Monitoring Process Change with Bayesian Methods

Mick Cooney

2 September 2015

Structure of Talk

- Discussion of Problem
- Bayesian Analysis and the Beta Distribution
- Adding Layers of Noise
- Distribution Distances and f-divergences

Monitoring Process Change

- NOT Change-point Analysis
- Time of change known - want to measure change effect
- Have measured metrics
- Need to determine change vs noise
- Generic technique for the problem

Sales-call Conversions

- Assume a binary outcome
- Conversion rate of sales calls to actual sales
- Amount irrelevant
- Data summarised monthly
- Change due to internal improvements leading to faster turnaround

Sales-call Conversions

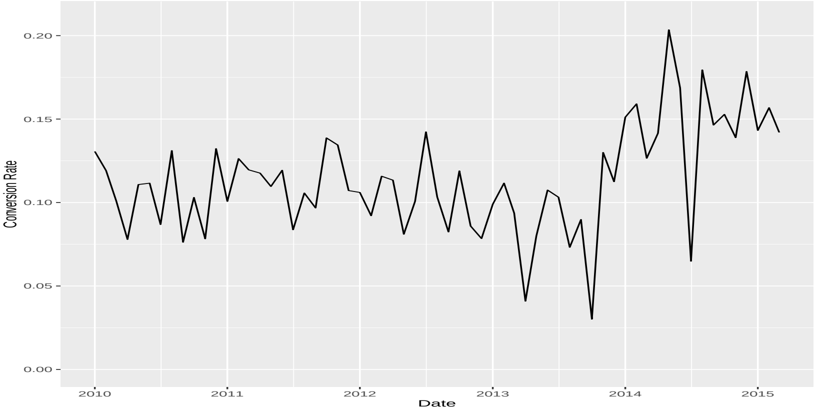
- Assume a binary outcome (0 or 1)
- Conversion rate of sales calls to actual sales
- Amount irrelevant
- Data summarised monthly
- Change due to internal improvements leading to faster turnaround

Generating Data

Want to generate time-series for θ, use normal distribution:

```
generate_process_rates <- function(mu0 = 0.10, sd0 = 0.03, mu1 = 0.15, sd1 = 0.03,
                                   start_date  = as.Date("2010-01-01"),
                                   end_date    = as.Date("2015-03-01"),
                                   change_date = as.Date("2014-01-01")) {

    month_vector <- as.yearmon(seq(start_date, end_date, by = "month"));
    switch_month <- as.yearmon(change_date);

    switch_idx <- match(switch_month, month_vector);

    pre_rate  <- rnorm(switch_idx - 1, mu0, sd0);
    post_rate <- rnorm(length(month_vector) - switch_idx + 1, mu1, sd1);

    rate_dt <- data.table(rate_date = as.Date(month_vector), underlying_rate = c(pre_rate, post_rate));

    return(rate_dt)
}
```

```r
plot_rate_dt <- generate_process_rates(mu0 = 0.10, sd0 = 0.02, mu1 = 0.15, sd1 = 0.03);

qplot(rate_date, underlying_rate, data = plot_rate_dt, geom = 'line', ylim = c(0, 0.21),
      xlab = 'Date', ylab = 'Conversion Rate');
```



```
generate_counts <- function(rate_dt, month_count) {
    rate_dt <- data.table(rate_dt, month_count = month_count);

    rate_dt[, conversion_count := mapply(rbinom, n = 1, month_count, underlying_rate)];
    rate_dt[, conversion_rate  := conversion_count / month_count];

    return(rate_dt);
}

generate_yearly_data <- function(rate_dt) {
    year_dt <- rate_dt[, list(a = sum(conversion_count), b = sum(month_count - conversion_count)),
                         by = list(data_year = format(rate_date, '%Y'))];
    year_dt[, c("cum_a", "cum_b") := list(cumsum(a) + 1, cumsum(b) + 1)];

    distrib_dt <- year_dt[, generate_beta_plot_data(cum_a, cum_b), by = data_year];

    return(distrib_dt);
}

generate_beta_plot_data <- function(a, b) {
    theta     <- seq(0, 1, by = 0.0001);
    prob_dens <- dbeta(theta, a, b);

    return(data.table(theta = theta, prob_dens = prob_dens));
}
```

Bayes Rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Continuous Form:

$$p(\theta|D) = \int p(D|\theta)\, p(\theta)\, d\theta$$

where

$p(\theta)$	Prior distribution for θ
$p(D|\theta)$	Probability of seeing data D given value θ
$p(\theta|D)$	Posterior distribution for θ

Binomial Likelihood

For single binomial trial with probability θ and outcome y:

$$p(y|\theta) = \theta^{y}(1-\theta)^{1-y}$$

For n trials with k successes:

$$p(k|\theta) = \binom{n}{k}\theta^{k}(1-\theta)^{n-k}$$

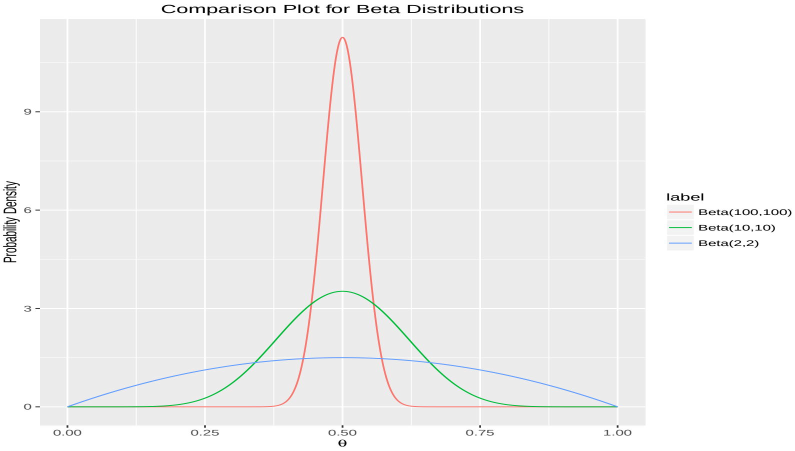
Beta Distribution

$$X \sim Beta(\alpha, \beta)$$

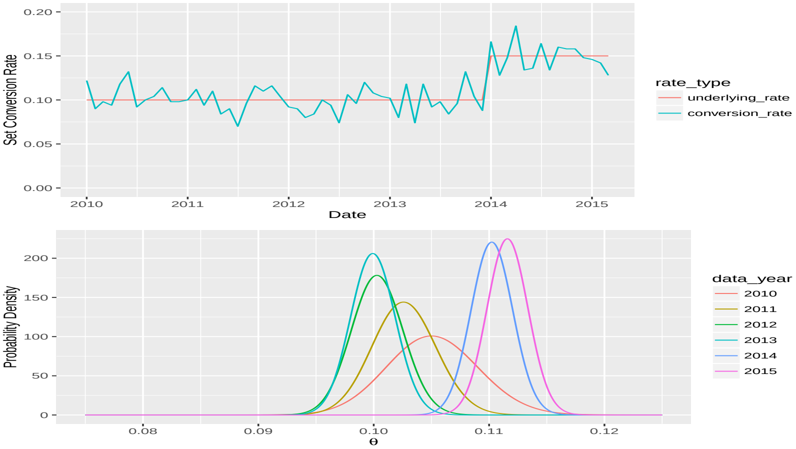
- Parameterised by two parameters α, β
- Correspond to assumed prior success/fail counts
- Simple to do update with new data

$$p(\theta|D) = Beta(\alpha + k, \beta + n - k)$$

```
## Error:  Unknown parameters:  NA
## Error:  Unknown parameters:  NA
## Error:  Unknown parameters:  NA
## Error:  Unknown parameters:  NA
## Error in getClass(Class):  "Scales" is not a defined class
```


Comparison Plot for Beta Distributions
Probability Density
9
6
3
0
0.00
0.25
0.50
0.75
1.00
θ
label
Beta(100,100)
Beta(10,10)
Beta(2,2)

First Pass at the Problem

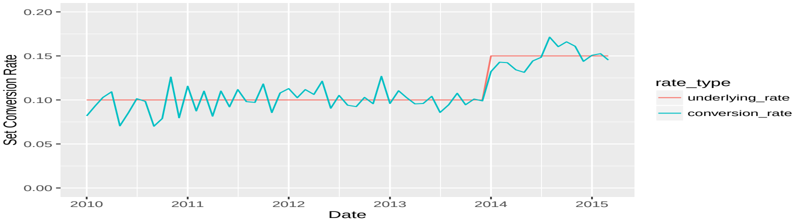


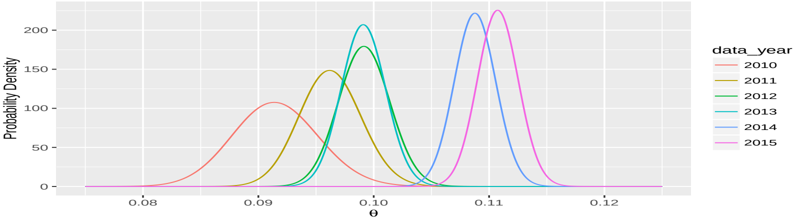
Randomising Counts per Month

- More random noise
- Call counts per month fixed (500 per month)
- Model instead as Poisson process

$$C \sim Pois(500)$$

Randomising Counts per Month



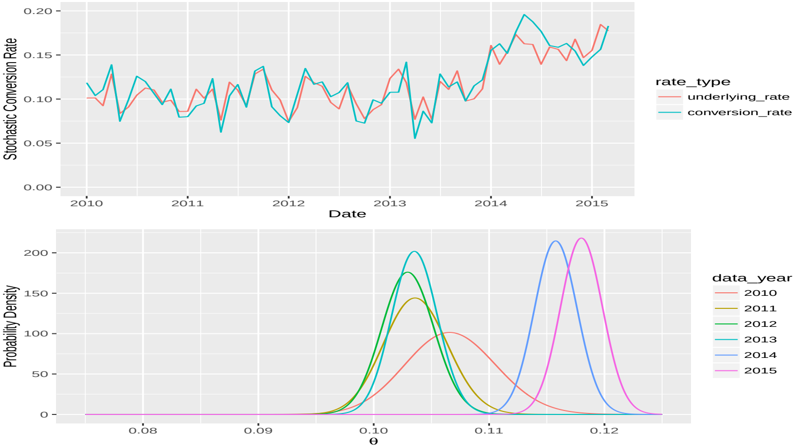


Stochastic Conversion Rate

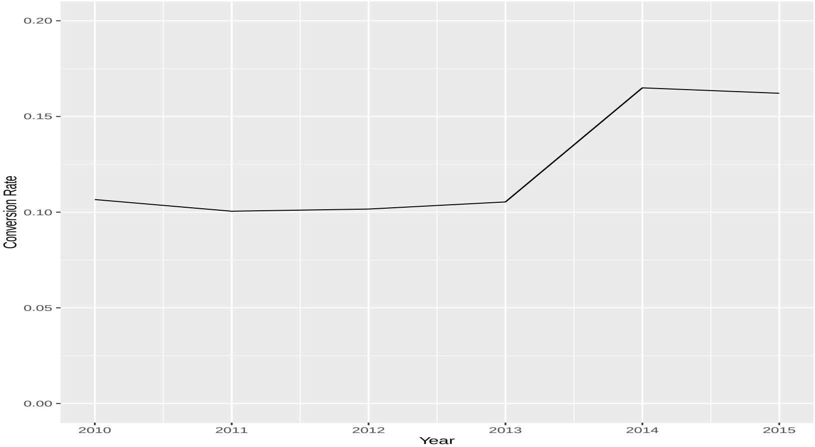
- Add noise to the conversion rate
- Model underlying rate with normal distribution
- Noise on conversion rate and monthly count

$$\theta \sim \mathcal{N}(\mu, \sigma)$$

Stochastic Conversion Rate



Taking a Step Back



Inconsistent?

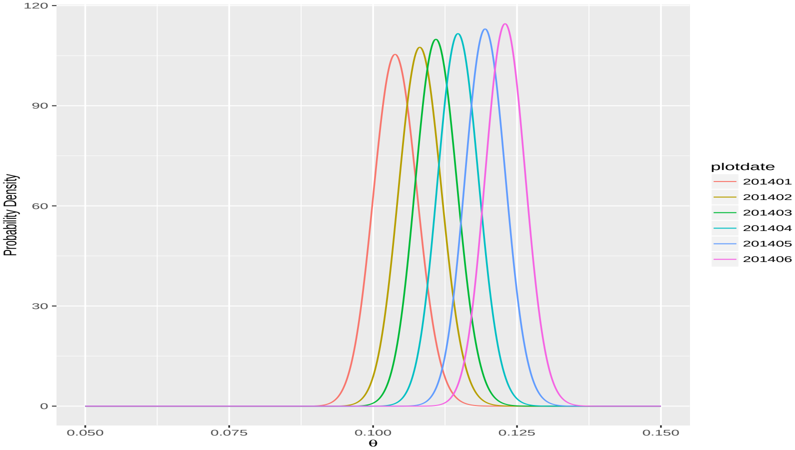
Building a New Prior

- Balancing act
- Try 6 months, use θ from data
- Re-parameterize $\mathit{Beta}(\alpha, \beta)$

$$\mathit{Beta}(\alpha, \beta) = \mathit{Beta}(\mu K, (1 - \mu)K)$$

$$\begin{aligned} \mu &= 0.0997 \\ K &= 6,000 \end{aligned}$$

Using the New Prior



Using a Higher μ

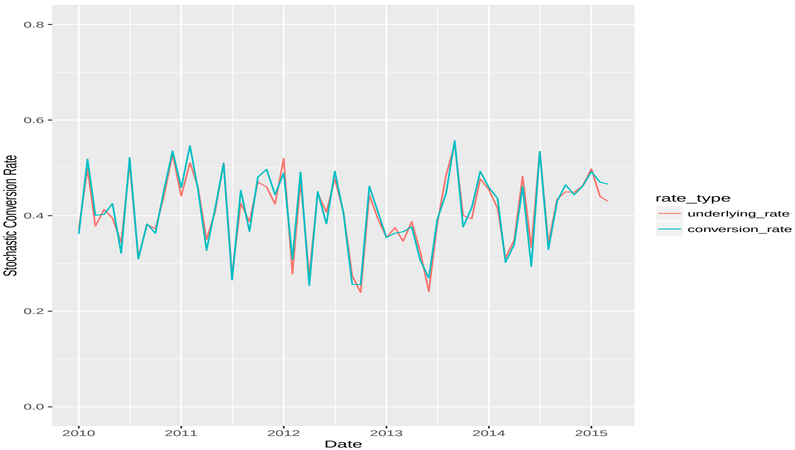
Before:

$$\mu_1 \sim \mathcal{N}(0.1, 0.02) \rightarrow \mu_2 \sim \mathcal{N}(0.15, 0.02)$$

Now:

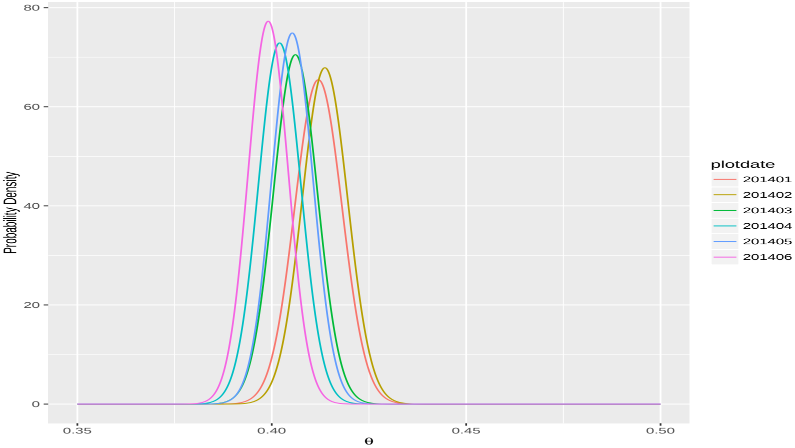
$$\mu_1 \sim \mathcal{N}(0.4, 0.08) \rightarrow \mu_2 \sim \mathcal{N}(0.45, 0.08)$$

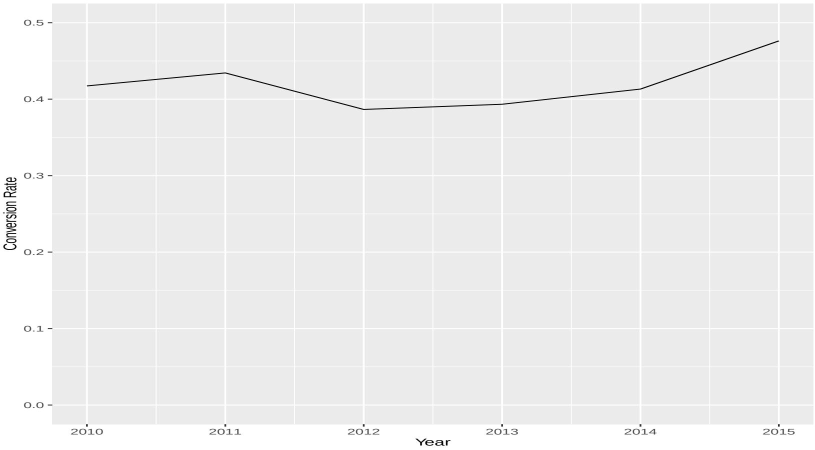
Analysis for $\mu = 0.40$



Very hard to spot a change!

Analysis for $\mu = 0.40$




Conversion Rate
0.0
0.1
0.2
0.3
0.4
0.5
2010
2011
2012
2013
2014
2015
Year

Differences between Distributions

A *metric* or *distance*

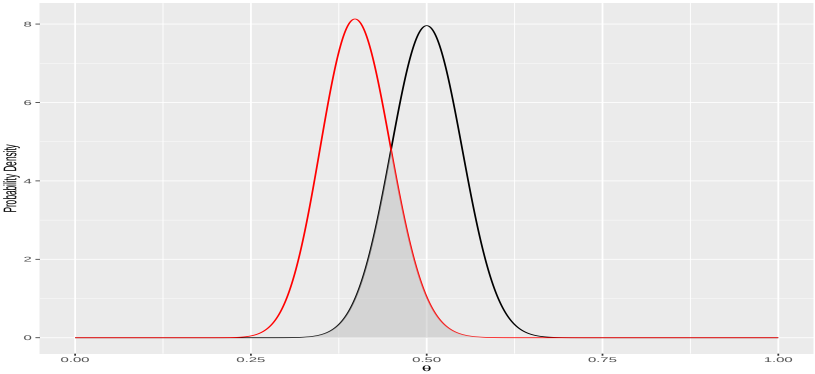
$$d : X \times X \to \mathbf{R}^{+},$$

1. $d(x, y) \geq 0 \; \forall x, y \in X$ (non-negativity)
2. $d(x, y) = 0 \textit{ iff } x = y \; \forall x, y \in X$ (identity of indiscernables)
3. $d(x, y) = d(y, x) \; \forall x, y \in X$ (symmetry)
4. $d(x, z) \leq d(x, y) + d(y, z) \; \forall x, y, z \in X$ (triangle inequality)

(1) and (2) together produce *positive definiteness*

Common-Area Metric

$$D(P, Q) = \int_0^1 \min(P(x), Q(x))\, dx$$



Kullback-Leibler Divergence

$$D_{KL}(P||Q) = \int_0^1 p(x) \ln \frac{p(x)}{q(x)} \, dx$$

- Not symmetric
- Does not obey Triangle Inequality
- Additional bits required to 'correct' signal P when using Q

Hellinger Distance

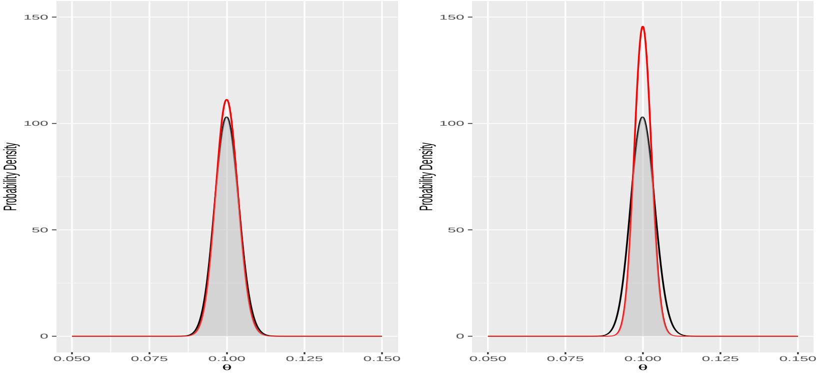
$$H^2(P, Q) = 1 - \int \sqrt{p(x)q(x)}\, dx$$

$$0 \leq H(P, Q) \leq 1$$

$$H^2(P, Q) \leq \delta(P, Q) \leq \sqrt{2} H(P, Q)$$

Distance Values for Beta Distribution

$$\mu = 0.10 \quad K_1 = 6,000 \quad K_2 = 7,000 \quad K_3 = 12,000$$



Distance Values for Beta Distribution

```
print(calculate_metrics(x_seq, Beta1, Beta1));
```

```
##  commonarea   hellinger          kl
## 4.44089e-16 4.44089e-16 0.00000e+00
```

```
print(calculate_metrics(x_seq, Beta1, Beta2));
```

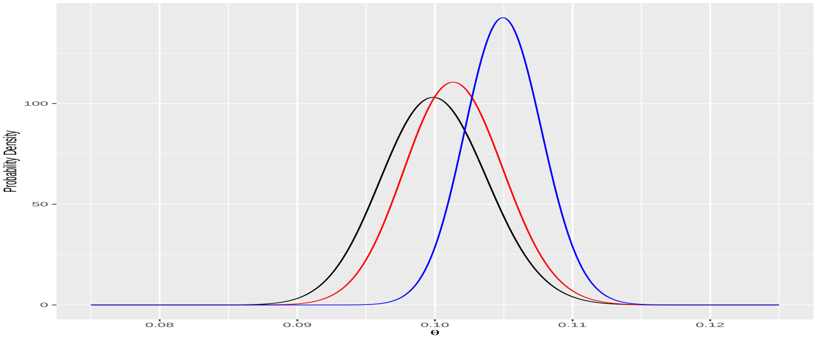
```
## commonarea  hellinger         kl
## 0.03729146 0.00148335 0.00626134
```

```
print(calculate_metrics(x_seq, Beta1, Beta3));
```

```
## commonarea  hellinger         kl
##  0.1660940  0.0290278  0.1534966
```

Distance Values for Beta Distribution

$$\mu_1 = 0.10 \;\; \mu_2 = 0.11 \;\; K_1 = 6,000 \;\; K_2 = 7,000 \;\; K_3 = 12,000$$

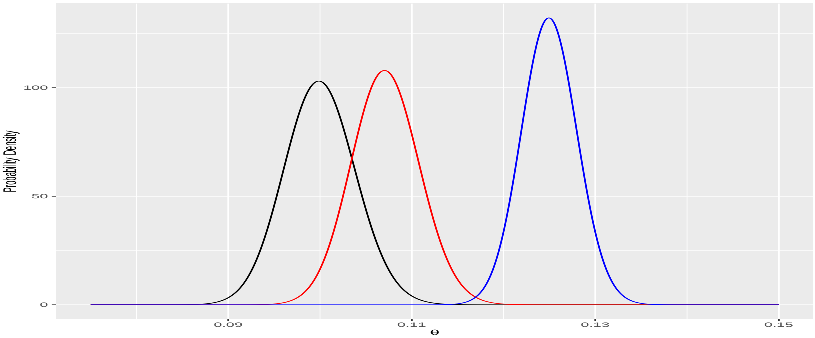


Distance Values for Beta Distribution

```
print(calculate_metrics(x_seq, Beta1, Beta1));


##  commonarea   hellinger          kl
## 4.44089e-16 4.44089e-16 0.00000e+00


print(calculate_metrics(x_seq, Beta1, Beta2));


## commonarea  hellinger         kl
##  0.1552227  0.0197388  0.0864029


print(calculate_metrics(x_seq, Beta1, Beta3));


## commonarea  hellinger         kl
##   0.559182   0.262379   1.818925
```

Distance Values for Beta Distribution

$$\mu_1 = 0.10 \;\; \mu_2 = 0.15 \;\; K_1 = 6,000 \;\; K_2 = 7,000 \;\; K_3 = 12,000$$

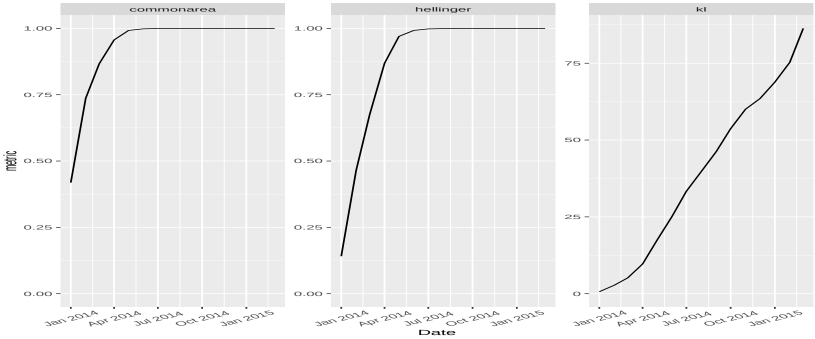


Distance Values for Beta Distribution

```
print(calculate_metrics(x_seq, Beta1, Beta1));


##  commonarea   hellinger          kl
## 4.44089e-16 4.44089e-16 0.00000e+00


print(calculate_metrics(x_seq, Beta1, Beta2));


## commonarea  hellinger         kl
##   0.655281   0.360161   1.956383


print(calculate_metrics(x_seq, Beta1, Beta3));


## commonarea  hellinger         kl
##   0.999673   0.998076  39.199406
```

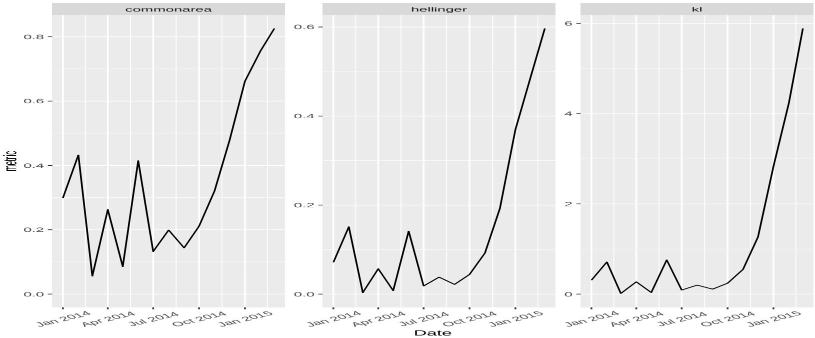
Create Comparison Charts

$$\mu_1 = 0.10 \;\; \mu_2 = 0.15$$



Create Comparison Charts

$$\mu_1 = 0.40 \quad \mu_2 = 0.45$$



Summary

- Binomial process with known change point
- Use Beta distribution for simplicity
- Aggregate data in meaningful way (decay data as necessary)
- Track changes using ‘distance metric’
- Decide on thresholding (if necessary)

Future Work

- Try with different distributions (Normal, Poisson, Multinomial)
- More comprehensive investigation of behaviour of distributions
- Randomised data to see patterns in metrics
- Look at statistical distance
- Time-series methods

Summary

michael.cooney@applied.ai
mickcooney@gmail.com

Slides and code available on github:
`https://github.com/kaybenleroll/dublin_r_workshops`