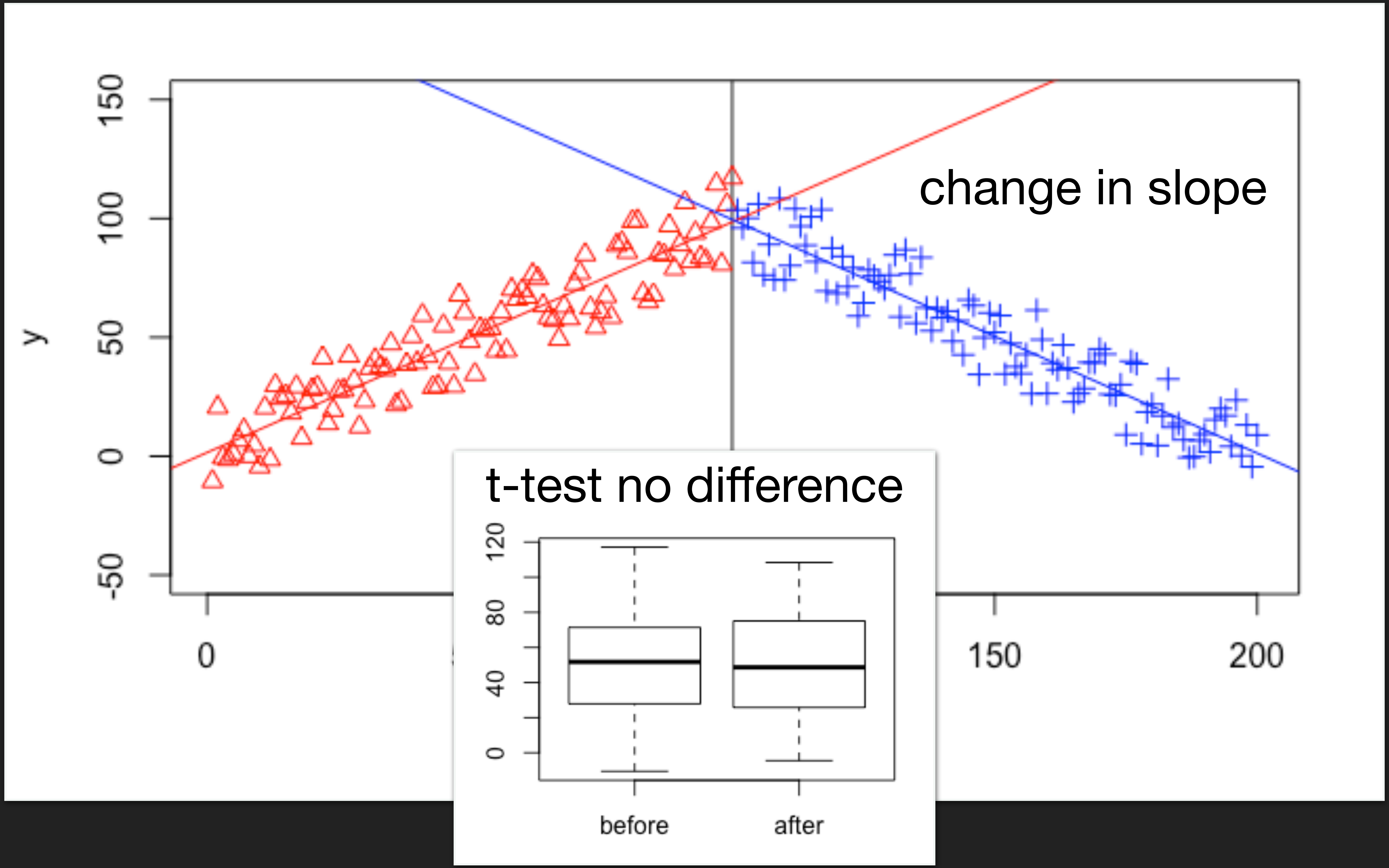
- One of the strongest quasi-experimental design to evaluate longitudinal effects of time-delimited interventions.
- How much did an intervention change an outcome of interest?
  - immediately and over time;
  - instantly or with delay;
  - transiently or long-term;
- Could factors other than the intervention explain the change?

Modeling 101

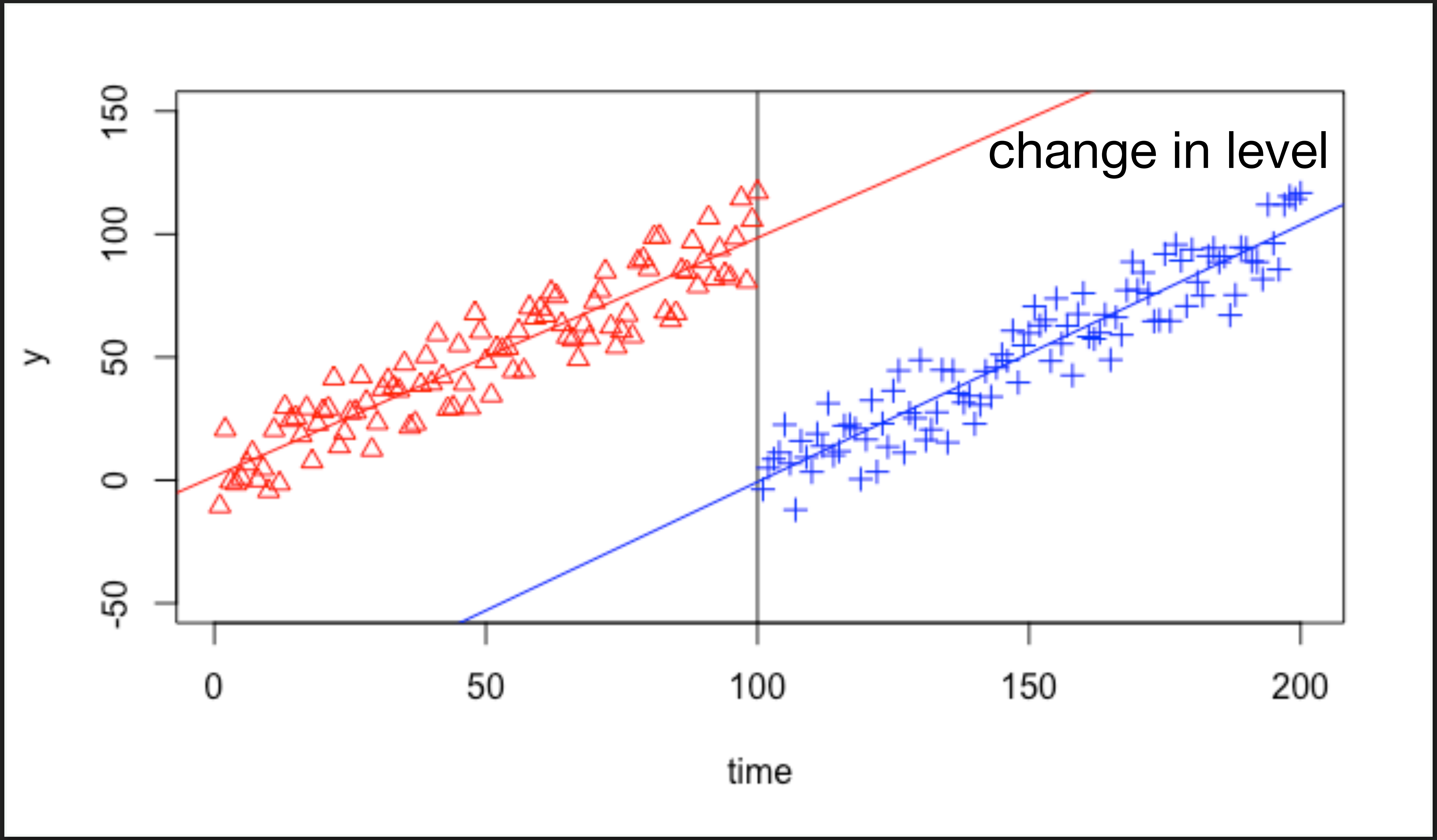
Evaluating the Effects of an Intervention



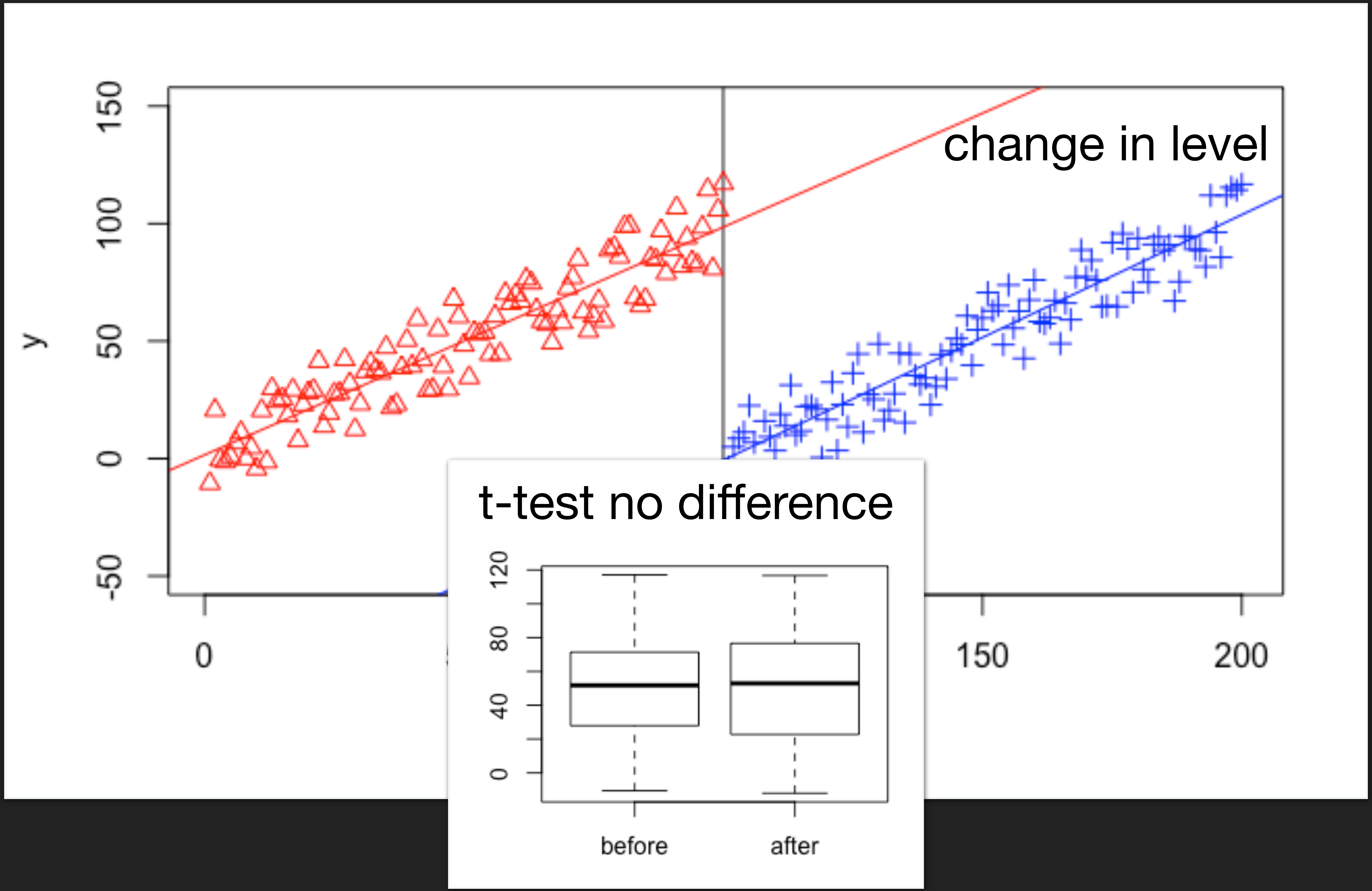
Evaluating the Effects of an Intervention



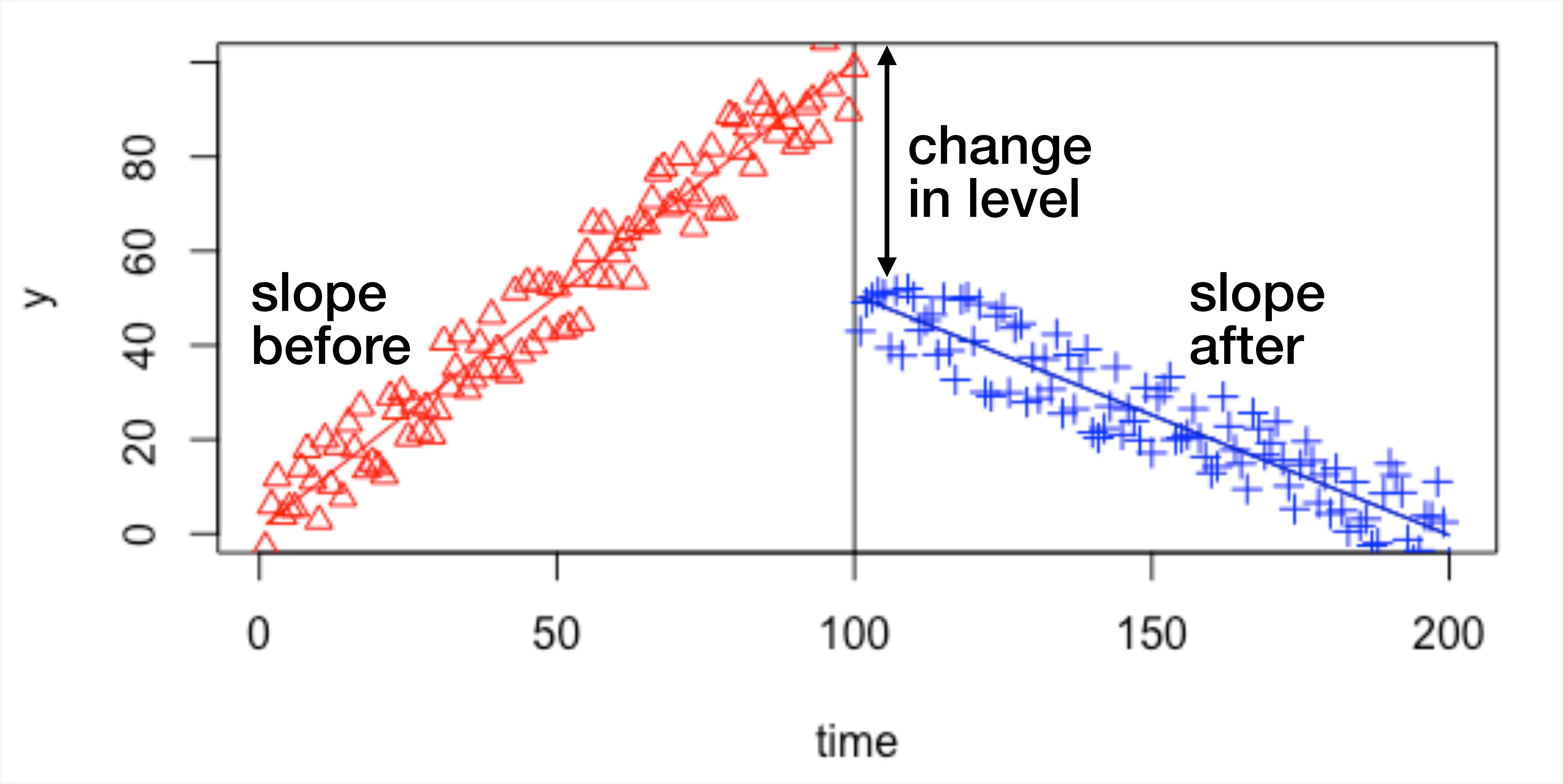
Evaluating the Effects of an Intervention



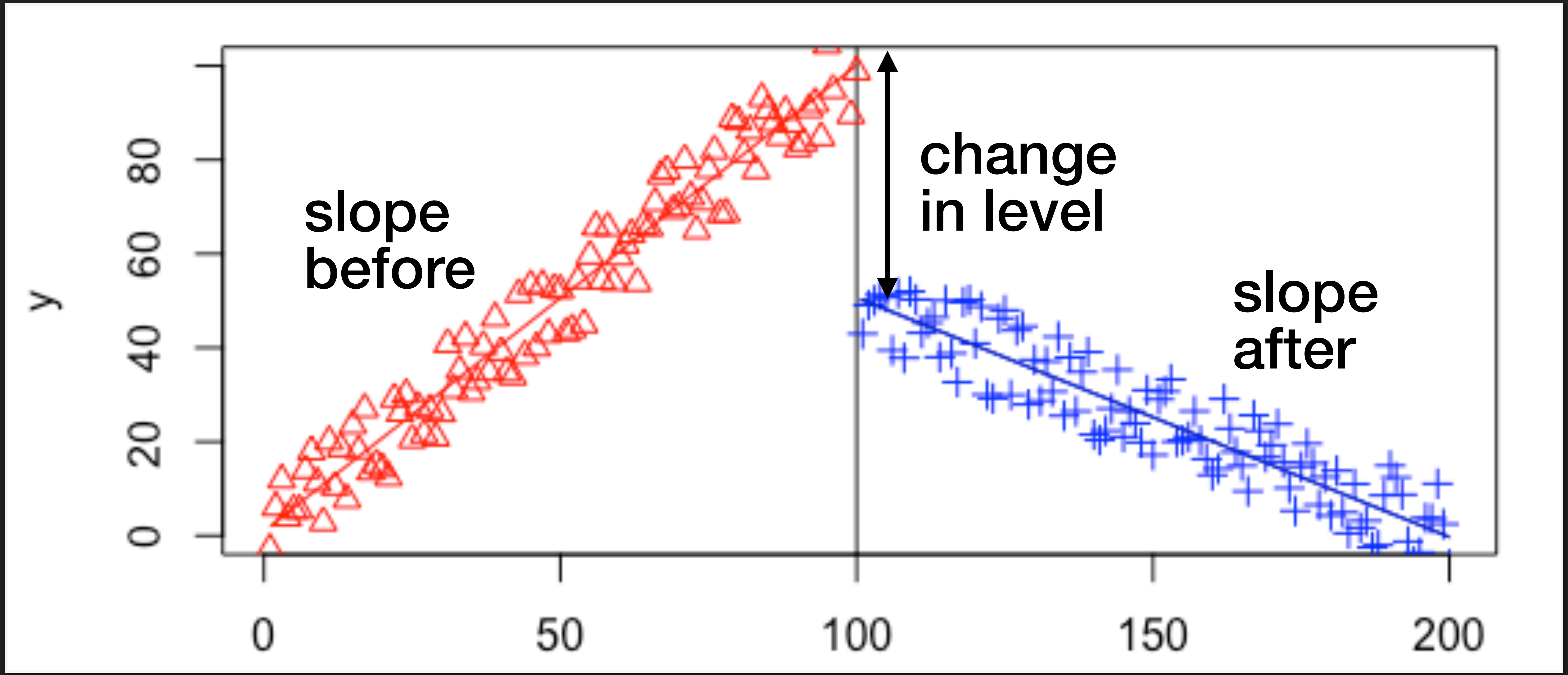
Evaluating the Effects of an Intervention



Segmented Regression Analysis of Interrupted Time Series Data



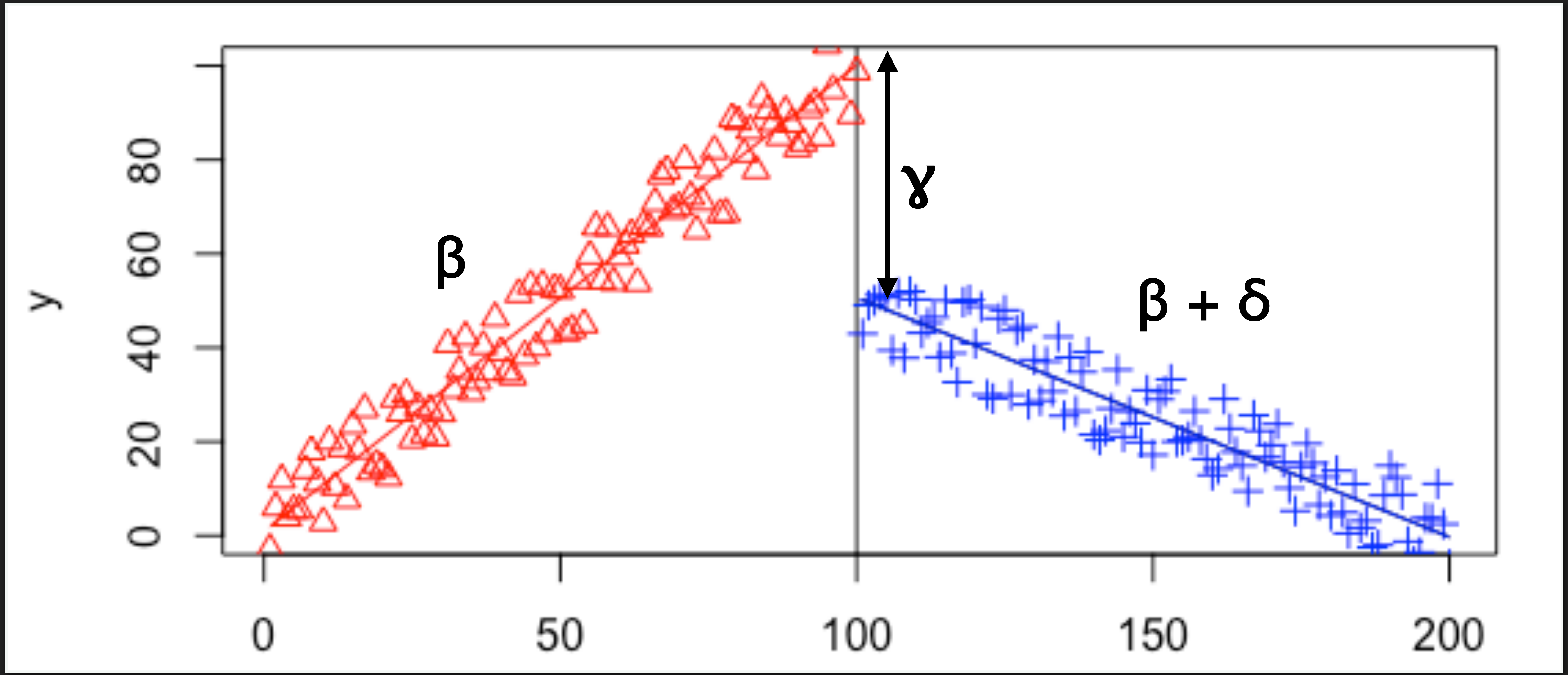
y
0
20
40
60
80
slope before
change in level
slope after
0
50
100
150
200
time



time: 1 2 3 … … … 100 101 102 … … … 200

time after intervention: 0 0 0 … … … 1 2 3 … … … 100

intervention: F F F … … … T T T … … … T



time:	1	2	3 … … …	100	101	102	… … …	200
time after intervention:	0	0	0 … … …	1	2	3	… … …	100
intervention:	F	F	F … … …	T	T	T	… … …	T

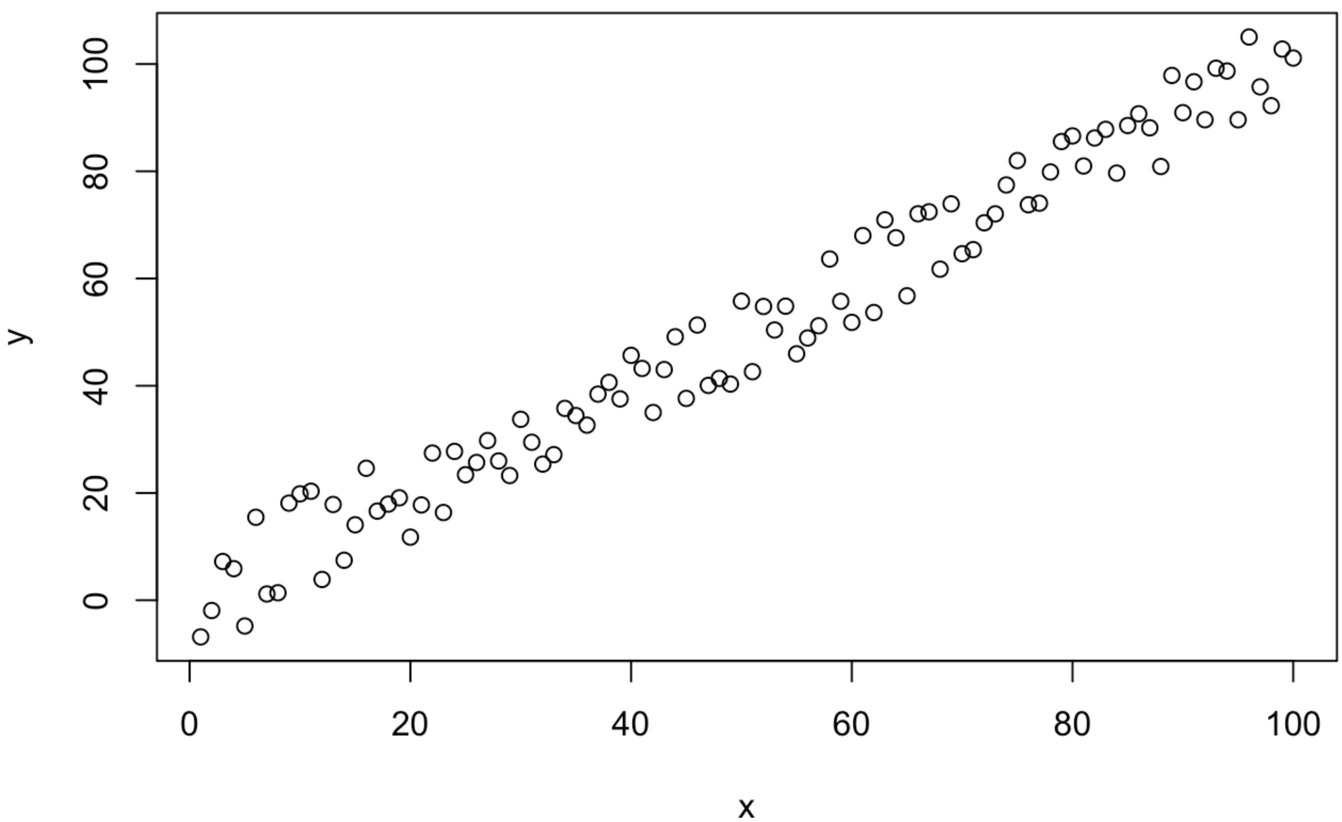
$$y_i = \alpha + \beta \cdot time_i + \gamma \cdot intervention_i + \delta \cdot time\_after\_intervention_i + \varepsilon_i$$

Interrupted Time Series

Let's create some data.

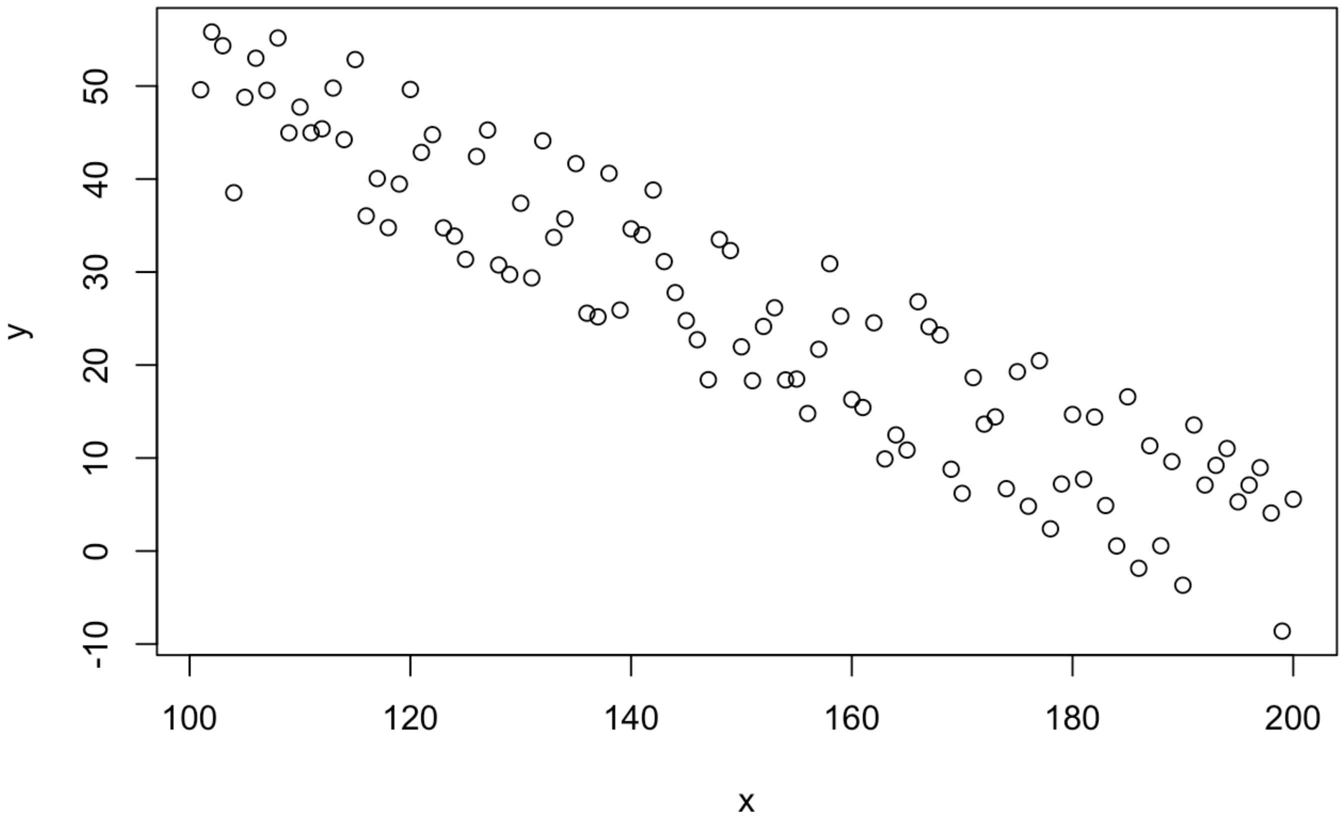
Here's a positive relationship.

```
j = 50

a = data.frame(x=1:100, y=jitter(1:100, j))
plot(a)
```



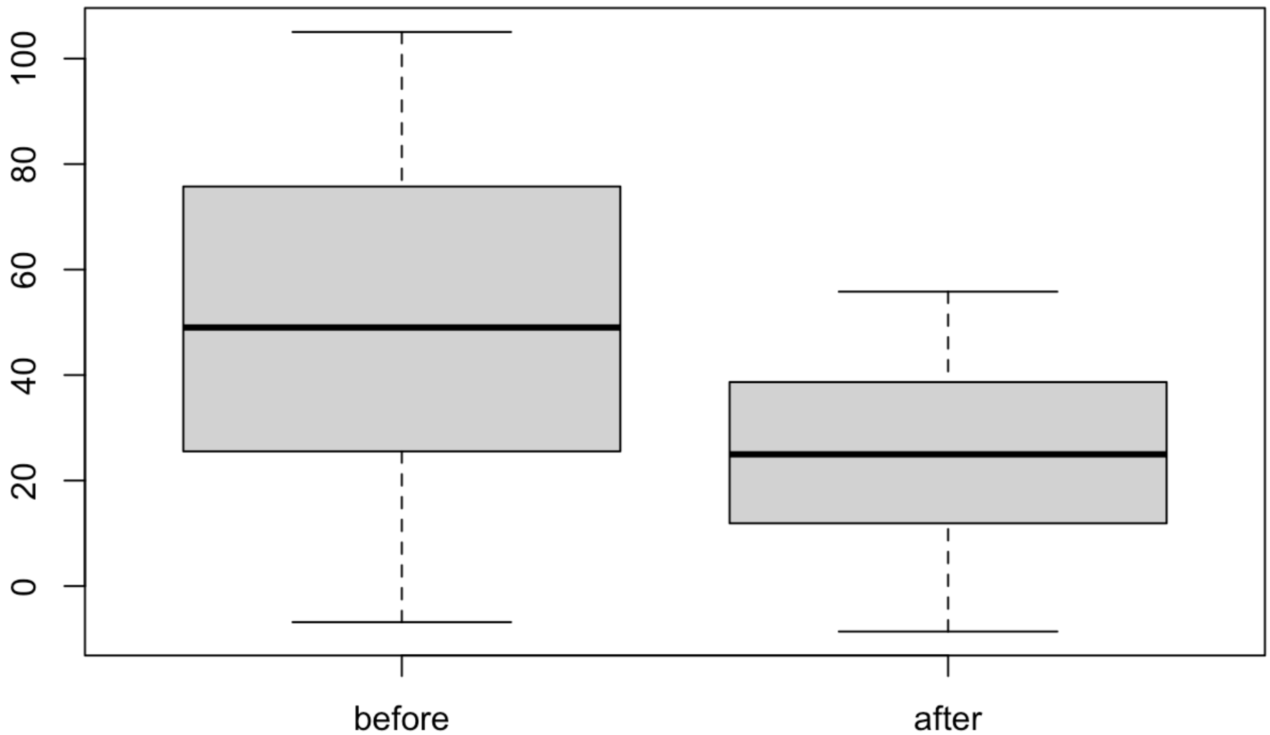
Here's a negative relationship.

```
b = data.frame(x=101:200, y=jitter(100:1, j))
bb = data.frame(x=101:200, y=jitter(seq(50,0.5,-0.5), 100))
plot(bb)
```



Are these any different?

```
boxplot(list(before=a$y,after=bb$y))
```

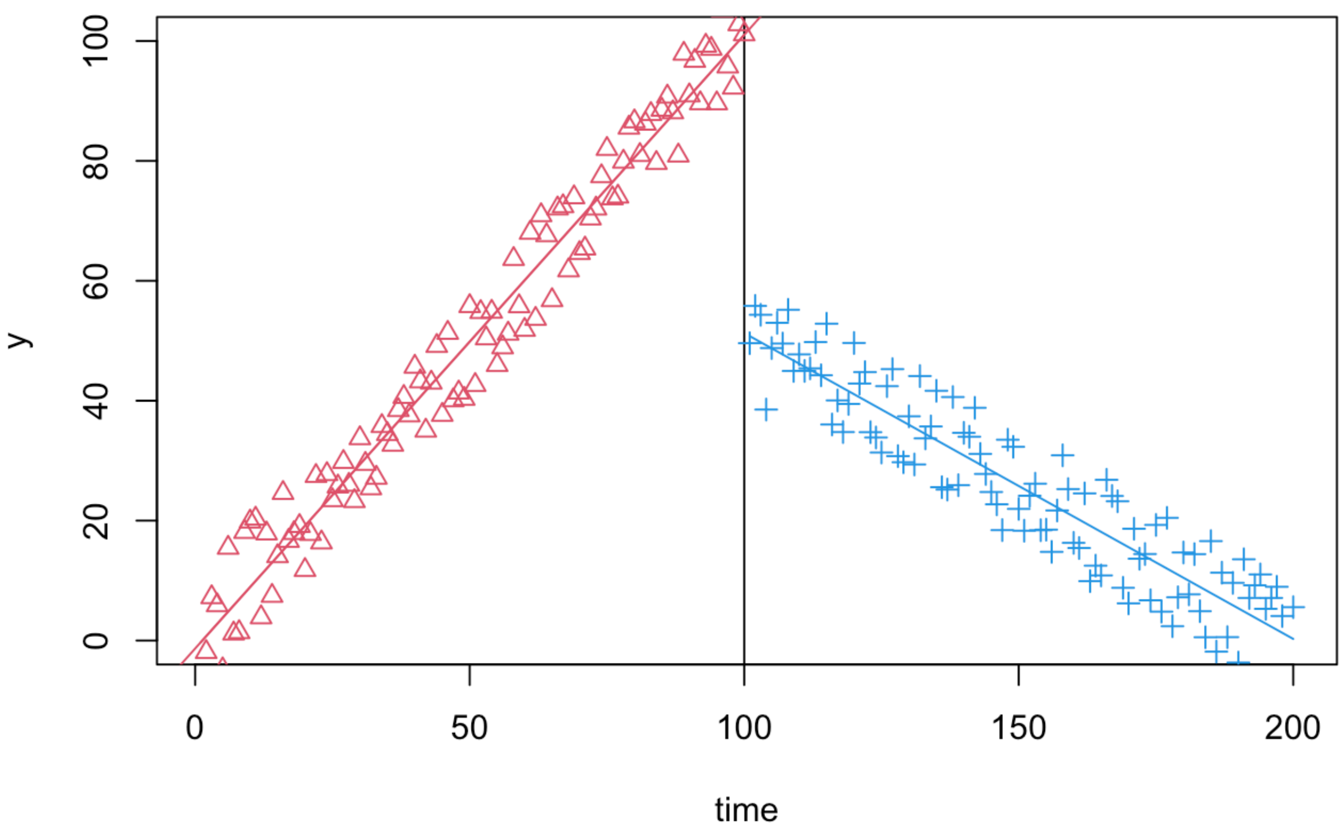



```
t.test(a$y,b$y)
```

```
##
##  Welch Two Sample t-test
##
## data:  a$y and b$y
## t = 0.0084188, df = 197.76, p-value = 0.9933
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.242917  8.313598
## sample estimates:
## mean of x mean of y
##  50.35624  50.32090
```

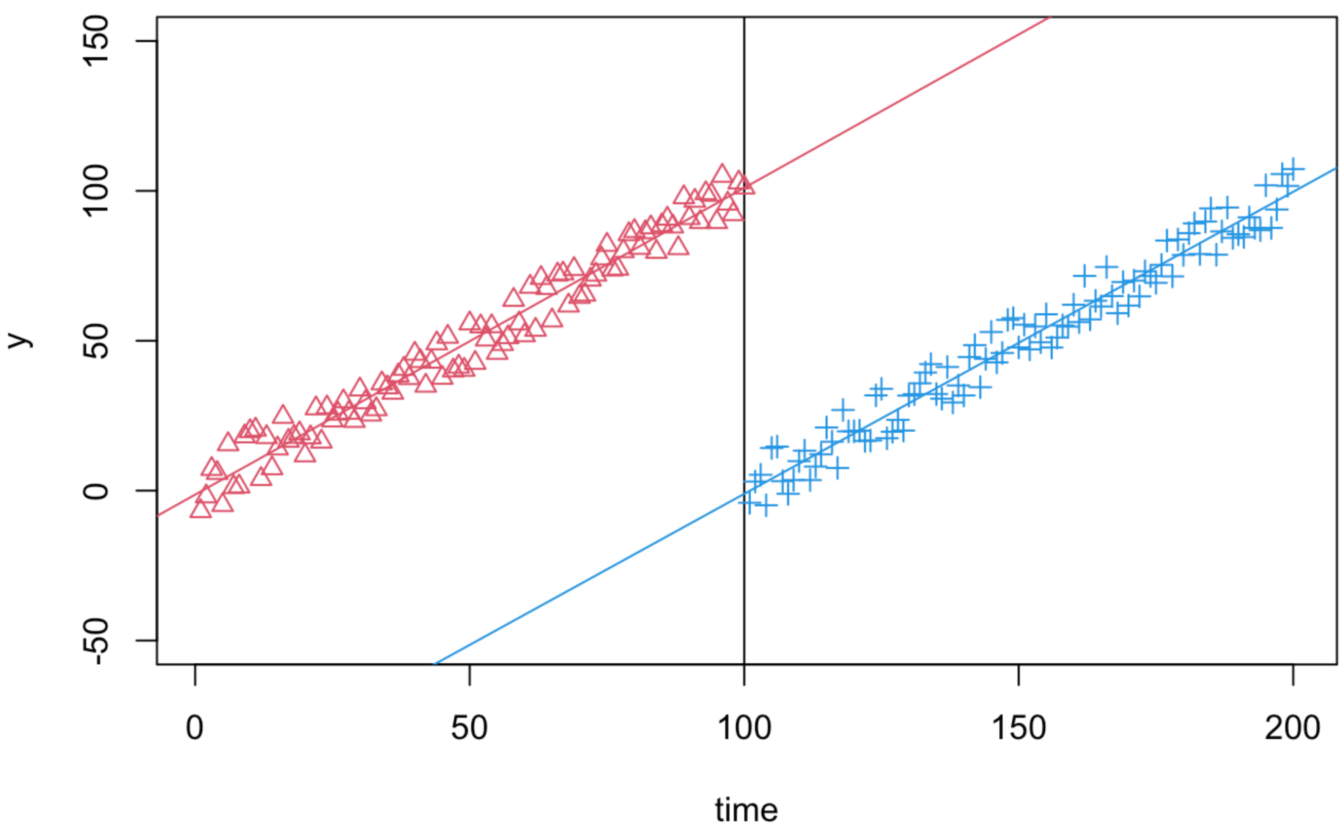
Let’s display them side by side.

```
plot(x=1:200, y=rep(1,200), type="n", ylim=c(0,100),
     xlab="time", ylab="y")
abline(v=100)
points(a$x, a$y, pch=2, col=2)
points(bb$x, bb$y, pch=3, col=4)

abline(lm(y~x, data=a), col=2)
lines(x=1:100, y=lm(y~x, data=a)$fit, col=2)
# abline(lm(y~x, data=bb), col=4)
lines(x=101:200, y=lm(y~x, data=bb)$fit, col=4)
```



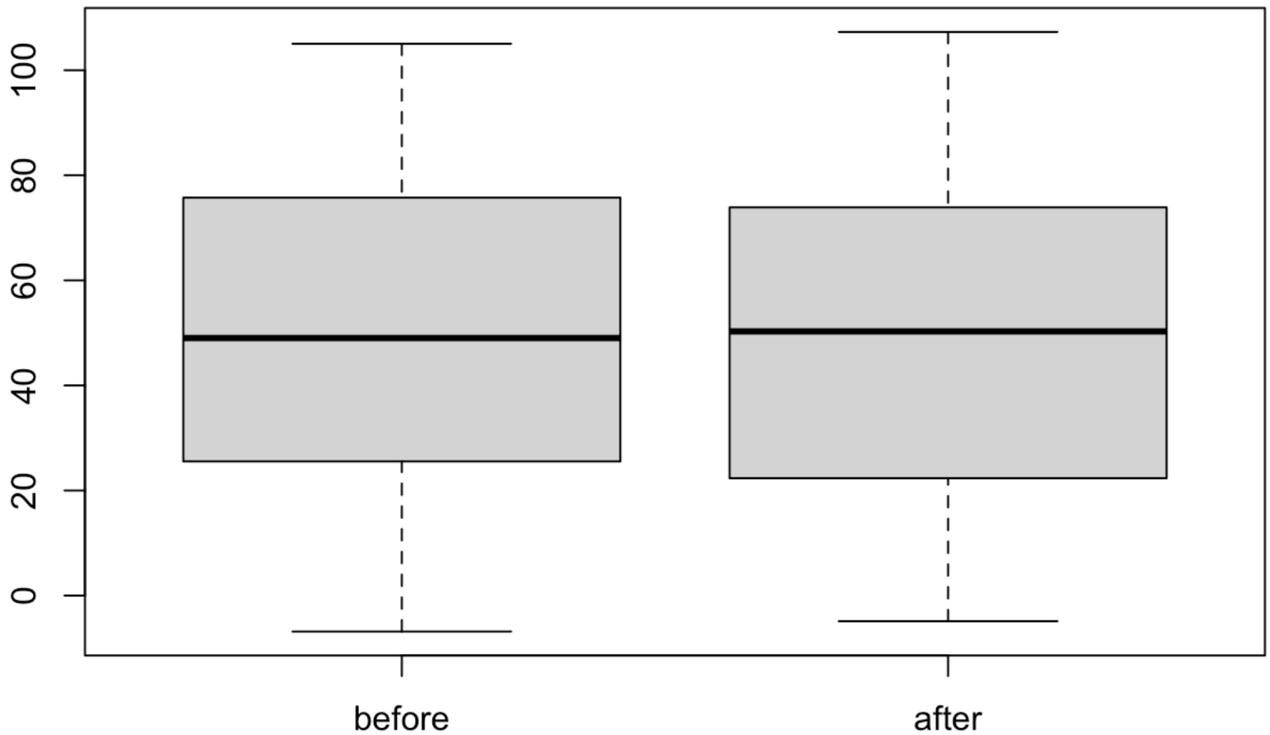
Let’s simulate a change in level.

```
a2 = data.frame(x=101:200, y=jitter(1:100, j))

plot(x=1:200, y=rep(1,200), type="n", ylim=c(-50,150),
     xlab="time", ylab="y")
abline(v=100)
points(a$x, a$y, pch=2, col=2)
points(a2$x, a2$y, pch=3, col=4)

abline(lm(y~x, data=a), col=2)
abline(lm(y~x, data=a2), col=4)
```



We can’t capture that with a simple test.

```
boxplot(list(before=a$y,after=a2$y))
```

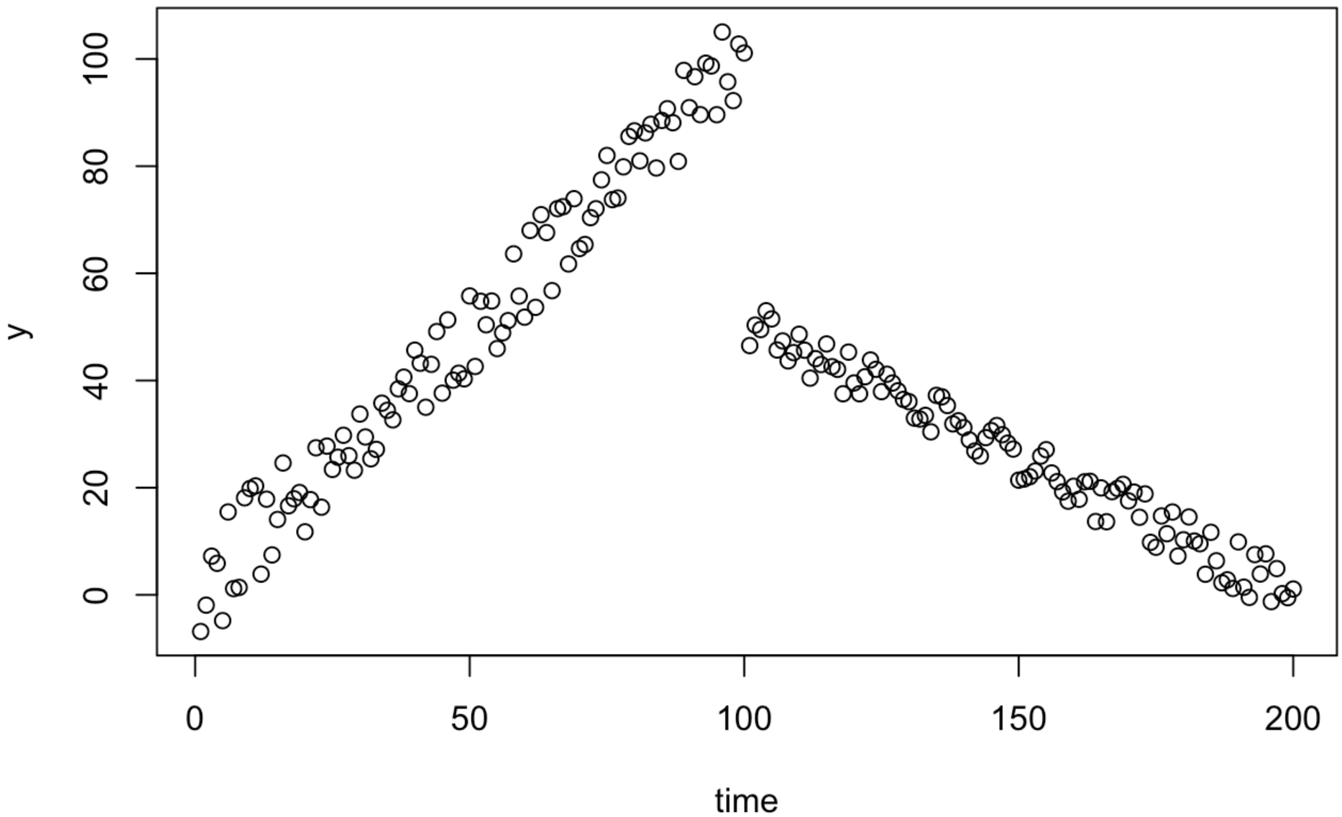



```
t.test(a$y,a2$y)
```

```
##
##  Welch Two Sample t-test
##
## data:  a$y and a2$y
## t = 0.12585, df = 197.97, p-value = 0.9
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.832732  8.900603
## sample estimates:
## mean of x mean of y
##  50.35624  49.82230
```

Now let's go back to the previous example:

```
m = rbind(a, data.frame(x=101:200, y=jitter(seq(50,0.5,-0.5), j)))
plot(m$x, m$y, xlab="time", ylab="y")
```



Here's what a simple model might look like:

```
summary(lm(y~x, data=m))
```

```
##
## Call:
## lm(formula = y ~ x, data = m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -56.927 -17.919  -3.334  10.738  66.565
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 50.20998    3.68836  13.613  < 2e-16 ***
## x           -0.12225    0.03182  -3.842 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.98 on 198 degrees of freedom
## Multiple R-squared:  0.06937,    Adjusted R-squared:  0.06467
## F-statistic: 14.76 on 1 and 198 DF,  p-value: 0.0001645
```

Let's see if we can model those trends and change in level explicitly.

```
m$time = m$x
m$intervention = m$time > 100
m$time_after_intervention = ifelse(m$time > 100, m$time - 100, 0)

m$time
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
## [109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
## [127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
## [145] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
## [163] 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## [181] 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
## [199] 199 200
```

```
m$intervention
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [109]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [121]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [133]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [145]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [157]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [169]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [181]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [193]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```
m$time_after_intervention
```

```
##   [1]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [19]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [37]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [55]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [73]   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  [91]   0   0   0   0   0   0   0   0   0   0   1   2   3   4   5   6   7   8
## [109]   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
## [127]  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44
## [145]  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62
## [163]  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
## [181]  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98
## [199]  99 100
```

```
rdd = lm(y ~ time + intervention + time_after_intervention, data=m)
summary(rdd)
```

```
##
## Call:
## lm(formula = y ~ time + intervention + time_after_intervention,
##     data = m)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.005 -3.423  0.052  3.256 10.934
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.32882    0.88844  -1.496    0.136
## time                      1.02347    0.01527  67.009   <2e-16 ***
## interventionTRUE        -49.79805    1.24712 -39.930   <2e-16 ***
## time_after_intervention  -1.53295    0.02160 -70.969   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.409 on 196 degrees of freedom
## Multiple R-squared:  0.9735, Adjusted R-squared:  0.9731
## F-statistic:  2398 on 3 and 196 DF,  p-value: < 2.2e-16
```

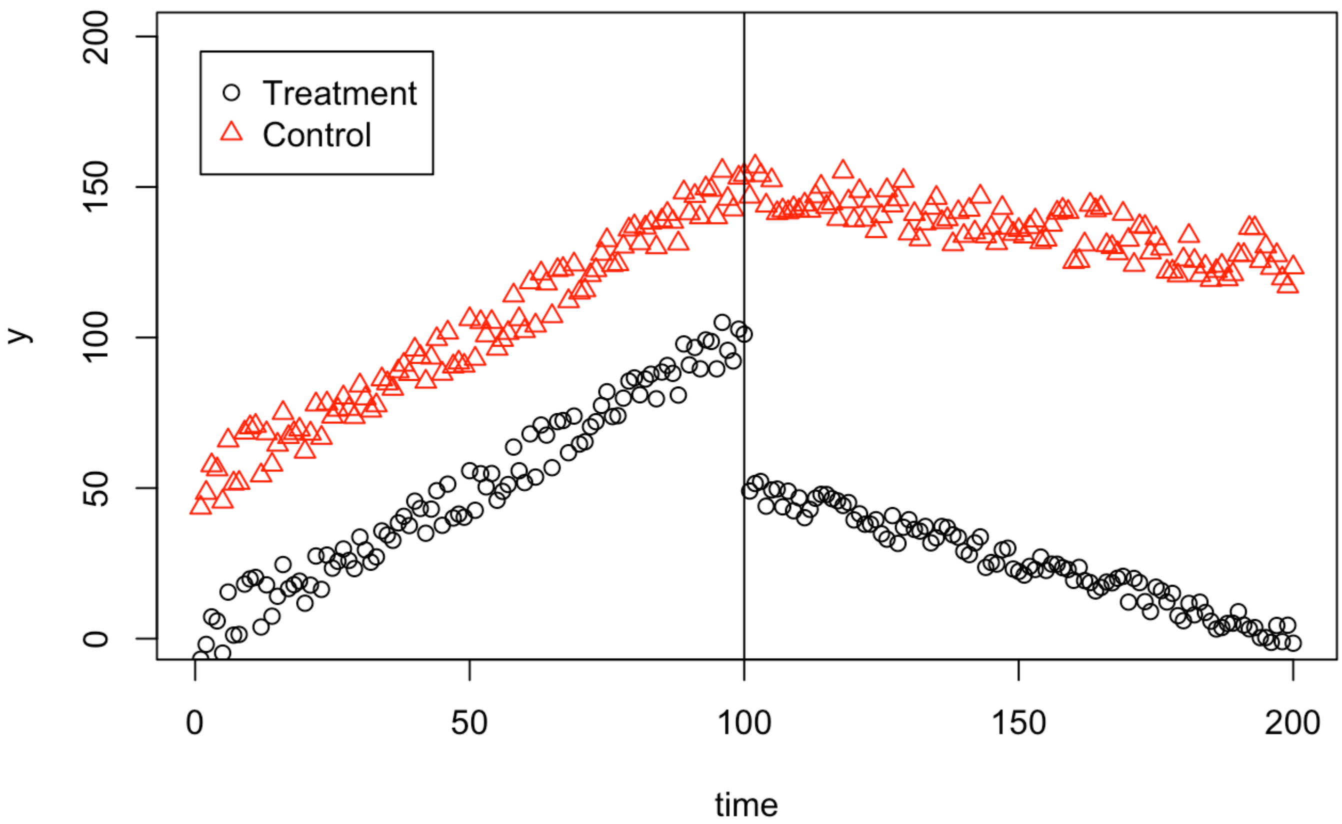
Q: Can you achieve the same result (i.e., capture both trends and the change in level) with only two variables?
A: Yes, with an interaction term!

```
rdd2 = lm(y ~ time * intervention, data=m)
summary(rdd2)
```

```
##
## Call:
## lm(formula = y ~ time * intervention, data = m)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.005 -3.423  0.052  3.256 10.934
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.32882    0.88844  -1.496    0.136
## time                     1.02347    0.01527  67.009   <2e-16 ***
## interventionTRUE       103.49739    2.50353  41.341   <2e-16 ***
## time:interventionTRUE  -1.53295    0.02160 -70.969   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.409 on 196 degrees of freedom
## Multiple R-squared:  0.9735, Adjusted R-squared:  0.9731
## F-statistic:  2398 on 3 and 196 DF,  p-value: < 2.2e-16
```

Now let's add a control series.

```
a2 = a
names(a2) = c("x","yt")
df = rbind(a2, data.frame(x=101:200, yt=jitter(seq(50,0.5,-0.5), j)))
df$yc = jitter(50) + df$yt
df[df$x>=100,]$yc = jitter(seq(150,125,-0.25), 4*j)

{
  plot(df$x, type="n", xlab="time", ylab="y")
  points(df$x, df$yt)
  points(df$x, df$yc, col = "red", pch=2)
  legend(1, 195, legend=c("Treatment", "Control"),
       col=c("black", "red"), pch=c(21,2))
  abline(v=100)
}
```



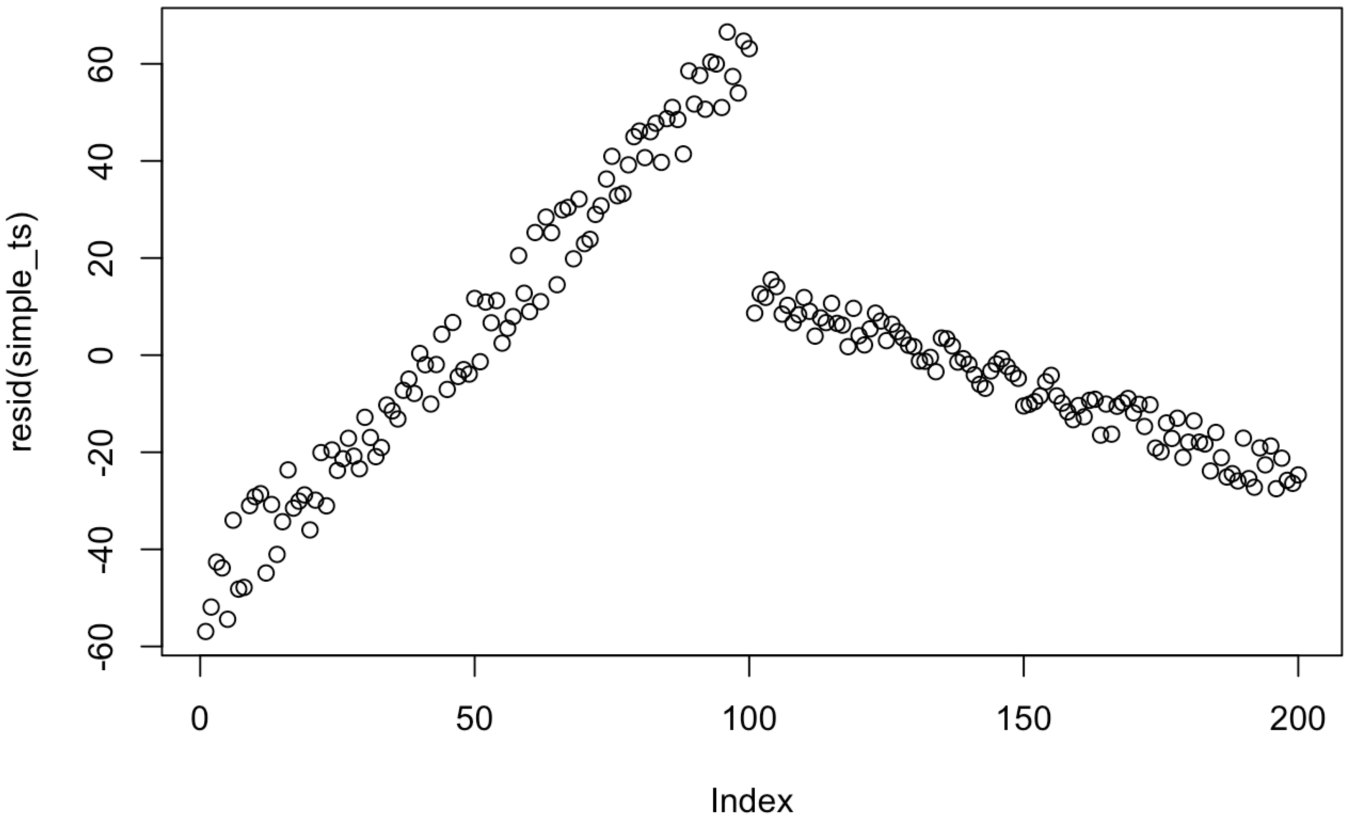
And set up the ITS variables.

```
dfm = data.frame(time = df$x, y = c(df[c("x","yt")]$yt,df[c("x","yc")]$yc))

dfm$group = c(rep("treated",200), rep("control",200))
dfm$intervention = dfm$time > 100
dfm$time_after_intervention = ifelse(dfm$time > 100, dfm$time - 100, 0)

rdd2c = lm(y ~ time
           + intervention
           + time_after_intervention
           + group
           + group:time
           + group:intervention
           + group:time_after_intervention
           , data=dfm)
summary(rdd2c)
```

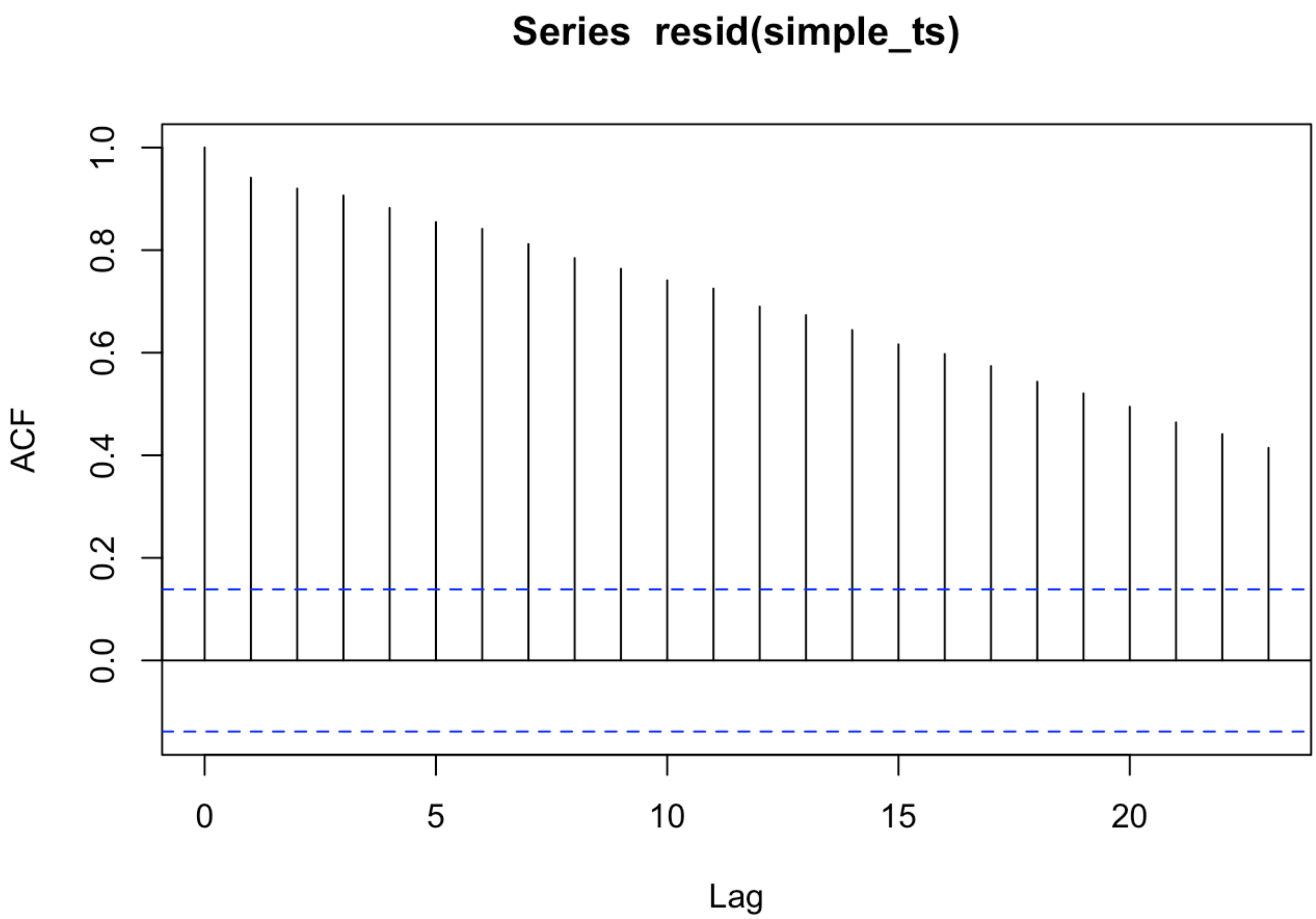
```
##
## Call:
## lm(formula = y ~ time + intervention + time_after_intervention +
##     group + group:time + group:intervention + group:time_after_intervention,
##     data = dfm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0371 -3.8688  0.0081  3.5227 11.2966
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             48.952934   1.008315  48.549  < 2e-16 ***
## time                                     1.024981   0.017335  59.129  < 2e-16 ***
## interventionTRUE                        -2.537424   1.415397  -1.793   0.0738 .
## time_after_intervention                 -1.280853   0.024515 -52.248  < 2e-16 ***
## grouptreated                           -50.281752   1.425973 -35.261  < 2e-16 ***
## time:grouptreated                       -0.001514   0.024515  -0.062   0.9508
## interventionTRUE:grouptreated          -46.944012   2.001674 -23.452  < 2e-16 ***
## time_after_intervention:grouptreated    -0.256779   0.034669  -7.407 8.02e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.004 on 392 degrees of freedom
## Multiple R-squared:  0.9897, Adjusted R-squared:  0.9895
## F-statistic:  5372 on 7 and 392 DF,  p-value: < 2.2e-16
```

Is there autocorrelation?

```
simple_ts = lm(y ~ time, data=m)
plot(resid(simple_ts))
```




```
# alternatively
acf(resid(simple_ts))
```



To formally test for autocorrelation, we can use the Durbin-Watson test

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(m$y ~ m$time)
```

```
##
##  Durbin-Watson test
##
## data:  m$y ~ m$time
## DW = 0.088829, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

From the p-value, we know that there is autocorrelation in the time series

A solution to this problem could be to use more advanced time series analysis (e.g., ARIMA) to adjust for seasonality and other dependency, or to use mixed-effects models when modeling multiple individual "treated" time series jointly.

One more example:
The Florida “Stand your ground” paper

Debate Around "Stand Your Ground" Laws

- Self-defense laws, removing the duty to retreat and allowing the use of lethal force in situations (inside and outside the home) where an individual perceives a threat of harm.
- Advocates:
  - the increased threat of retaliatory violence deters would-be burglars.
- Critics:
  - weakening the punitive consequences of using force may serve to escalate aggressive encounters.

Box. States That Have Enacted "Stand Your Ground" Laws[a]

State Name (Year Original Law Signed)
Utah (1994)[b]
Florida (2005)
Alabama (2006)
Alaska (2006)
Arizona (2006)
Georgia (2006)
Indiana (2006)
Kansas (2006)
Kentucky (2006)
Louisiana (2006)
Michigan (2006)
Mississippi (2006)
Oklahoma (2006)
South Carolina (2006)
South Dakota (2006)
Tennessee (2007)
Texas (2007)
West Virginia (2008)
Montana (2009)
Nevada (2011)
New Hampshire (2011)
North Carolina (2011)
Pennsylvania (2011)

Florida Natural Experiment

- Florida was the first state to implement a stand your ground law, removing the duty to retreat principle.
- Idea: Use the years that have elapsed since the enactment of the Florida law to assess its impact on rates of homicide and homicide by firearm.

Box. States That Have Enacted "Stand Your Ground" Laws[a]

State Name (Year Original Law Signed)

Utah (1994)[b]
Florida (2005)
Alabama (2006)
Alaska (2006)
Arizona (2006)
Georgia (2006)
Indiana (2006)
Kansas (2006)
Kentucky (2006)
Louisiana (2006)
Michigan (2006)
Mississippi (2006)
Oklahoma (2006)
South Carolina (2006)
South Dakota (2006)
Tennessee (2007)
Texas (2007)
West Virginia (2008)
Montana (2009)
Nevada (2011)
New Hampshire (2011)
North Carolina (2011)
Pennsylvania (2011)

Potential Limitations of Interrupted Time Series Designs

- The possibility that other factors that occur simultaneously may distort estimates of intervention effects, e.g.,
  - national changes in social or economic variables (e.g., a recession)
  - events that have a profound and lasting impact on society (e.g., natural disasters).

- Study design features to address limitations:
  - analysis of homicide rates in 4 comparison states (New York, New Jersey, Ohio, and Virginia),
  - analysis of control outcomes (suicide and suicide by firearm).

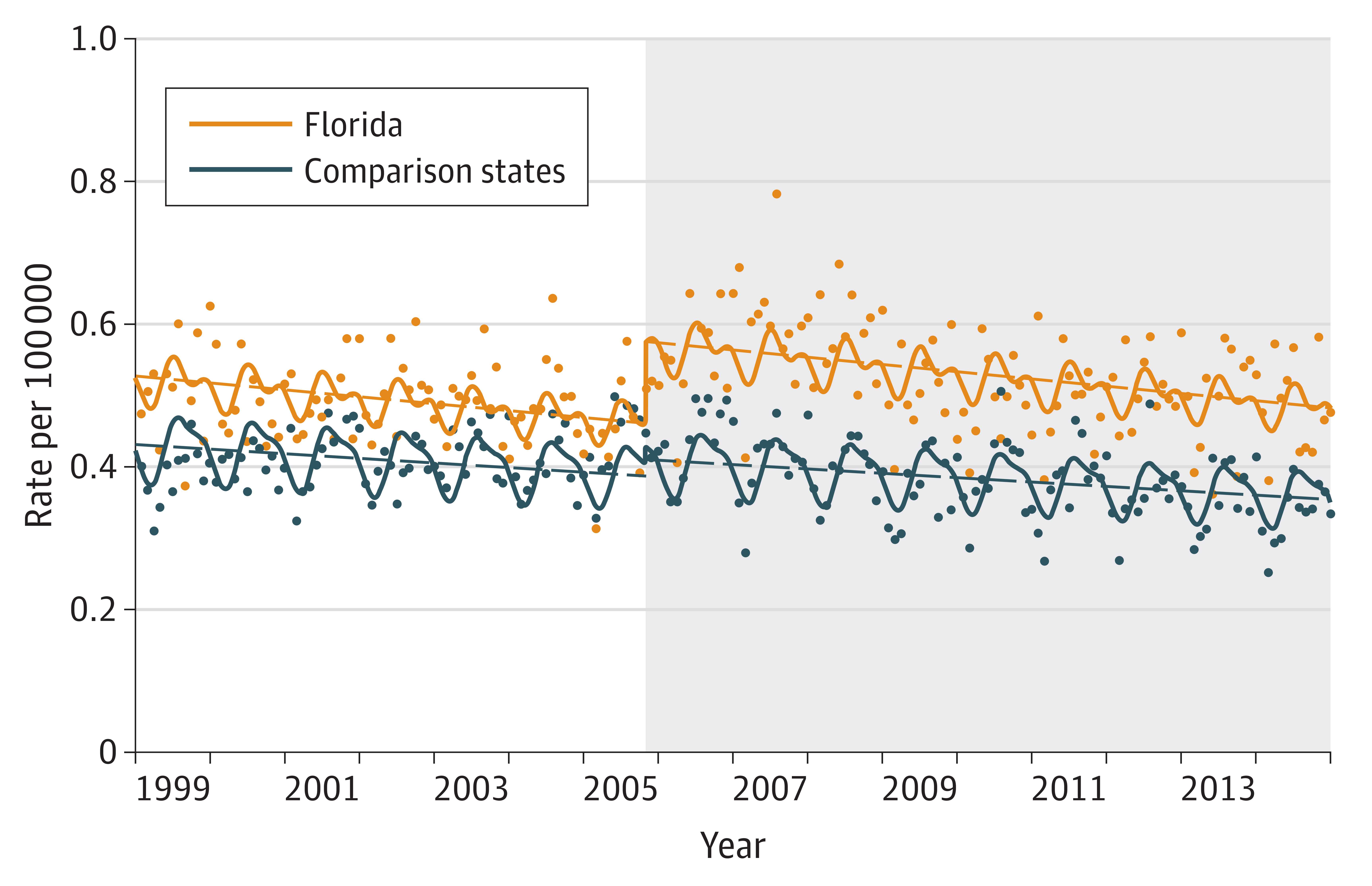
Data Sources

- Monthly death totals for Florida between Jan 1999 and Dec 2014, from CDC.
- Classified cases by:
  - place of occurrence (within or outside the State of Florida),
  - cause of death (homicide or suicide),
  - mechanism (firearms or other means), and
  - month of occurrence.

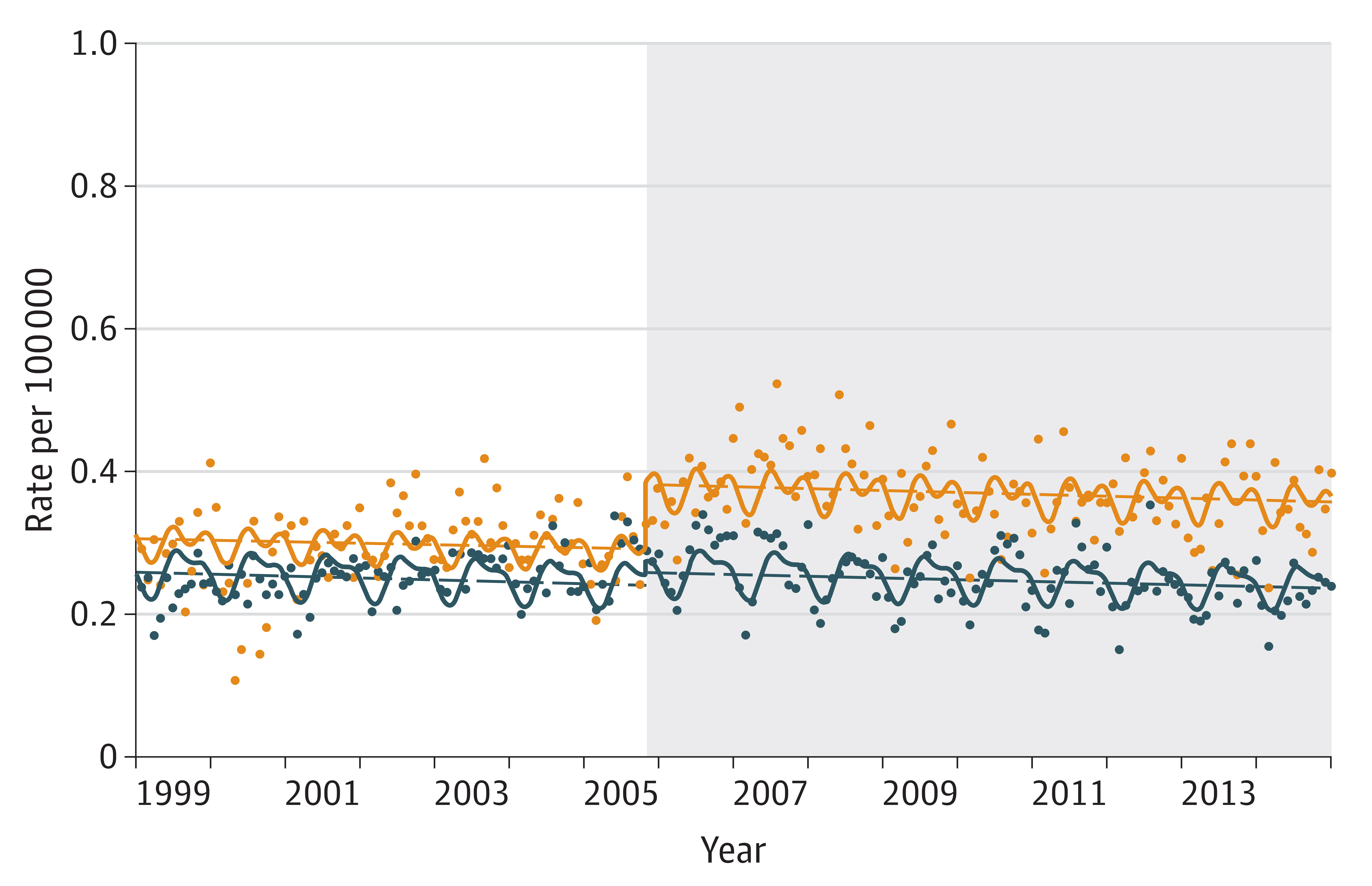
Data Analysis

- Evaluate whether post-intervention trends in homicide and homicide by firearm in Florida differed significantly from pre-intervention trends.
- Segmented quasi-Poisson regression analysis to analyze trends in both periods and estimate an effect size, taking underlying trends into account.
- Because of time sequencing of data points used in time series analysis, residual autocorrelation can lead to the violation of regression assumptions.
  - Generate robust standard errors (using a sandwich estimator) to produce more conservative estimates of uncertainty.

Homicide Rates in Florida and Comparison States



Homicide by Firearm Rates in Florida and Comparison States



Discussion

- Since Florida's stand your ground law took effect in October 2005, rates of homicide (+24.4% through 2014) and homicide by firearm (+31.6%) in the state have significantly increased.
- These increases appear to have occurred despite a general decline in homicide in the United States since the early 1990s.
- In contrast, rates of homicide and homicide by firearm did not increase in states without a stand your ground law (New York, New Jersey, Ohio, and Virginia), or for either suicide or suicide by firearm.
- Findings support the hypothesis that increases in the homicide and homicide by firearm rates in Florida are related to the stand your ground law.

Credits

- Graphics: Dave DiCello photography (cover)

- Shadish, William R., Thomas D. Cook, and Donald Thomas Campbell. Experimental and quasi-experimental designs for generalized causal inference. Boston: Houghton Mifflin, 2002.
  - Chapter 6: Interrupted time series
  - Chapter 7: Regression discontinuity design
- Morgan, S. L., & Winship, C. (2015). Counterfactuals and causal inference. Cambridge University Press.
  - Chapter 11: Repeated Observations and the Estimation of Causal Effects
- Humphreys, D. K., Gasparrini, A., & Wiebe, D. J. (2017). Evaluating the impact of Florida's "stand your ground" self-defense law on homicide and suicide by firearm: an interrupted time series study. JAMA Internal Medicine, 177(1), 44-50.
- Bernal, J. L., Cummins, S., & Gasparrini, A. (2017). Interrupted time series regression for the evaluation of public health interventions: a tutorial. International Journal of Epidemiology, 46(1), 348-355.
- Bhaskaran, K., Gasparrini, A., Hajat, S., Smeeth, L., & Armstrong, B. (2013). Time series regression studies in environmental epidemiology. International Journal of Epidemiology, 42(4), 1187-1195.
- Wagner, A. K., Soumerai, S. B., Zhang, F., & Ross-Degnan, D. (2002). Segmented regression analysis of interrupted time series studies in medication use research. Journal of Clinical Pharmacy and Therapeutics, 27(4), 299-309.
- Trockman, A., Zhou, S., Kästner, C., & Vasilescu, B. (2018). Adding sparkle to social coding: an empirical study of repository badges in the npm ecosystem. In Proceedings of the 40th International Conference on Software Engineering (pp. 511-522).