Change Point Analysis

Problem Statement

- A sample of observations (counts) from a Poisson process, where events occur randomly over a given time period

- A chart representation typically shows time on the horizontal axis and the number of events on the vertical axis.

- We want to identify the point(s) in time at which the rate of event occurrences changes, where the number of events is increasing or decreasing in frequency.

Poisson Distribution

pwäˈsän ˌdistrəˌbyo͞oSH(ə)n/

noun: Poisson distribution; plural noun: Poisson distributions

A discrete frequency distribution that gives the probability of a number of independent events occurring in a fixed time.

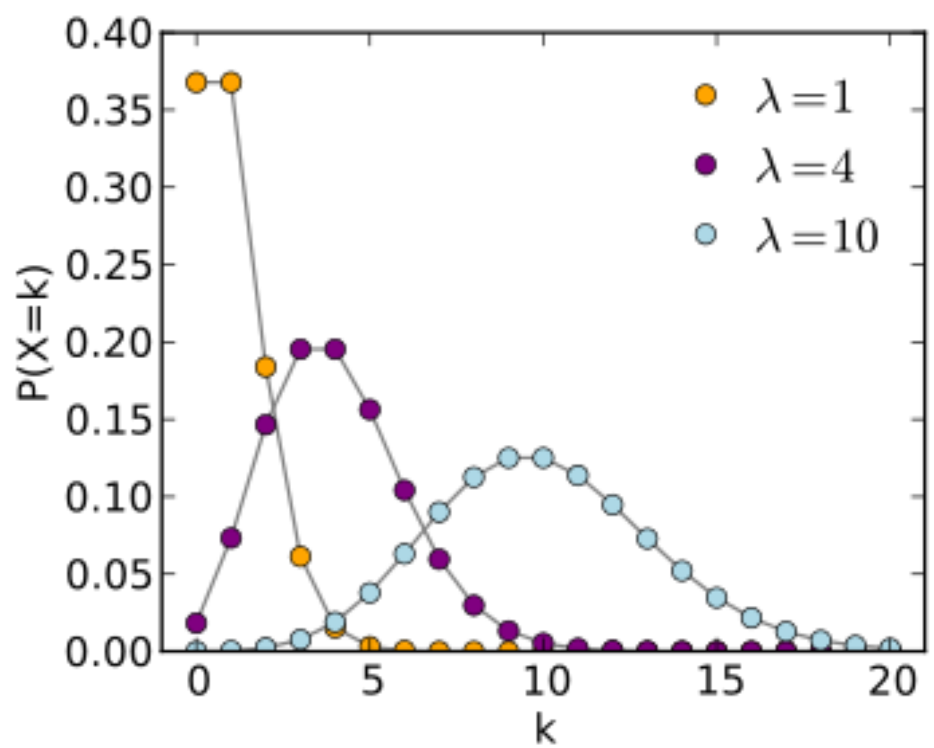
Origin:

Named after the French mathematical physicist Siméon-Denis Poisson (1781–1840).

https://www.google.com/webhp?sourceid=chrome-instant&ion=1&espv=2&ie=UTF-8#q=poisson%20distribution

Poisson Distribution

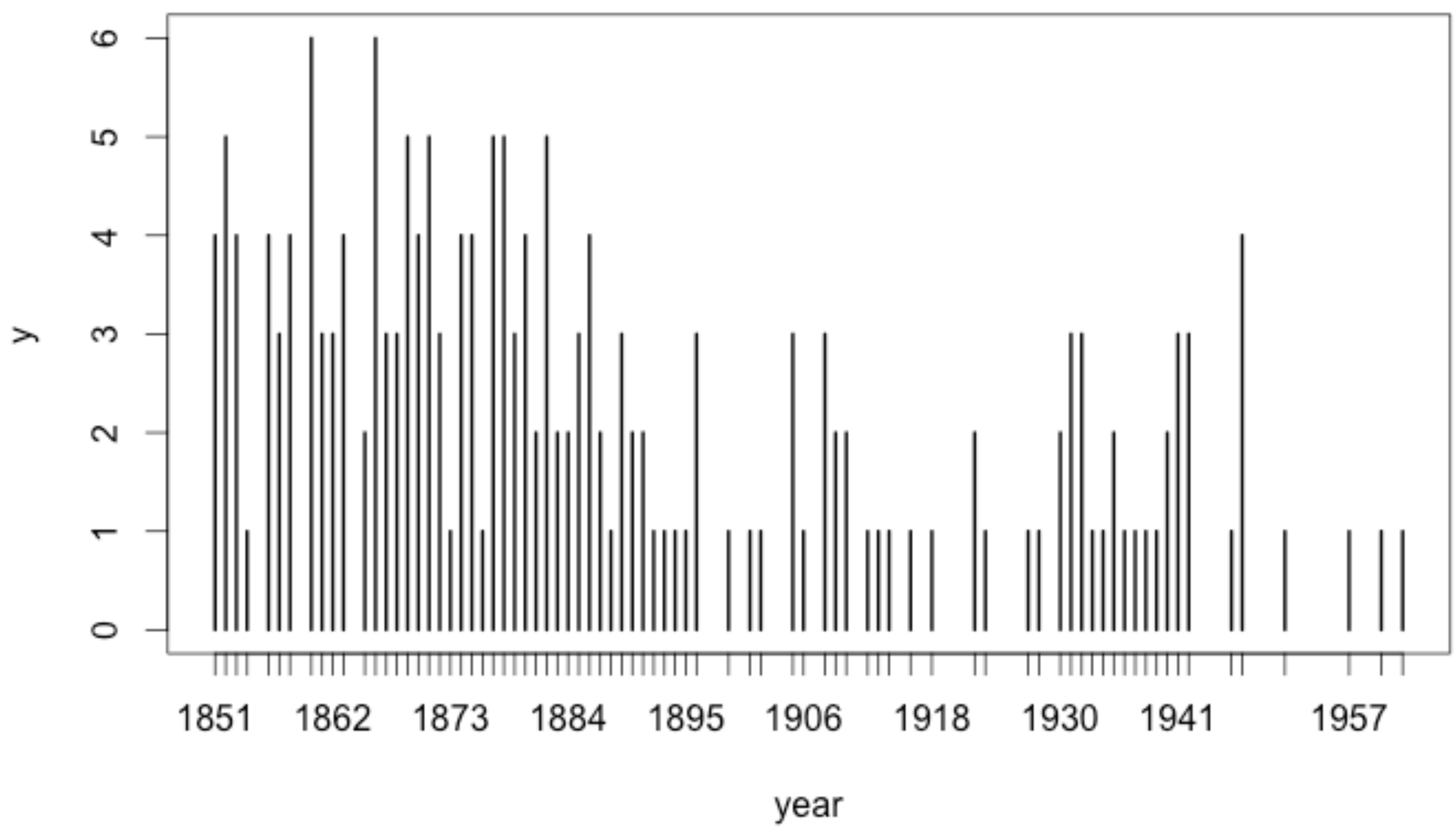
[https://en.wikipedia.org/wiki/Poisson_distribution]



The horizontal axis is the index k, the number of occurrences. λ is the expected value.

Classic Example

Coal Mining Disasters



Calculations

- Estimate the rate at which events occur before the shift (**μ**).
- Estimate the rate at which events occur after the shift (**λ**).
- Estimate the time when the actual shift happens (**k**).

Methodology

From the article:

“Apply a Markov Chain Monte Carlo (MCMC) sampling approach to estimate the population parameters at each possible k, from the beginning of your data set to the end of it. The values you get at each time step will be dependent only on the values you computed at the previous time step (that’s where the Markov Chain part of this problem comes in). There are lots of different ways to hop around the parameter space, and each hopping strategy has a fancy name (e.g. Metropolis-Hastings, Gibbs Sampler, “reversible jump”).”

Markov Chain Monte Carlo (MCMC)

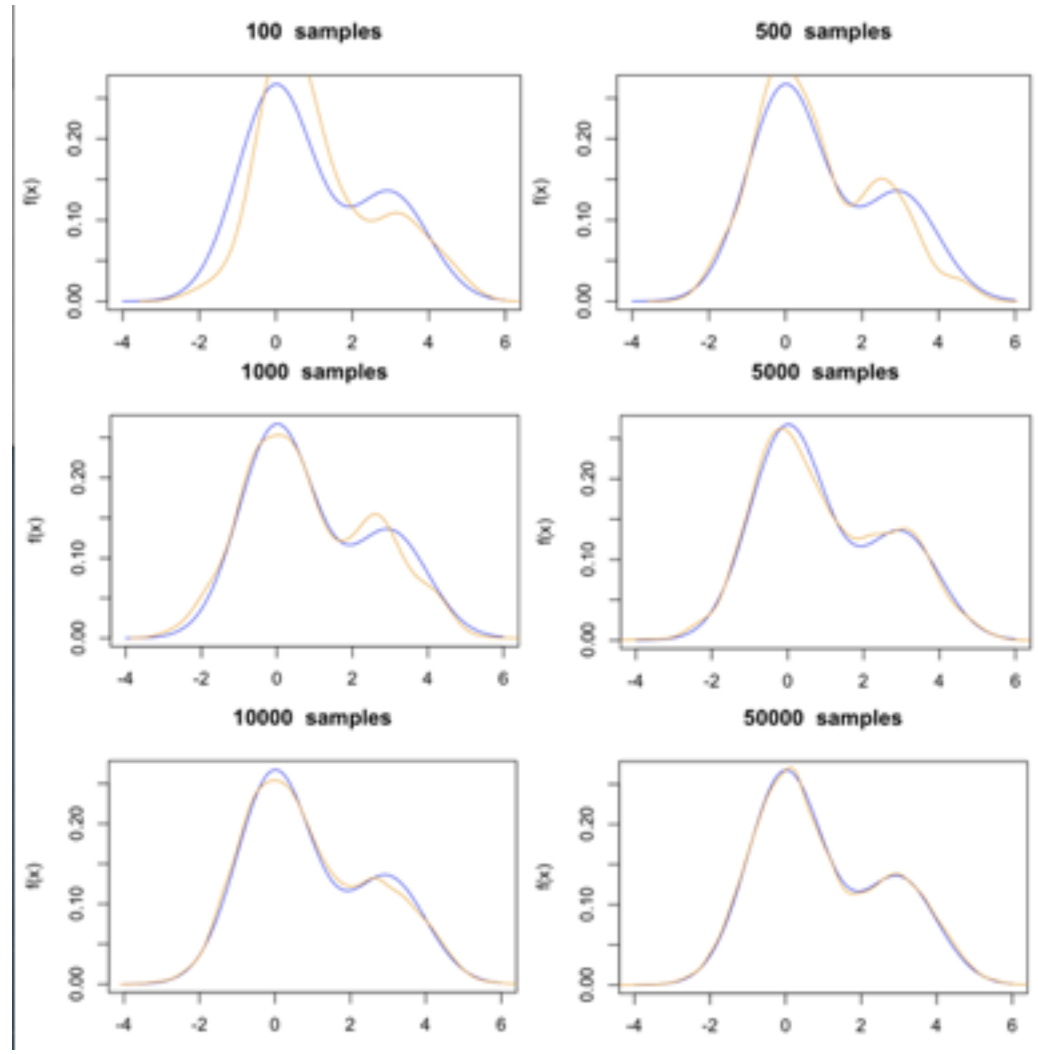
From Wikipedia:

“In statistics, Markov chain Monte Carlo (MCMC) methods are a class of algorithms for sampling from a probability distribution based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a number of steps is then used as a sample of the desired distribution. The quality of the sample improves as a function of the number of steps.”

http://stats.stackexchange.com/questions/165/how-would-you-explain-markov-chain-monte-carlo-mcmc-to-a-layperson

MCMC, continued

“Convergence of the Metropolis-Hastings algorithm. MCMC attempts to approximate the blue distribution with the orange distribution”



Back to the Example…

Rizzo, author of *Statistical Computing with R*
https://www.crcpress.com/Statistical-Computing-with-R/Rizzo/9781584885450

From the article: “Here are the models for prior (hypothesized) distributions that Rizzo uses, based on the Gibbs Sampler approach:

- **mu** comes from a Gamma distribution with shape parameter of (0.5 + the sum of all your frequencies UP TO the point in time, k, you’re currently at) and a rate of (k + b1)

- **lambda** comes from a Gamma distribution with shape parameter of (0.5 + the sum of all your frequencies after the point in time, k, you’re currently at) and a rate of (n – k + b1) where n is the number of the year you’re currently processing

- **b1** comes from a Gamma distribution with a shape parameter of 0.5 and a rate of (mu + 1)

- **b2** comes from a Gamma distribution with a shape parameter of 0.5 and a rate of (lambda + 1)

- **L** is a likelihood function based on k, mu, lambda, and the sum of all the frequencies up until that point in time, k”

Gibbs Sampler Code

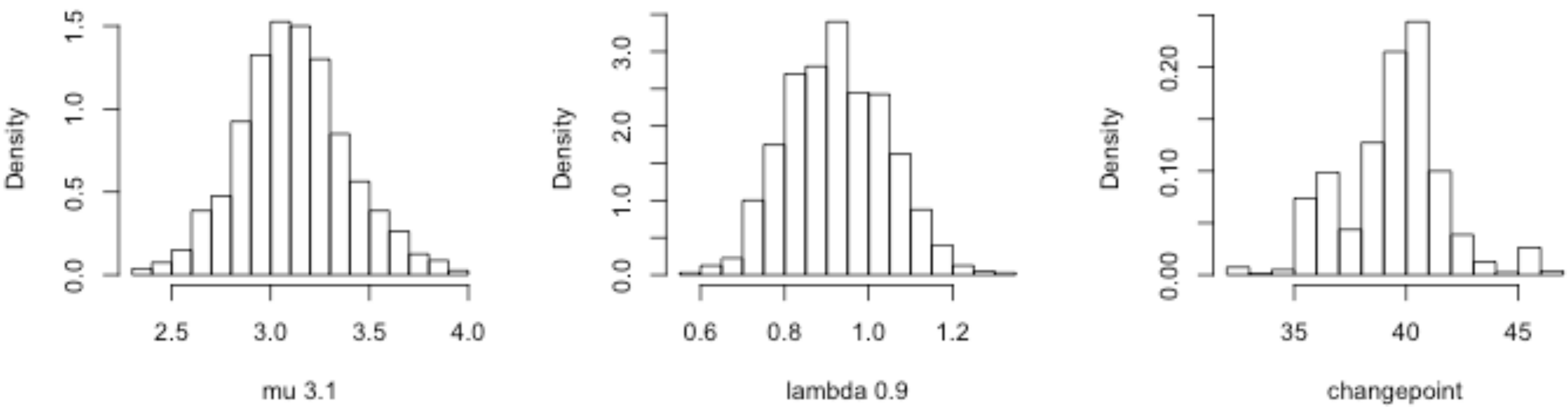
"At each iteration, you pick a value of k to represent a point in time where a change might have occurred. You slice your data into two chunks: the chunk that happened BEFORE this point in time, and the chunk that happened AFTER this point in time. Using your data, you apply a Poisson Process with a (Hypothesized) Gamma Distributed Rate as your model. This is a pretty common model for this particular type of problem. It's like randomly cutting a deck of cards and taking the average of the values in each of the two cuts… then doing the same thing again… a thousand times."

Simulation Results

“Knowing the distributions of mu, lambda, and k from hopping around our data will help us estimate values for the true population parameters. At the end of the simulation, we have an array of 1000 values of k, an array of 1000 values of mu, and an array of 1000 values of lambda — we use these to estimate the real values of the population parameters. Typically, algorithms that do this automatically throw out a whole bunch of them in the beginning (the “burn-in” period) — Rizzo tosses out 200 observations — even though some statisticians (e.g. Geyer) say that the burn-in period is unnecessary.”

Simulation Results, continued

Histograms

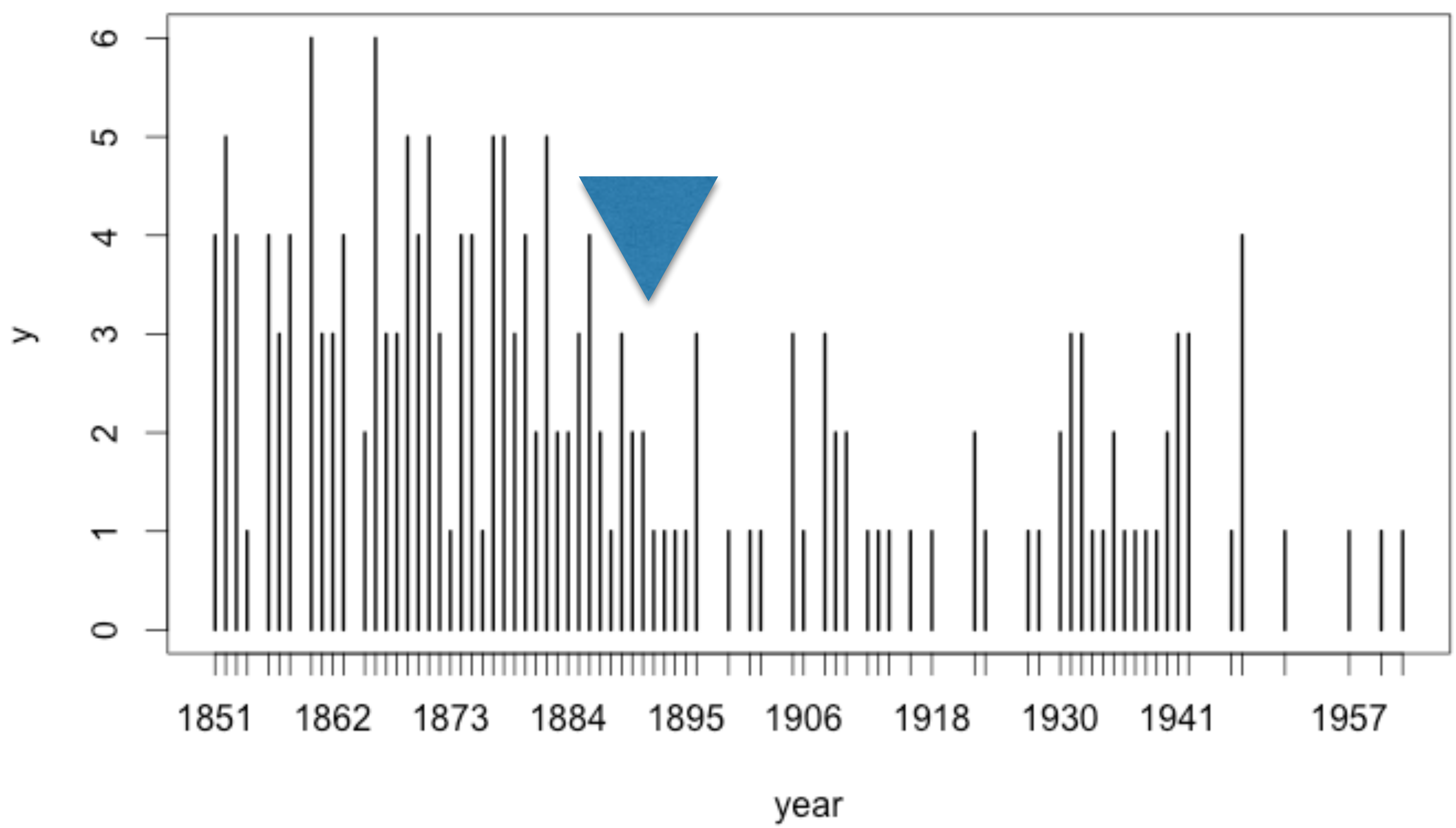


Demo

- Source Code from Rizzo’s Book
- Run in RStudio
- “The change point happened between the 39th and 40th observations, the arrival rate before the change point was 3.14 arrivals per unit time, and the rate after the change point was 0.93 arrivals per unit time. (Cool!)”

Results

- Add 40 to 1851 => Change Point @ 1891



Random Thoughts

- How soon can we detect change points? It can be quite easy to locate one in retrospect, but what about real-time change detection?

- Identifying changes in trend is difficult, so why is change point detection better than traditional time series analysis?

- My recurring thought has been that if someone showed you this chart, then you could just point to the change point and save some compute cycles. Duh!