11 - Anomaly Detection

Outliers, Changes, Outbreaks, Goodness-of-fit

SYS 6018 | Fall 2020

11-anomaly.pdf

Contents

1 Anomaly Detection Intro **2**
- 1.1 Required R Packages . . . 2
- 1.2 Anomaly Detection . . . 2
- 1.3 Example #1: Benford's Distribution . . . 3

2 Goodness of Fit Testing **6**
- 2.1 Hypothesis Testing . . . 6
- 2.2 GOF Test Statistics . . . 7
- 2.3 Testing . . . 10

3 Outlier Detection **12**
- 3.1 Distance based approach . . . 12
- 3.2 Likelihood Based Approach . . . 13
- 3.3 Mixture Model Approach . . . 14

4 Two-Sample Testing (A/B Testing) **16**
- 4.1 Example (A/B Testing): Clinical Trials . . . 16
- 4.2 Friedman's Supervised Modeling Approach to Testing . . . 18

5 Hotspot Detection **19**
- 5.1 Example: Land Mine and UXO Detection . . . 19
- 5.2 Hotspot mixture model . . . 19
- 5.3 Spatial Scan Statistic . . . 21
- 5.4 Network Scan Statistic . . . 28

6 Outbreak Detection **29**

1 Anomaly Detection Intro

1.1 Required R Packages

We will be using the R packages of:

- `tidyverse` for data manipulation and visualization
- `readxl` for loading excel data into R (part of tidyverse, but not loaded)
- `mclust` for model-based clustering

```
library(mclust)
library(tidyverse)
library(readxl)
```

1.2 Anomaly Detection

Anomaly Detection: The identification of unusual observations. Statistically, this means finding observations that come from a different distribution that the *normal* or *usual* observations.

1. **Goodness of Fit (GOF)**
   - Tests if data conform to a given distribution (or distributional family)
   - Use case: Failure of the first digits in a financial statement to conform to Benford's distribution may indicate fraud.
2. **Two-Sample Tests (A/B Testing)**
   - Tests if two datasets come from the same distribution
   - Often simplified to test if one group has a larger mean than the other
   - Use case: Determine if a new surgical technique leads to faster recovery times; determine if a new website popup increases purchases.
3. **Outlier Detection**
   - Tests if a single observation or small set of observations come from the same distribution as the rest of the data
   - Use case: Detect data entry errors.
4. **Hotspot Detection**
   - Identification of regions that have *unusually* high density
   - Use case: association rule mining; cross-sell to customers based on what they have in their shopping cart.
   - Use case: locate ambulances in a city so they are close to the locations that have an unusually high call for service rate.
5. **Outbreak Detection**
   - A sequential method that repeatedly tests for a change in an event distribution
   - Focus first on determining *if* a change occurred, but also estimating *when* it occurred
   - For outbreak detection, the changes of interest are those that conform to an expected *outbreak pattern*
   - Use case: quickly detect the presence of West Nile Virus from the *chief complaints* field of health records and initiate a rapid mosquito control spraying
6. **Changepoint Detection**
   - Detecting when changes in a sequential process occur
   - Use case: estimating if a manufacturing system has gone "out of control" so the bad parts can be reworked and the maintenance can be deployed.

1.3 Example #1: Benford's Distribution

```
#> Error in select(., country = 2, area = 4): unused arguments (country = 2, area = 4)
```

country	area1	area2
Afghanistan	682,213	652,864
Albania	45,480	27,398
Algeria	6,566,516	2,381,741
Andorra	529	468
Angola	2,224,875	1,246,700
Antigua and Barbuda	161	443
Argentina	2,162,374	2,736,690
Armenia	45,381	28,342
Australia	1,023,244	7,633,565
Austria	65,819	82,445
Azerbaijan	78,797	86,100
Bahrain	769	778
Bangladesh	843,549	130,168
Barbados	480	431
Belarus	468,359	202,900
Belgium	58,572	30,278
Belize	96,465	22,806
Benin	688,202	114,305

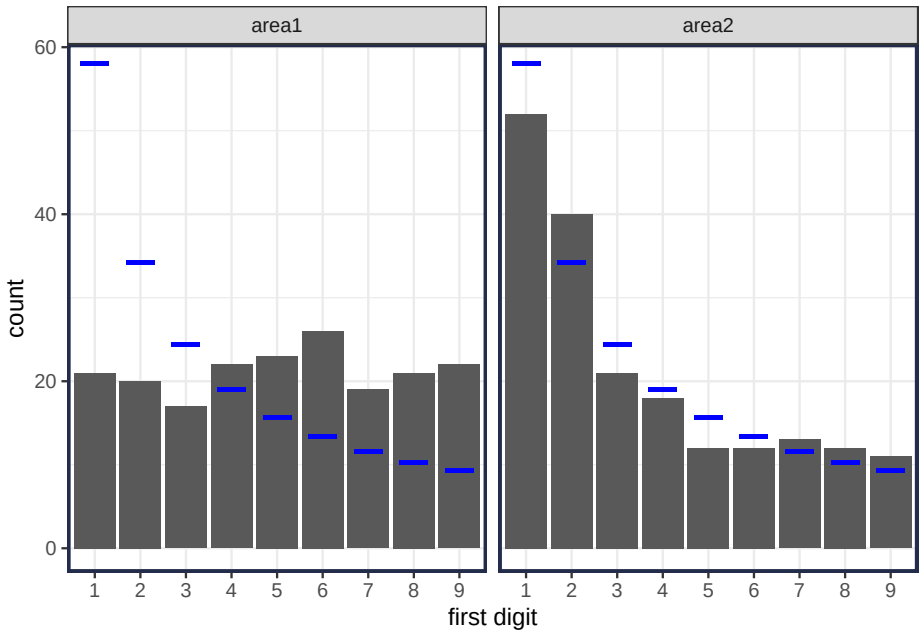
Based on Fewster (2009) A Simple Explanation of Benford's Law, *The American Statistician*, 63, 1, pp 26–32. One of the area measurements is fake.

- Someone that fakes numbers, say on a financial statement, may be tempted to use a random number generator
  - But they better watch out for Benford's Law
- Note on terminology:
  - Law = probability distribution
- Benford's PMF:

$$\Pr(\text{first digit} = x) = \log_{10}\left(1 + \frac{1}{x}\right) \quad \text{for } x = 1, 2, \ldots, 9$$

- R code for a Benford's pmf

```
#-- pmf for Benford's distribution
dbenford <- function(x) log10(1 + 1/x)

#-- first digit
dbenford(1:9)
#> [1] 0.30103 0.17609 0.12494 0.09691 0.07918 0.06695 0.05799 0.05115 0.04576
```

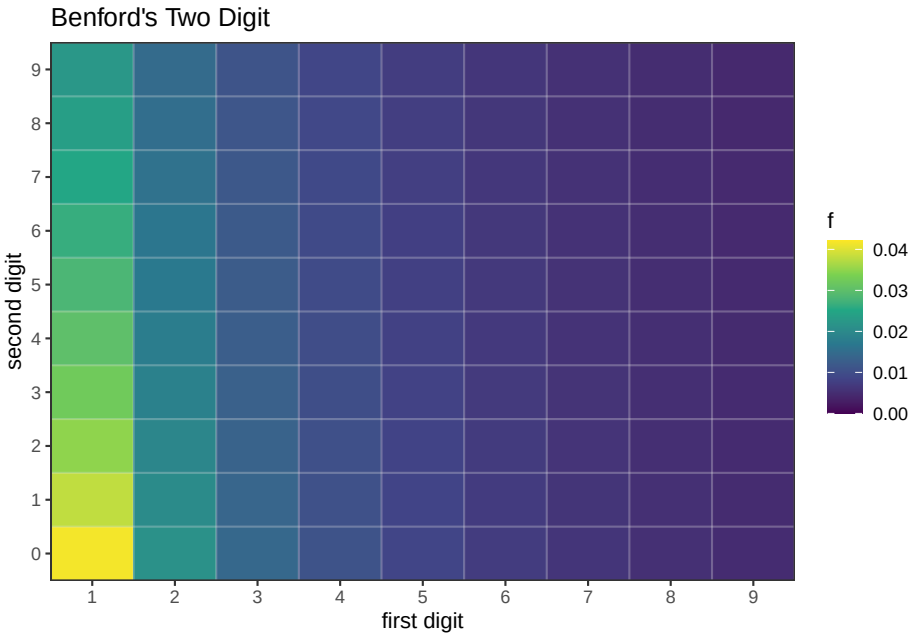


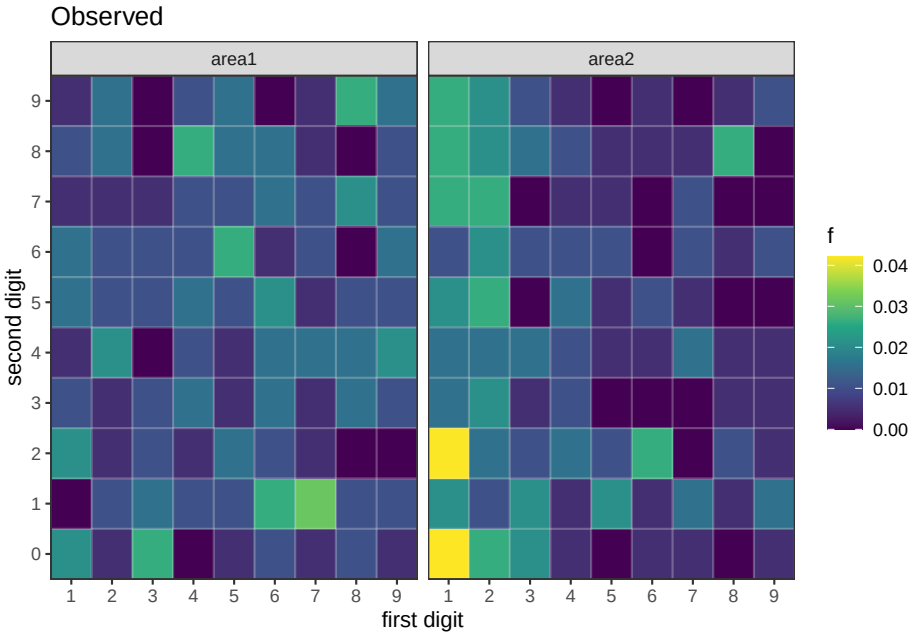
- Distribution of the first two digits

$$\Pr(\text{first two digits} = x) = \log_{10}\left(1 + \frac{1}{x}\right) \quad \text{for } x = 10, 11, \ldots, 99$$

```
#-- first two digits
X = expand_grid(first=1:9, second=0:9) %>%
  mutate(two = str_c(first, second) %>% as.integer) %>%
  mutate(f = dbenford(two))
```

first	second									
	0	1	2	3	4	5	6	7	8	9
1	0.041	0.038	0.035	0.032	0.030	0.028	0.026	0.025	0.023	0.022
2	0.021	0.020	0.019	0.018	0.018	0.017	0.016	0.016	0.015	0.015
3	0.014	0.014	0.013	0.013	0.013	0.012	0.012	0.012	0.011	0.011
4	0.011	0.010	0.010	0.010	0.010	0.010	0.009	0.009	0.009	0.009
5	0.009	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.007	0.007
6	0.007	0.007	0.007	0.007	0.007	0.007	0.007	0.006	0.006	0.006
7	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.005
8	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.005
9	0.005	0.005	0.005	0.005	0.005	0.005	0.005	0.004	0.004	0.004
Sum	0.120	0.114	0.109	0.104	0.100	0.097	0.093	0.090	0.088	0.085


Benford's Two Digit
second digit
first digit
f
0.04
0.03
0.02
0.01
0.00


Observed
area1
area2
second digit
first digit
f
0.04
0.03
0.02
0.01
0.00

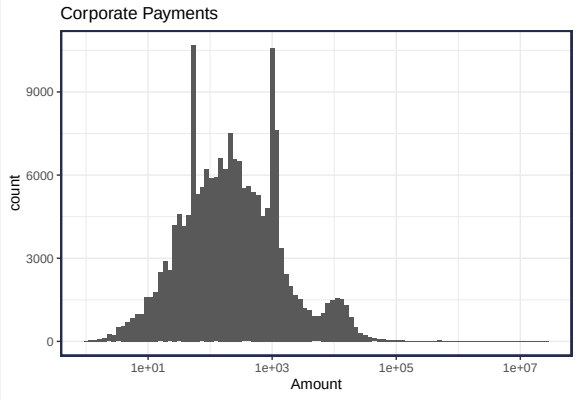
2 Goodness of Fit Testing

Tests if data conform to a given distribution (or distributional family)

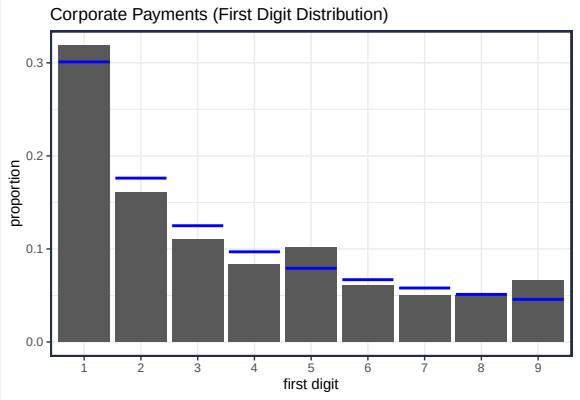
Your Turn #1 : Benford's

Mark Nigrini provides a dataset of the 2010 payments from a division of a West Coast utility company https://www.nigrini.com/BenfordsLaw/CorporatePaymentsData.xlsx.

```
## Download xlsx file to local machine, then load into R.
## Notice that the data is in the "Data" sheet
library(readxl)
corp_payments = read_excel("../topics/anomaly/data/CorporatePaymentsData.xlsx",
                           sheet="Data") %>%
  filter(Amount > 1)  # only consider positive payments more than $1
```




```
#-- Extract the first (non-zero) digit and add as column to data frame
corp_payments = corp_payments %>%
  mutate(first = stringr::str_extract(Amount, '[1-9]'))
```



How can we test if the first digit of the `Amount` column conform to Benford's distribution?

2.1 Hypothesis Testing

Let $D = (X_1, X_2, \ldots, X_n)$ be the observed random variables (i.e., the data).

1. Specify a *null* hypothesis, $\mathcal{H}_0$
2. Choose a *test statistic* $T = T(X_1, \ldots, X_n)$ that is a function of the observed data
   - T is a *random variable*; it has a distribution.

- Let $t = T(x_1, \ldots, x_n)$ be the *observed* value of the test statistic
- It is common to structure the test statistic so that *extreme* means large values of T
- Select a test statistic that has good *power* to reject the null when an alternative hypothesis is true.
  - (power also known as *sensitivity* and *true positive rate* in the contect of binary classification problems)

3. Calculate the p-value. The p-value is *the probability that chance alone would produce a test statistic as extreme/unusual as the observed test statistic if the null hypothesis is true*
   - E.g., p-value $= \Pr(T \geq t \mid \mathcal{H}_0)$, when large T indicates extreme/unusual
4. Make a decision about the null hypothesis, i.e., reject the null.
   - Notice that we haven't specified any alternative hypotheses at this point.
   - This description is following the usual frequentist set-up
   - See the Alternative Hypotheses section below for a discussion of how to approach hypothesis testing in the Bayesian framework

- Think of T or the p-value as the *evidence against* the null hypothesis
  - Its common to set a threshold (e.g., p-value $\leq .05$) and *reject* the null hypothesis when this threshold is crossed.
  - This is a form of *outlier detection*. Reject null if t_{obs} is an *outlier*; that is t_{obs} is from a different distribution that what is specified in $\mathcal{H}_0$.
- Note: To calculate a p-value, we need to know/estimate the *distribution* of $T|\mathcal{H}_0$!
  - Even if we don't know the exact distribution of T under the null, we can often approximate it using simulation (Monte Carlo)

2.1.1 Alternative Hypotheses

- The choice of test statistic should be driven by the expected deviations from the null
  - That is, we can come up with *better* test statistic if we know what sort of deviations from $\mathcal{H}_0$ are expected.
  - *better* meaning more *power* to correctly reject the null
- If you take a Bayesian approach, you would specify all possibly hypothesis $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_p\}$, along with all the prior distributions, and estimate $\Pr(\mathcal{H}_j \mid D)$ (or $\Pr(\mathcal{H}_j \mid T)$, where T is the test statistic).
  - But we are taking the *frequentist* approach for this lesson
- Consult any Statistics text (and the assigned readings) to see more details about Statistical Hypothesis Testing

2.2 GOF Test Statistics

- Going back to the original question about the corporate payments conforming to Benford's distribution, we can state the *null hypothesis* formally:

$$\mathcal{H}_0 : X \overset{\text{iid}}{\sim} Benf$$

- X is the *first* digit(s)
- $Benf$ stands for Benford's distribution for the first digit(s). This has pmf:

$$f(x) = \log_{10}\left(1 + \frac{1}{x}\right)$$

  - Note: there are no parameters to estimate!
- A generic *alternative hypothesis* is:

$$\mathcal{H}_1 : X \not\sim Benf$$

- We will present two popular test statistics, the χ^2 and the *likelihood ratio* test statistics

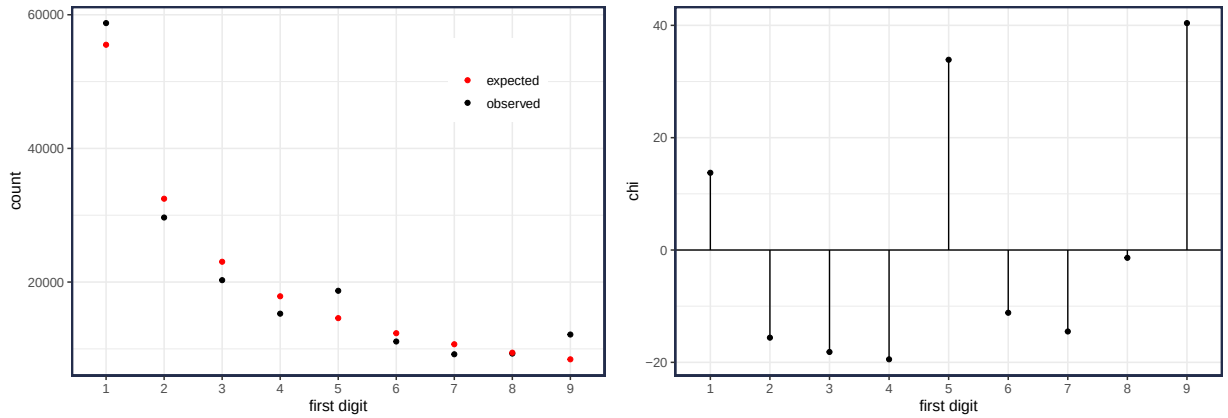
2.2.1 χ^2 Test Statistic

- The Pearson's χ^2 (chi-squared) test statistic is commonly used in goodness-of-fit testing.
- It requires the data to be *discrete* or *categorical*
  - Continuous data can be *binned*
- Test Statistic:

$$\chi^2 = \sum_{j=1}^{J} \frac{(y_j - e_j)^2}{e_j} = \sum_{j=1}^{J} \frac{(y_j - np_j)^2}{np_j}$$

  - J: number of categories or possible values
  - y_j: *observed* count in category j
  - e_j: *expected* count, under $\mathcal{H}_0$, in category j
    - $n = \sum_{j=1}^{J} y_j$ is the total number of observations
    - p_j: the proportion of events under the null (i.e., $p_j = \Pr(X = j \mid \mathcal{H}_0)$)
  - Asymptotically, χ^2 statistics converges to a chi-squared distribution with $J-1$ degrees of freedom
- R code for corporate payments data

```
#-- Get counts
y = table(corp_payments$first)     # count of first digits
n = length(corp_payments$first)    # number of observations

#-- chi-squared
e = n*dbenford(1:9)   # expected count vector
chi = (y-e)/sqrt(e)   # vector of deviations
(chisq = sum(chi^2)) # chi-squared test statistic
#> [1] 4258
```



- Note: there is a build-in R function `chisq.test()` which does these calculations

```r
chisq.test(y, p=dbenford(1:9))$statistic
#> X-squared
#>      4258
```

2.2.2 Likelihood Ratio Test Statistic

When an *alternative hypothesis* can be specified with a distribution, the log-likelihood ratio test statistic is commonly used in goodness-of-fit testing.

- The general binary hypothesis formulation is:
  - $\mathcal{H}_0 : X \sim f_0(X)$ (null hypothesis)
  - $\mathcal{H}_1 : X \sim f_1(X)$ (alternative hypothesis)
- The *likelihood ratio* is:

$$\begin{aligned} LR &= \frac{f_1(X_1, \ldots, X_n)}{f_0(X_1, \ldots, X_n)} \\ &= \prod_{i=1}^{n} \frac{f_1(X_i)}{f_0(X_i)} \qquad \text{if } X\text{'s are iid} \end{aligned}$$

- The *log-likelihood ratio*, when the observations are iid, becomes:

$$\begin{aligned} \log LR &= \sum_{i=1}^{n} \log \frac{f_1(X_i)}{f_0(X_i)} \\ &= \sum_{i=1}^{n} \log f_1(X_i) - \sum_{i=1}^{n} \log f_0(X_i) \end{aligned}$$

- The hypotheses for the corporate payments data:
  - $\mathcal{H}_0 : X \overset{\text{iid}}{\sim} Benf$
  - $\mathcal{H}_1 : X \overset{\text{iid}}{\sim} Cat(p_1, p_2, ..., p_9)$ where $\{p_k\}$ do *not* match Benford's probabilities.
- There are many reasonable choices for setting $\mathcal{H}_1$ parameters $(p_1, \ldots, p_9)$
  - *Discrete Uniform*: $p_1 = \ldots = p_9 = 1/9$
  - *MLE*: $\hat{p}_k = y_j/n$ (This would be the most common)

Your Turn #2

Write out the log-likelihood ratio for the MLE alternative hypothesis.

Using MLE, the $\log LR = 2009.39$.

- Note: $2 \times \log LR$ has an asymptotic chi-squared distribution (same as the chi-squared test statistic).
  - Thus, both statistics provide similar information

2.3 Testing

- The two test statistics, χ^2 and $\log LR$, provide evidence against $\mathcal{H}_0$.
- But how do we know if these values of the test statistics are *unusually* large? Perhaps by chance alone the values are as large as they are.
- We can answer this with a solid probabilistic statement if we knew the *distribution of the test statistic under the null hypothesis*
  - We don't often know this exactly, but there are usually good approximations that hold as the sample size grows (asymptotically).
  - In the computer age, another option is simulation
- There are two primary options:
  1. Use an asymptotic distribution (e.g., the chi-squared distribution)
  2. Use Monte Carlo simulation

2.3.1 Monte Carlo Simulation

- If we can sample from the null hypothesis, then it becomes straightforward to estimate the distribution of *any* test statistic and consequently, p-values.

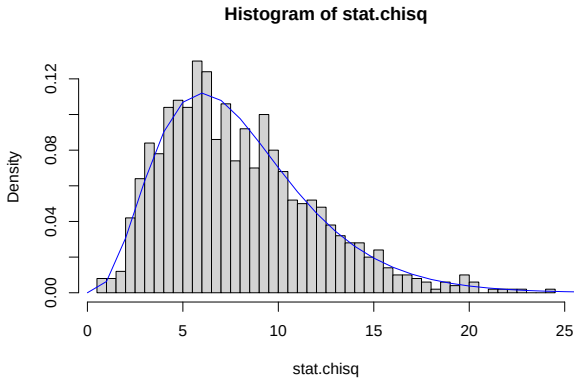
Algorithm: Monte Carlo GOF Test

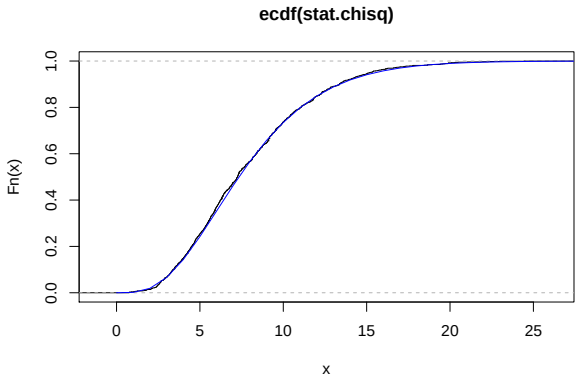
1. Calculate the test statistic, t, from the original data D.
2. Generate M data sets, the same size as D, from the null distribution
   - $\{D_1, \ldots, D_M\}$
3. For each simulated data set, calculate the test statistic T^*
   - $\{T_1^*, \ldots, T_M^*\}$

4. p-value = $\frac{1+\text{number of } T^*\text{'s greater than or equal to } t}{M+1}$

```r
#-- Monte Carlo based p-value
M = 1000                              # number of simulations
stat.chisq = numeric(M)               # initialize statistic
for(m in 1:M){
  #- generate observation under the null of Benford
  y.sim = rmultinom(1, size=n, prob=dbenford(1:9))
  #- calculate test statistic
  stat.chisq[m] = chisq.test(y.sim, p=dbenford(1:9))$statistic
}

#- calculate p-values
(1 + sum(stat.chisq > chisq)) / (M+1)   # chi-square p-value
#> [1] 0.000999
```

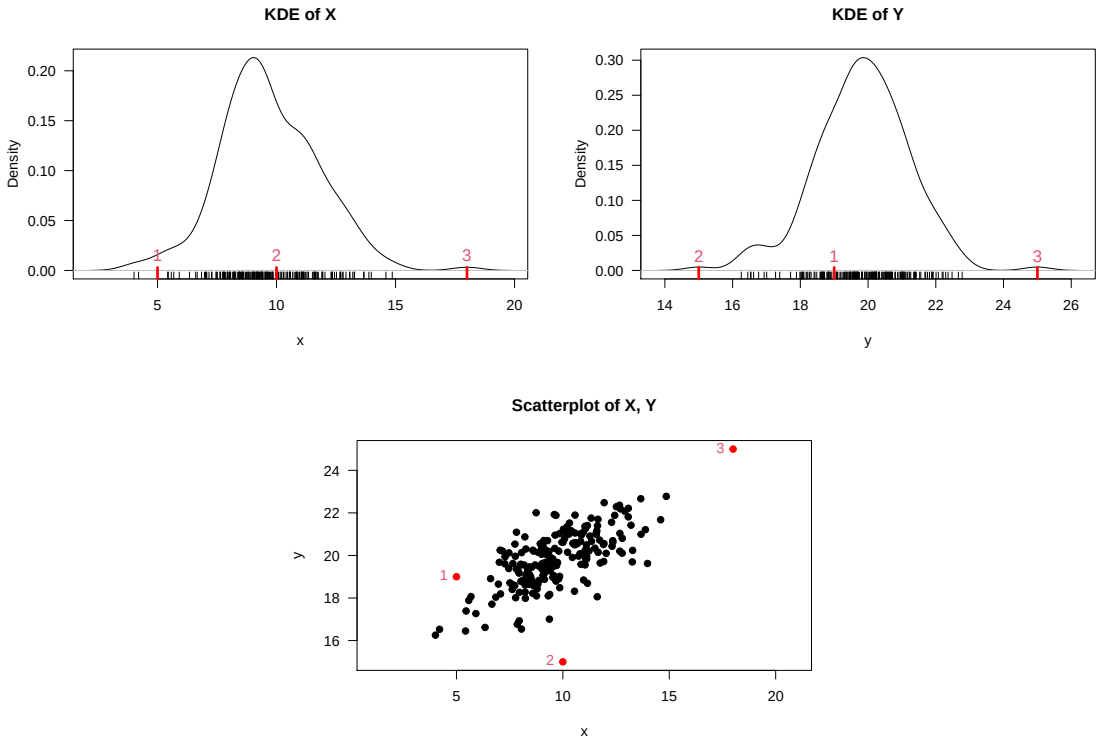
```r

#- plots (Notice: the monte carlo distribution is similar to the chi-square)
hist(stat.chisq, breaks=50, freq=FALSE) # histogram of simulated chisq statistics
lines(0:40, dchisq(0:40, df=8), col="blue")          # overlay the asympotic distribution


plot(ecdf(stat.chisq))                          # ECDF of simulated chisq statistics
lines(0:40, pchisq(0:40, df=8), col="blue") # overlay the asympotic distribution
```





3 Outlier Detection

- Outlier Detection tests if a single observation or small set of observations come from the same distribution as the rest of the data
- There are strong connections between outlier detection and mixture models and model based clustering
- Are any of the *red* points outliers?



3.1 Distance based approach

- One approach to outlier detection, with strong connections to clustering, is to calculate the *distance* from an observation to the centroid
  - This assumes the "normal" (non-outlier) observations are from a unimodal distribution
  - To allow for an ellipse shape (orientation) and different spreads in each dimension, use the squared *Mahalanobis Distance*

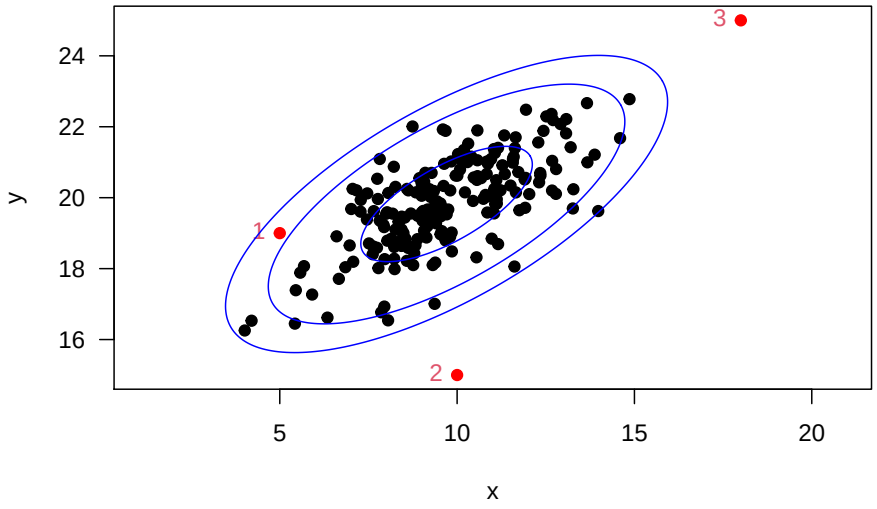
$$D_i^2 = (\mathbf{x_i} - \bar{\mathbf{x}})^\top \hat{\Sigma}^{-1} (\mathbf{x_i} - \bar{\mathbf{x}})$$

  - Note that $\bar{\mathbf{x}}$ and $\hat{\Sigma}$ are estimated from all of the data.
- Estimated Parameters:

$$\bar{x} = 9.70, 19.82$$

$$\hat{\Sigma} = \begin{bmatrix} 4.22 & 1.95 \\ 1.95 & 1.91 \end{bmatrix}$$

obs	x	y	Dsq
1	5	19	7.056
2	10	15	24.471
3	18	25	18.123



3.2 Likelihood Based Approach

- From the scatterplot, it appears the a 2D Gaussian/Normal model could be a decent approximation to the distribution of the non-outlier observations
- We can use this to calculate the *log-likelihood* of observation i using the (MLE) estimated parameters

Gaussian Log-Likelihood

$$\log L_i = \mathcal{N}(\mathbf{x}_i; \mu = \bar{x}, \Sigma = \hat{\Sigma})$$
$$= $$

- Notice that this is a function of the squared Mahalanobis distance!
- Robust estimation:
  - If indeed we have outliers, then these will be affecting out estimated parameters $\bar{x}$ and $\hat{\Sigma}$.
  - Robust estimation techniques can help limit the damage caused by the outliers

- Another, more structured approach, is mixture models!

3.3 Mixture Model Approach

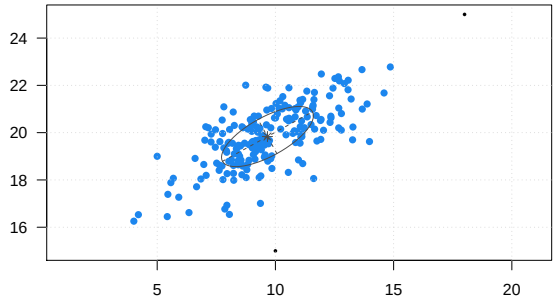
- We can go back to our mixture model formulation and propose that the outliers come from a different distribution than the normal observations
- In mixture formulation

$$f(x) = \pi h(x) + (1 - \pi)g(x)$$

  - $h(x)$ is the pdf for the *outliers*
  - $g(x)$ is the pdf for the *normal observations*
  - π is the prior probability that an observation will be an outlier
  - $E[\text{number of outliers}] = n\pi$
- There are several options for the outlier distribution $h(x)$
  - The uniform distribution on the *minimum bounding box* is one simple approach
    - $h(x) = 1/V$, where V is the volume of the bounding box
    - $V = \prod_{j=1}^{p} (max(x_j) - min(x_j))$
- The `Mclust()` function in the R package `mclust` permits this formulation using the `initialization=list(noise=TRUE)` argument

```
#-- Fit mixture model with uniform noise
library(mclust)
mc = Mclust(X, initialization=list(noise=TRUE), verbose=FALSE)
summary(mc)
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust VVE (ellipsoidal, equal orientation) model with 1 component and a noise
#> term:
#>
#>  log-likelihood   n df   BIC   ICL
#>          -713.4 203  7 -1464 -1473
#>
#> Clustering table:
#>   1   0
#> 201   2
```

```
plot(mc, what="classification", asp=1, las=1)
grid()
```


24
22
20
18
16
5
10
15
20

4 Two-Sample Testing (A/B Testing)

4.1 Example (A/B Testing): Clinical Trials

A placebo-controlled randomized trial proposes to assess the effectiveness (i.e., cure rate) of Drug A in curing infants suffering from sepsis. A clinical trial of $n = 600$ infants using Drug A found that 40% were cured of sepsis while 36% of the $n = 1200$ infants on a placebo were cured.

Your Turn #3

1. Is Drug A better than the placebo?
2. How much better?

Let $p_1 = \Pr(\text{ cure} \mid \text{Drug A})$ and $p_2 = \Pr(\text{ cure} \mid \text{Placebo})$

- $\mathcal{H}_0 : p_1 = p_2$ or $p_1 - p_2 = 0$
- $\mathcal{H}_a : p_1 > p_2$ or $p_1 - p_2 > 0$

The 1800 patients were randomly assigned to the treatment (Drug A) or placebo group. It turned out that:

- of the $n_1 = 600$ given Drug A, $n_1\bar{p}_1 = 600(0.4) = 240$ were cured
- of the $n_2 = 1200$ given Drug A, $n_1\bar{p}_2 = 1200(0.36) = 432$ were cured
- of the $n_1 + n_2 = 1800$ patients, a total of 672 (37.3%) were cured

Under the null hypothesis, $\mathcal{H}_0 : p_1 = p_2$, there is no real difference in the cure rate between treatment and placebo. The observed difference is due **only** to the random assignment.

4.1.1 Simulation Based Testing (Permutation Test)

We can see what the outcomes would have been if we used a different assignment into treatment and placebo.

- Regroup all patients
- Draw n_1 samples, at random, and calculate $\bar{p}_1^*$
- Use the remaining n_2 to calculate $\bar{p}_2^*$
- Calculate the test statistic $Z^* = (\bar{p}_1^* - \bar{p}_2^*)$

This is a possible outcome if the null hypothesis was actually true.

- If we repeat this procedure for all possible re-groupings, then we get the exact[1] distribution of the test statistic, if the null hypothesis was true.

[1] conditional on the data and study design

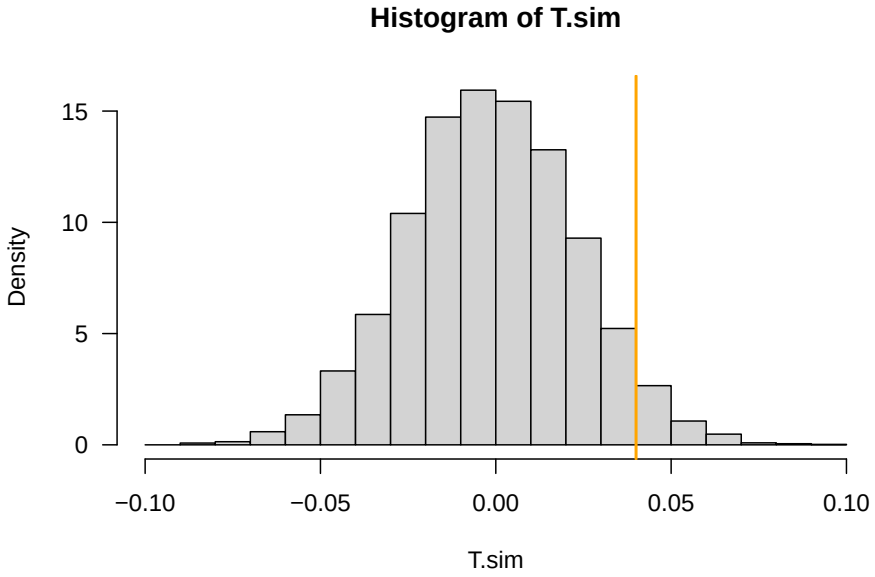
- But, there are $\binom{1800}{600}$(*huge number*) possible regroupings (permutations)
- Monte Carlo simulation can be used to approximate this distribution
- Just repeat the re-grouping procedure many times (say 1000 or 10000)
  - gives a set of observed values under the null model
- The estimated p-value is the proportion of simulated test statistic values that are more extreme than the **observed** test statistic

4.1.2 R Code

```r
#- observed data
n1 = 600
p1 = 0.40
n2 = 1200
p2 = 0.36
p0 = (n1*p1 + n2*p2)/(n1+n2)  # average cure rate
T.obs = p1 - p2        # Test Statistic: observed difference
```

```r
#- Simulation Data
n = n1 + n2             # number of patients
x = n1*p1 + n2*p2       # total number cured

#- Run Simulation
set.seed(100)                              # set seed for replication
nsim = 10000                               # of simulations
x1.sim = rhyper(nsim, m=x, n=n-x, k=n1)    # simulated # cured in pop 1
x2.sim = x - x1.sim                        # simulated # cured in pop 2
T.sim = x1.sim/n1 - x2.sim/n2              # simulated test statistics
```

```r
#- plots
hist(T.sim, breaks=seq(-.1,.1,by=.01), freq=FALSE, las=1) # histogram
abline(v=T.obs,col="orange",lwd=2)         # add observed test statistic
```



```r
#- p-value
(sum(T.sim >= T.obs) + 1) / (nsim +1)     # non-parametric p-value
#> [1] 0.05349
```

4.2 Friedman's Supervised Modeling Approach to Testing

- Worth reading: https://statweb.stanford.edu/~jhf/ftp/gof

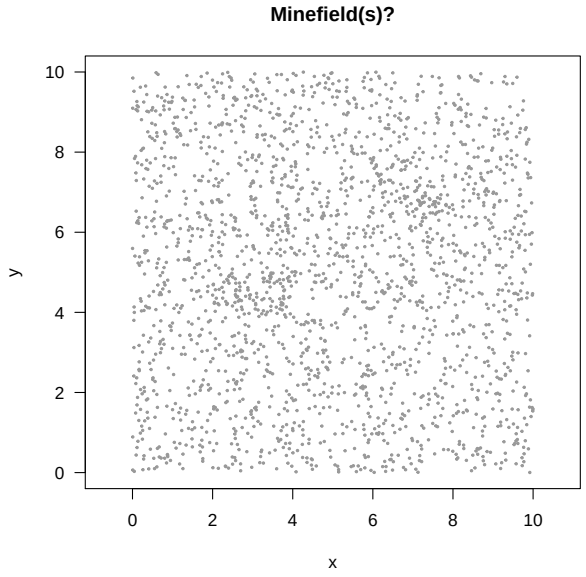
5 Hotspot Detection

Identification of regions that have *unusually* high density

5.1 Example: Land Mine and UXO Detection

Ground penetrating radar is able to detect the presence of land mines and unexploded ordinance (UXO). However, it also detects clutter; rocks and non-interesting metallic objects.

- The items of interest will appear as hotspots
- Uncertain shape and direction



5.2 Hotspot mixture model

$$f(x) = \pi_0 g(x) + \sum_{k=1}^{K} \pi_k f_k(x)$$

- $g(x)$ is the pdf for the *normal/background observations*
- π_0 is the prior probability that an observation comes from the background distribution
- $f_k(x)$ is the pdf for the kth *hotspot*
- π_k is the prior probability that an observation comes from the kth hotspot
- K is the number of hotspots
- $E[\text{number of observations in hotspots}] = n \sum_{k=1}^{K} \pi_k$

5.2.1 Considerations

- The success of this formulation will depend on the forms and restrictions imposed on the components
- If $g(x)$ is allowed too much flexibility, then no hotspots will be detected
- Restrictions should be put on $\sum_k \pi_k$ to prevent the hotspot components from being dominant

- The form of the hotspot component densities $f_k(x)$ should match the expected shapes if a hotspot were actually to occur.
  - E.g., if a hotspot represents a mine field, then the allowable shapes of f_k should match what is possible/probably in mine fields (prior info)
  - E.g., if we think the hotspots will be circular, then restrict f_k to have $\Sigma_k = \lambda_k I$

5.2.2 `mclust` R package

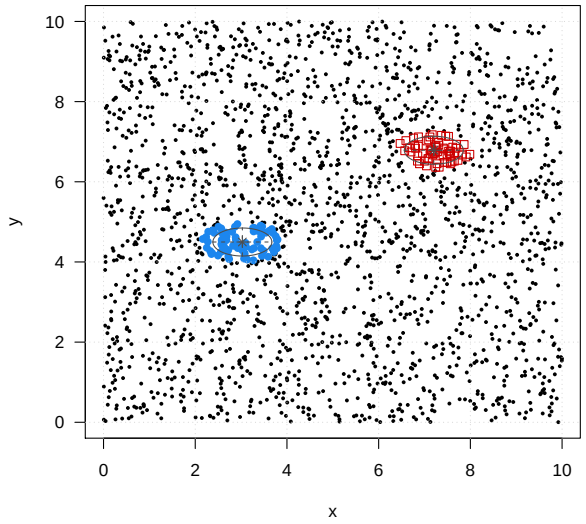
- The `Mclust()` function in the `mclust` R package can facilitate the hotspot mixture model *only if the background distribution is uniform.*
  - $g(x) = 1/V$, where V is the volume of the bounding box

$$V = \prod_{j=1}^{p} (max(x_j) - min(x_j))$$

  - This is the same setup as was used in outlier detection, but reversed (so to speak) in the sense that we expect most observations to come from the background (uniform) component.
- Recall that `Mclust()` chooses the number of clusters and their form by optimizing BIC
  - If we want more control, we can specify the `G=`, `modelNames=`, or `prior=` arguments

```
# Note: X is the two column matrix of point coordinates
library(mclust)
Kmax = 4
mc = Mclust(X, G=1:Kmax,      # set of hotspots to consider
            initialization=list(noise=TRUE), # uniform background
            verbose=FALSE)    # don't show progress bar
summary(mc)
#> ----------------------------------------------------
#> Gaussian finite mixture model fitted by EM algorithm
#> ----------------------------------------------------
#>
#> Mclust EEI (diagonal, equal volume and shape) model with 2 components and a
#> noise term:
#>
#>  log-likelihood    n df    BIC    ICL
#>           -9820 2144  9 -19709 -19973
#>
#> Clustering table:
#>    1    2    0
#>   73   64 2007
```

```

plot(mc, what="classification", las=1)
grid()
```

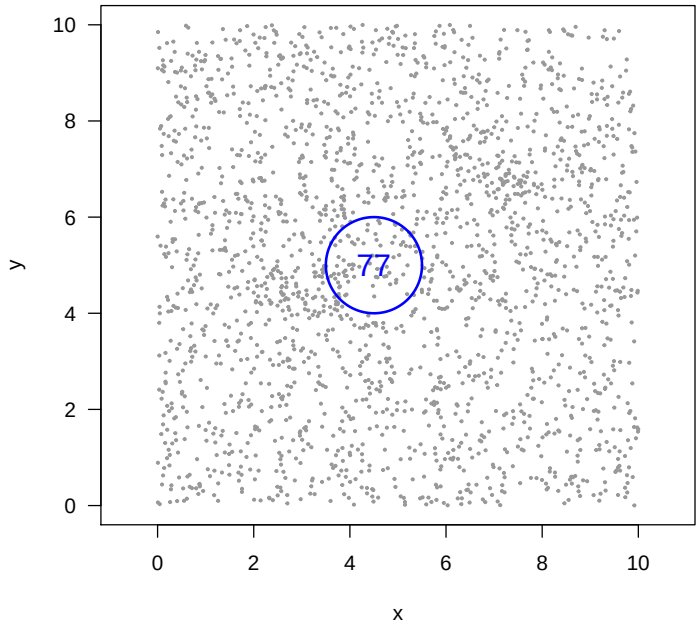


5.3 Spatial Scan Statistic

The spatial scan statistic is an approach to detect spatial *hotspots* (sometimes called *clusters*) that adjusts for *multiple testing*.

5.3.1 Set-up

- Let S be a spatial region (e.g., circle)
- Let $N(S)$ be the *number of events in* S
- Only focus on the region S; ignore everywhere else for now (but we eventually need to scan over all regions)



- **Model:** $N(S) \sim Pois(\lambda_S)$

$$\Pr(N(S) = y \mid \lambda_S) = \frac{\lambda_S^y e^{-\lambda_S}}{y!}$$

- **Hypotheses:**

No Outbreak	$\mathcal{H}_0: \ \lambda(S) = \lambda_S$	
Outbreak in S	$\mathcal{H}_1(S): \ \lambda(S) = q\lambda_S$	where $q > 1$

Your Turn #4

Under $\mathcal{H}_1(S)$, given a count $N(S) = y$, what is the MLE for q, when λ_S is *known*?

- **Test Statistic: D(S)**
  - Use Likelihood Ratio
  - Consider that λ_S is *known*

D(S): Likelihood Ratio Test Statistic

5.3.2 Significance Testing

- Simulate N_0's according to $\mathcal{H}_0$. $N_0 \sim Pois(\lambda_S)$
- Calculate $D_0(S)$ for each simulation
- Estimated p-value is the proportion of times that the simulated D_0's exceed y (observed)
- Example:
  - For the data in the plot, suppose $\lambda_S = 63$ for region S (blue circle)
  - We observed $N(S) = 77$
  - $D(S) = (\frac{77}{63})^{77} \exp(63 - 77) = 4.2701$

$$\begin{aligned}\Pr(D(S) \geq 4.2701 | \mathcal{H}_0, \lambda_0 = 63) &= \Pr(N_0(S) \geq 77 | \lambda_0 = 63) \\ &= \text{R: 1-ppois(77-1, lambda=63)} \\ &= 0.0479\end{aligned}$$

  - *Significance!* Are you ready to conclude that S is a *hotspot*?

That number is a p-value.

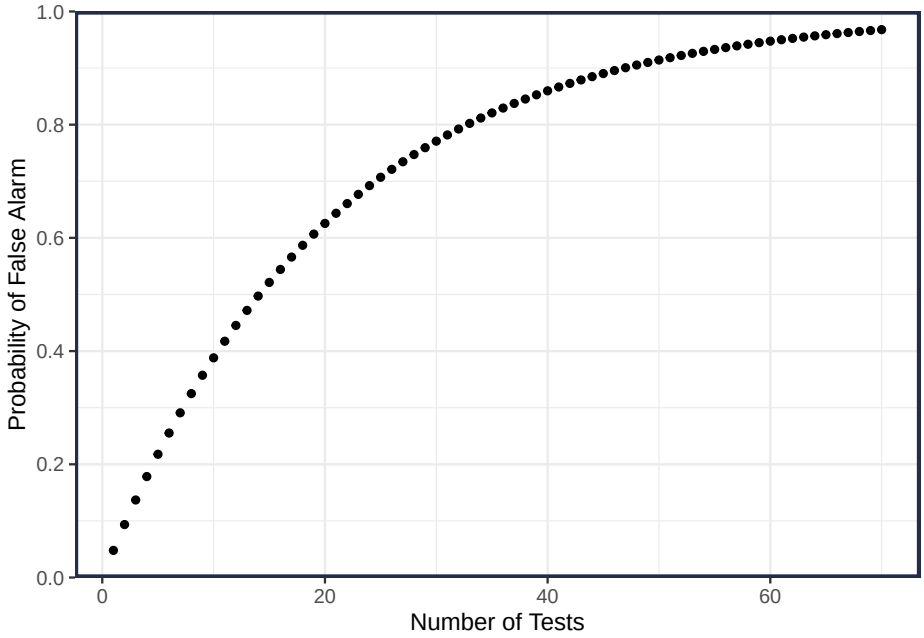
5.3.3 Multiple Testing

- Before we jump the gun, remember that we want to *search the entire area* for hotspots.
- If we considered a different region S', but with same $\lambda_0 = 63$, what is the probability that we would get $D(S') \geq 4.2701$, even if S' is not a hotspot?

Your Turn #5

If we searched 5 non-overlapping regions $S_1, \ldots, S_5$ (all with $\lambda_0 = 63$), what is the probability that *at least one* will have $D_j(S) \geq 4.2701$, even if there are *no hotspots*?

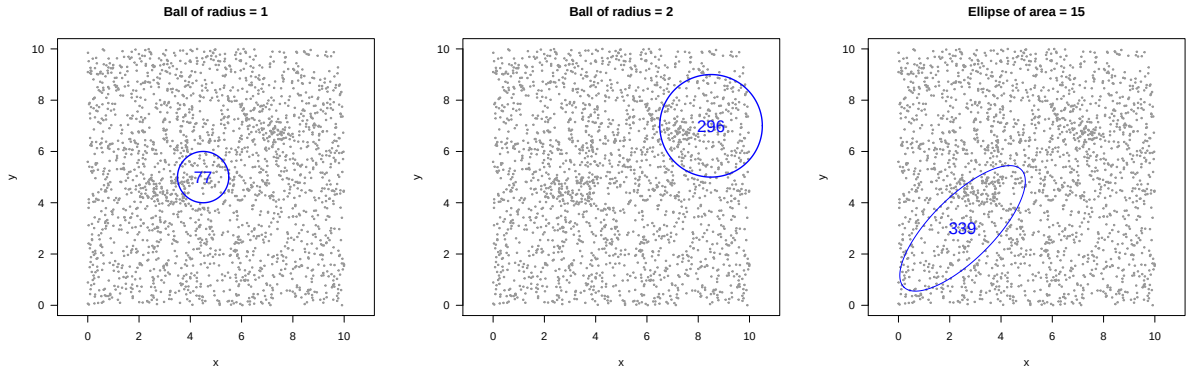
- The probability of at least one *false alarm* starts to grow fast as the *number of tests* increases



- There are a few options:
  1. Bonferroni: Only reject $\mathcal{H}_0$ if p-value $\leq \alpha/K$, where K is the number of tests/comparisons
     - Can be too conservative if the tests are *correlated*
  2. Scan Statistics (i.e., only test significance of the *maximum* score)

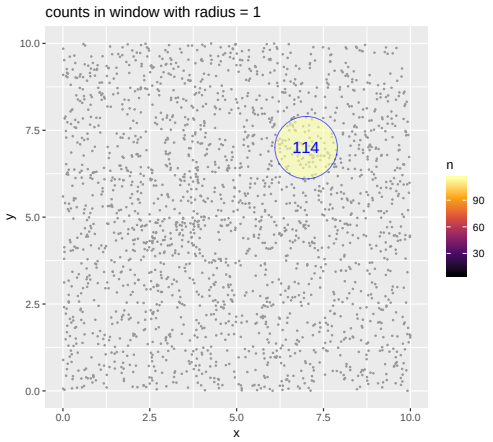
5.3.4 Spatial Search

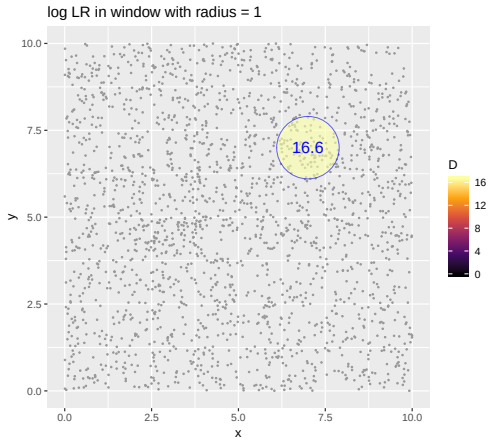
- Let $\mathcal{S}$ be the *search set*, the set of spatial regions that will be searched
  - Every S will be in $\mathcal{S}$ (i.e., $S \in \mathcal{S}\ \forall S$)
- The spatial regions may be restricted by *size* and/or *shape*. E.g.,
  - all balls/circles of radius r (or areas πr^2)
  - All ellipses of area a



- *Scan* the search windows over all locations and record the counts, $N(S)$ and corresponding discrepancy scores $D(S)$.

Click on image to see animation





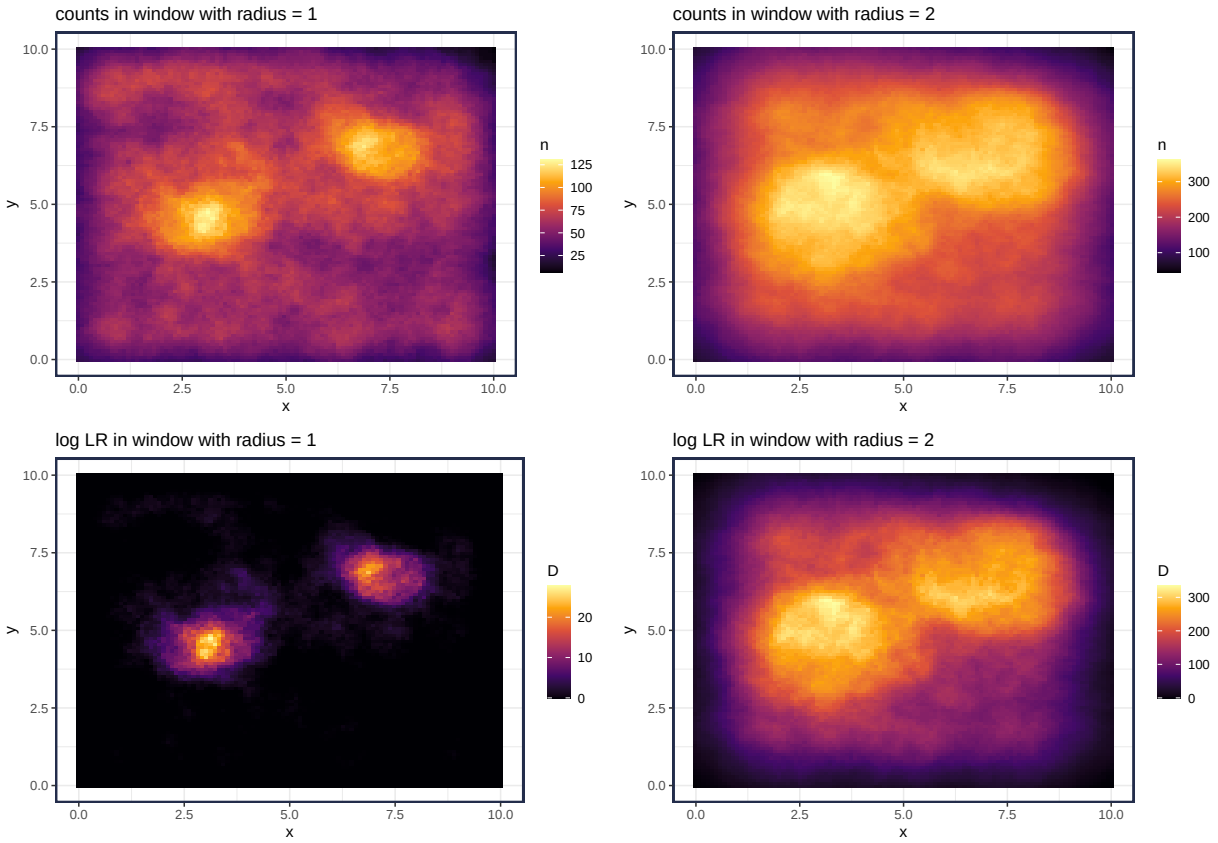
5.3.5 Scan Stat (Test Statistic)

- The *spatial scan stat* test statistic, for balls of radius r, is the discrepancy score (log LR) for the region that gives the maximum discrepancy

$$D_r^* = \max_{S \in \mathcal{S}_r} D_r(S)$$

  - So, move the balls over the entire area, calculate the log LR for each ball, and use the largest value as the test statistic

5.3.6 Results



- NOTE: There is strong connection to kernel density estimation! Do you see how we can build the search from special KDE functions?

- Maximum Scores (aka the *scan statistics*)

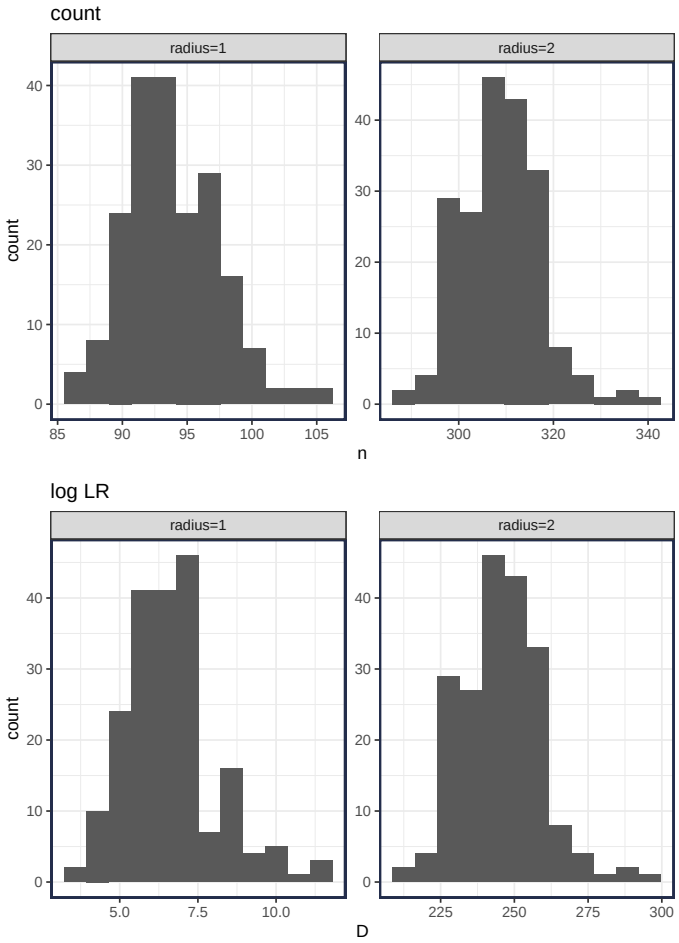
radius	x	y	n	D*
1	3.13	4.75	131	27.9
2	3.13	5.96	363	335.7

5.3.7 Monte Carlo Significance Testing

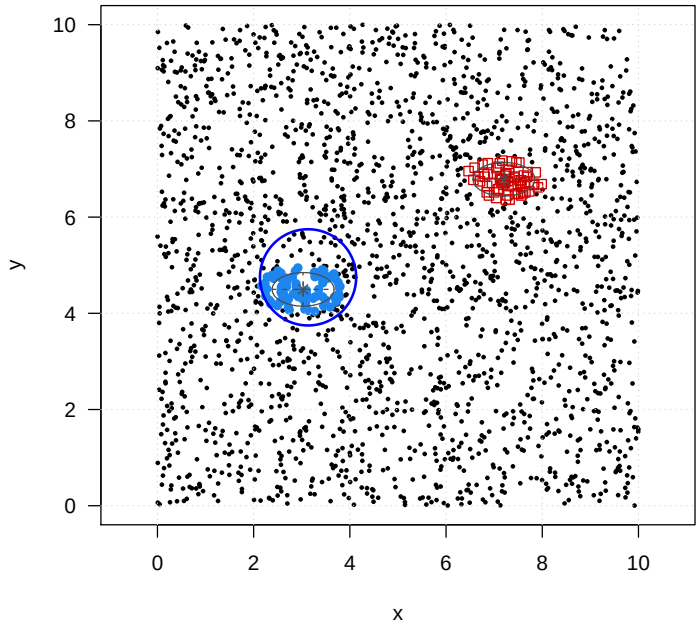
- Is one of the D^*'s large enough to conclude there is a *hotspot*?
- What radius should be used?
- To answer these questions, we need to find the distribution of D^* under the null hypothesis of *no hotspots*

1. **Simulate from the null hypothesis**
   - We assume the data came from a *spatial Poisson point process* with known intensity $\lambda(x)$
   - Simulate n events (n is the number of events we observed)
   - Density is $f(x) = \lambda_0(x) / \int \lambda_0(u)\, du$

2. **Search for Hotspots**
    - Must choose the shapes/sizes for the potential hotspots
3. **Record scan statistics**
4. **Redo steps 1-2 M times**
5. **Calculate p-values**
    - p-value $= \frac{1+\sum_m \mathbb{1}(D_m^* \geq D_{\text{obs}}^*)}{1+M}$



- No simulation exceeded our observed scores ($D_{r=1}^* = 27.90$ and $D_{r=2}^* = 335.71$).
    - Thus, very small estimated p-values (i.e., $1/(M+1)$)
    - Thus, there is likely *at least one hotspot* (i.e., at least one circle of radius 1 and radius 2 that has an *unusually* large count)



5.4 Network Scan Statistic

- The *network scan statistic* is very much like the spatial scan statistic, the difference being the hotspots are a *connected set of nodes*

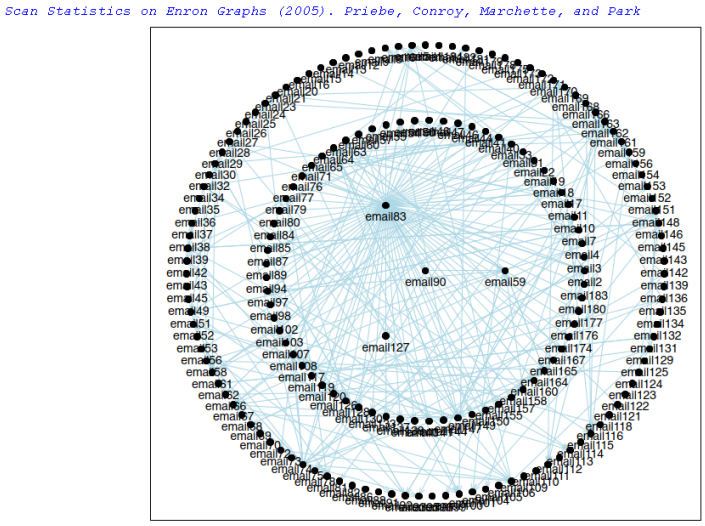


Figure 4. Plot of the 'detection' Enron email graph D_{132} (sans isolates) for which our scan statistic methodology detects an anomaly. The center vertex, *email*90, is $v^* = \arg\max_v \tilde{\Psi}_{2,132}$.

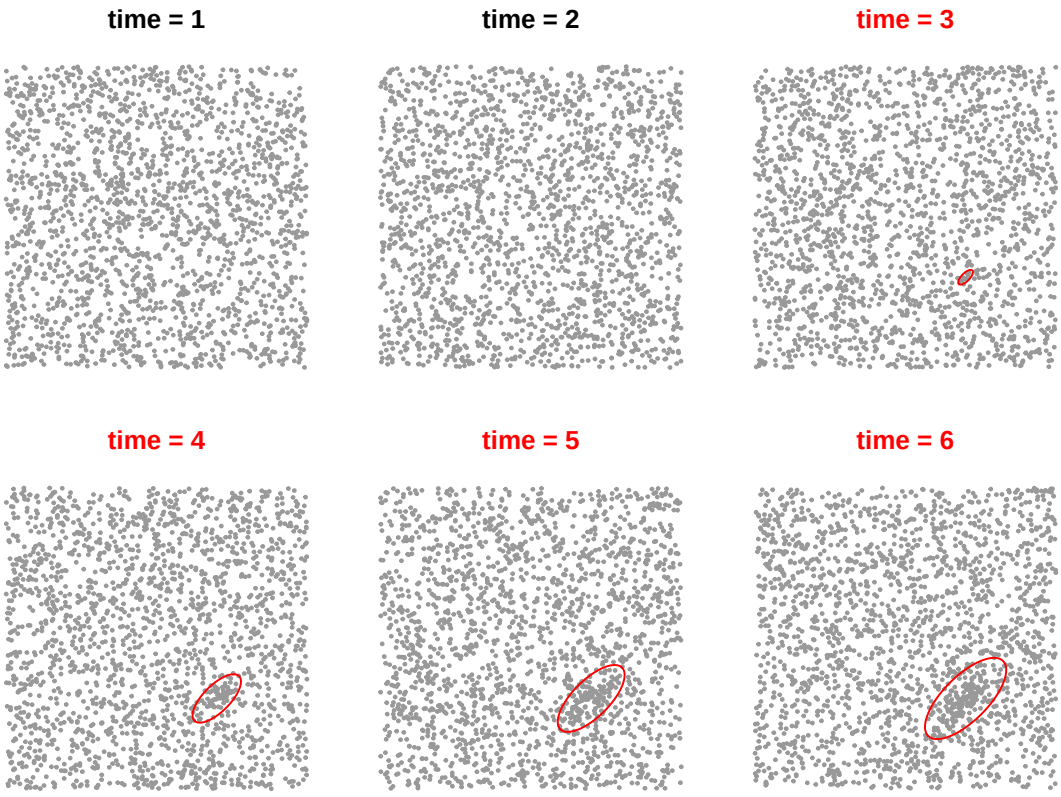
- The computational challenge of searching all reasonable hotspots is substantial

6 Outbreak Detection

- Consider having the spatial observations occurring over time.
  - E.g., daily, weekly, hourly
- Denote $N(t, S)$ as the number of events in spatial region S at time period t
- There is a time point τ when an outbreak starts

$$\mathcal{H}_0: \quad \lambda(t, S) = \lambda_0(t) \qquad \text{for all } t$$

$$\mathcal{H}_1(\tau, S): \quad \lambda(t, S) = \begin{cases} \lambda_0(t) & \text{for } t < \tau \\ q(t)\lambda_0(t) & \text{for } t \geq \tau \text{ and } q(t) > 1 \end{cases}$$



- One option is to compute $D^*(t)$ over each time t and look for the time when there is an increase
  - Use Statistical Process Control (SPC, control charts)
- If the signal (i.e., outbreak) is small, it may take many time periods before there is enough evidence to detect the outbreak
  - Another option is to include time into the likelihood ratio test statistic
  - See: Kulldorff, M. (2001), Prospective time periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164: 61-72 for more details