Statistical Anomaly Detection for Cybersecurity

Brian Terry

Supervised by Ben Powell

Abstract

In this paper, we will use the motivating example of detecting network attacks to study statistical anomaly detection. We will introduce the concept of hypothesis testing and the ways we can model aspects of networks. We will cover problems which develop with single hypothesis testing due to large sample sizes. We will cover Family Wise Error Rates, looking at the Bonferroni bound as well as the Šidák, Holm and Hochberg procedures. We will briefly cover the k-FWER criteria. We will also look at the Benjamini-Hochberg procedure which addresses the topic of False Discovery Rates.

This dissertation is submitted for the degree of
BSc Mathematics

Department of Mathematics
University of York
15th June, 2020

For my grandma.

Contents

Chapter 1

Introduction

Statistical anomaly detection, the science of making judgements based on a statistically analysed scenario, is present in all levels of industry and research. It has seen an explosion of popularity in the past century with the introduction of hypothesis testing and is known by almost any university undergraduate studying STEM. Statistical anomaly detection has had to keep up with the digital age in many ways. Methods which were originally designed to handle a handful of sample data are now facing hundreds of thousands of inputs. More robust methods are needed to handle these cases.

Parallel to this, cybersecurity has undeniably become an important part of our lives. The world's reliance on computers is increasing at an astonishing rate - more and more our day to day tasks are facilitated by technology. This however means cyberattacks are all the more devastating. Companies and governments need fast, efficient procedures to detect attacks so they can respond effectively.

This project examines recent developments within the field of statistical anomaly detection and looks to apply it to cybersecurity. It is important to consider the fact we not only want to detect when a computer network is compromised/under attack but in addition not shut down the entire network over a false alarm - wasting valuable time and resources.

In chapter 2 of this paper, we will introduce the concepts of networks and models in cybersecurity, briefly introducing node, edge and network based models. Further to this we will cover the fundamentals of hypothesis testing, defining the language around probability distributions as well as several relevant distributions. We will explain what hypotheses actually are, the concept of type I and type II errors in addition to the size and power of tests.

In chapter 3, we will cover multiple hypothesis testing, introducing the multiple comparisons problem and providing several ways to tackle it, including a look at Family Wise Error Rates (FWER) and the k-FWER criteria. We will outline several procedures, namely the Bonferroni bound as well as the Šidák, Holm and Hochberg procedures which control the FWER. We will give example calculations of all these procedures with 'toy' data.

Finally, in chapter 4, we will explain the concept of false discovery rates (FDR), outline the Benjamini-Hochberg (BH) Algorithm as well as see examples of how to estimate the FDR and how to use the BH algorithm. We will consider the advantages to using the BH Algorithm as well as acknowledging some of it's pit falls (although we will not cover how to overcome them as the solutions are beyond the scope of this paper).

Chapter 2

Cybersecurity and Hypothesis Testing

2.1 Secure Networks

In the first class of MIT's 2014 course 'Computer Systems Security' [1], Nickolai Zeldovich lays out that when designing secure computer networks, we need to consider three core tenets:

1) Policy - How we go about ensuring confidentiality, integrity and accessibility of data.

2) Threat Model - Assumptions we make about our adversary and his scope.

3) Mechanism - The software, hardware and system we use to store and access the data.

He highlights there is always a trade off in designing a secure system. We want our network and the data stored within it to be accessible as much as it is secure, and for our data to be as complete as it can be. This means balancing our policy decisions to not be at one extreme or the other. It's also unrealistic to assume an all powerful adversary. We need to put restrictions on what a hacker can and can't do - although it's best to be conservative with this. Most hackers employ relatively ingenious tricks to circumnavigate security measures! The process of designing secure systems is an iterative one. As the security analyst improves their system, an attacker also improves his methods of attack.

James Mickens, in the second class of the course [2], highlights once an attacker has hijacked a network process, the attacker can do anything they want to with the privileges and priority of that process. If the process is allowed to read confidential files, it can read those files; if it can send emails, we have a potential for a spam email campaign. It's important to consider that this would undermine a firewall to stop non-trusted computers from influencing the network effectively meaning our infected computers look on the surface like secure computers.

2.2 Models in Cybersecurity

Using statistical models in cybersecurity allows us to use historical data. This has the major advantage of fine tuning our procedures before an attack has taken place. Researcher Nick Heard in his 2018 talk 'Data Science in Cyber-Security and Related Statistical Challenges' for Microsoft Research [3] suggests that historical data can be used to build a model for 'normal behaviour' of the network. We can then compare this against a network's future behaviour to see if it has been compromised.

Networks can be viewed as graphs - structures of nodes and edges. As an example figure 1.1 shows a model network. The circles A, B, C and D are all nodes in this graph. They are connected by various (directed) edges (the arrows).

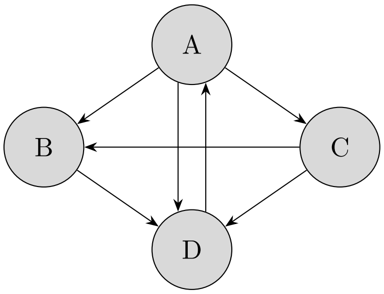

Figure 2.1: A model network.

Graphs have many uses. Within the context of cybersecurity, we can view nodes as access points for data to be stored or viewed by a user. Edges can then be interpreted as (one-way) paths for traffic to flow through between two nodes. The network is simply a closed collection of nodes and edges, representing how data (a.k.a. network traffic) transfers between access points, i.e. the graph itself.

Heard [3] suggests, when deciding how to model behaviour within the network we can look at three kinds of models:

1) Node-based models.

2) Edge-based models.

3) Overall network models.

These models let us use statistical methods to detect anomalies within them indicative of an attack allowing us to respond accordingly.

2.3 Basic Concepts of Probability and Statistical Distributions

When talking about statistical anomaly detection, we need to understand some basic definitions used in probability and statistics. Chiefly among them is probability mass and density functions. Gustav Delius in his 2017 lecture series 'Introduction to Probability & Statistics' [4] sets out good definitions for these concepts which I've reproduced below (edited only to keep a consistent style).

Definition 1. *Probability Mass Function: Given a sample space Ω and a discrete random variable X, the function $p_X : R \to [0,1]$ defined by*

$$p_X(x) = P(X = x)$$

is called the probability mass function of X.

Definition 2. *Probability Density Function: Let X be a continuous random variable then its density function f_X satisfies*

$$f_X(x) \geq 0, \text{ for all } x \in \mathbb{R};$$

$$\int_{-\infty}^{\infty} f_X(x)dx = 1.$$

Intuitively speaking, the probability mass function in definition 1 represents the probability of a random variable X taking on the value of x. The probability density function, f_X is the continuous analogue of the probability mass function. An important fact is that f_X equals 0 for any specific value of x and only has value when considering a continuous range for x.

In this paper, we will make use of three probability distributions, namely the Poisson, Uniform and Normal distributions. Gustav Delius gives clear and accurate definitions for these distributions in his notes [4] on pages 33, 37 and 40 respectively. I reproduce these definitions below with only slight changes for clarity.

Definition 3. *The Poisson Distribution: We say that the random variable X has the Poisson distribution with parameter $\lambda > 0$, and write $X \sim Pois(\lambda)$, if the possible outcome is in the set $\{0, 1, 2, ...\}$ and it has mass function*

$$p_X(n) = \begin{cases} \frac{\lambda^n}{n!} e^{-\lambda}, & \text{if } n = 0, 1, 2, ... \\ 0, & \text{otherwise.} \end{cases}$$

Definition 4. *The Uniform Distribution: We say that the continuous random variable X has the uniform distribution on $[a, b]$, and write $X \sim U[a, b]$, if the density of X is*

$$f_X(x) = \begin{cases} \frac{1}{b-a}, & \text{if } x \in [a, b] \\ 0, & \text{if } x \notin [a, b]. \end{cases}$$

Definition 5. *The Normal Distribution: We say that the continuous random variable X has the normal distribution with mean μ and variance $\sigma^2 > 0$, and write $X \sim N(\mu, \sigma^2)$, if the density of X is*

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \text{ for } x \in \mathbb{R}.$$

It is very common to encounter the standard normal random variable $X \sim N(0,1)$ to the end that 'standard notation' has been introduced. The density of the standard normal random variable is often written as

$$\phi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-x^2}{2}\right).$$

The distribution function is written as

$$\Phi(x) = \int_{-\infty}^{x} \phi(s) ds.$$

The variable name z is commonly used in place of x when values of x are sampled from a normal distribution.

2.4 Basic Concepts of Hypothesis Testing

Suppose two computers send information from one to the other in the form of packets. On average ten packets of information are sent every minute. It is observed that for a particular minute, fifteen packets of information were sent. Growing suspicious, we might want to check how likely the event of more than fifteen packets of information being sent is. We can model the two computers sending information between each other as a Poisson process.

Taking $\lambda = 10$ for our Poisson distribution, we can calculate that the probability of fifteen packets of information travelling from one computer to the other is 4.86%. We might grow suspicious at this stage given there is a less than 5% chance of this occurring. This in essence is hypothesis testing. We started with an assumption that nothing was abnormal with our system and then proceeded to find the probability of an event occurring under those assumptions. We now introduce two key terms for hypothesis testing.

Definition 6. *Null Hypothesis: A characteristic of a population assumed true initially.*

Definition 7. *Alternative Hypothesis: An alternative characteristic of a population which may explain the population better than the null.*

Hypothesis testing does not tell us anything about the certainty of the null hypothesis or if the alternative hypothesis is true.

To illustrate the above, consider the following example. A bank account is accessed online once every hour. We monitor these logins to check if there is any attempt by someone to login who shouldn't be accessing the content of the bank account. We say over a sixty hour period, if three attempts are made, we should restrict the access to the account or ask for further validation of identity. Translated into the language of hypothesis tests, we say our null hypothesis is that no one is fraudulently attempting to access the bank account, the alternative hypothesis is that there is cause for concern someone might be attempting to access the bank account for malicious reasons and that the significance level is 5%. One might comment that three attacks over a span of sixty hours is too many and action should be taken more decisively. However, we could also argue that this is too few issues to warrant a response (we've all entered a password incorrectly before). Our outlook on this problem leads us to look at type I and type II errors.

Definition 8. *Type I Error: Rejecting the null hypothesis when it is in fact true.*

Definition 9. *Type II Error: Failing to reject the null hypothesis when it is in fact false.*

These concepts can be alternatively conveyed in the below table:

	Null is False	Null is True
Null Rejected	Acceptable Outcome	Type I
Null not Rejected	Type II	Acceptable Outcome

2.5 Size and Power

Type I and II errors lead us to the concept of size (and by extension significance) and power of a statistical test. The following three definitions are taken from Ben Powell's 2019 lecture series, 'Probability and Statistics II: Statistical Inference 2 and Linear Models' [5].

Definition 10. *Size: The size of a test is the probability that it will reject the null hypothesis when the null hypothesis is true.*

$$\alpha = P(\textit{rejecting } H_0|H_0 \textit{ is true}) = P(\textit{making a type I error})$$

Definition 11. *Significance of a Test: The significance of a test is an upper bound on its size.*

If we define $\beta = P(\text{not rejecting } H_0|H_0 \text{ is false})$ (i.e. the probability of committing a type II error) then we can define statistical power.

Definition 12. *Power: The power of a test is the probability that it will reject the null hypothesis when the alternative hypothesis is true.*

$$\textit{power} = 1 - \beta = P(\textit{rejecting } H_0|H_0 \textit{ is false})$$

Steffen Lauritzen, in the eleventh class of Oxford's 2004 course 'Statistical Inference' [6], lays out what is known as Neyman-Pearson Theory. Ultimately, this paradigm looks to find optimal tests. When designing tests, we want to maximise power under the alternative hypothesis but minimise size. Power depends directly on the alternative hypothesis with size depending directly on the null hypothesis along with the alternative hypothesis. This means the two are linked together forcing us to engage in a trade off. This is an important consideration when designing tests.

But by what metric do we reject the null hypothesis? This is where the concept of a 'p-value' comes in. This is a percentage level that if a sample of data reaches or supersedes we reject the null hypothesis. This figure is commonly set at 5% of a sample. However, the concept has received wide criticism from members of the statistical community as noted by Ronald L. Wasserstein & Nicole A. Lazar making a statement on behalf of the ASA regarding p-values [7]. 5% is an arbitrary value to set our deciding level at. However, when we argue to change it we run into the issue that we can simply set the bar at any level we want it at. This raises serious questions about accuracy of experiments as well as ethics. To this end, arguments and techniques using Bayesian statistics have been developed to try and 'fix' some of the issues that occur with hypothesis testing as covered by Bradley Efron in his 2007 paper 'Size, Power and False Discovery Rates' [8]. We will examine some of these concepts in sections 4.2 and 4.3.

One interesting way of interpreting p-values is Fisher's 'scale of evidence' (as seen in Table 1.1). This 'scale' helps us identify how strong the evidence we have calculated is.

α	0.1	0.05	0.025	0.01	0.001
Evidence against H_0	Borderline	Moderate	Substantial	Strong	Overwhelming

Table 2.1: Bradley Efron gives the above table on page 31 his book 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9]. The scale allows for a more holistic examination of evidence.

Fisher's scale of evidence is an interesting concept worth dwelling on for a moment. This approach allows us to determine an appropriate response to the given situation as opposed to a binary null/non-null. In the context of network model tests, we can use the result of the test to warrant a full network shut down (overwhelming evidence), a local shut down of a sub-network (strong evidence), a shut down of just a singular node/edge (substantial evidence), having a human examine the data being sent along an edge (moderate evidence) or even simply logging the problem for future reference (borderline evidence). This could lead to saving a lot of money (by not repeatedly shutting down networks) whilst allowing for many potential security risks to be caught.

Chapter 3

Multiple Hypothesis Testing

3.1 Intersecting Hypotheses

A hypothesis in this subject can be a set of further hypotheses as Juliet Popper Shaffer lays out in her 1995 paper, Multiple Hypothesis Testing [10]. For instance, H_{ij} contains the hypotheses H_i and H_j.

Definition 13. *Intersection of Hypotheses: A blanket hypothesis which covers multiple hypotheses.*

For instance, the hypotheses H_{ij}, H_{jk} are covered by the intersection H_{ijk}.

Definition 14. *Component of an Intersection Hypothesis: A hypothesis contained within an intersection hypothesis.*

It is important to note we can reduce our work by considering large sets of differences and inferring facts from what we discover with the large scale examination. Consider it this way: if we take H_{ij} to be the hypothesis that the connection between two computer nodes i and j is secure and that $H_{ijk \ldots n}$ is the hypothesis that the entire network consisting of n nodes is secure, then if significant evidence is found to suggest that $H_{ijk \ldots n}$ is false, then we know H_{ij} *might* be false as well and so we can begin sorting through possibilities.

3.2 Multiple Comparisons Problem

A serious issue arises when performing multiple hypothesis testing - namely the possibility that if we do enough tests, we will eventually find a discovery when in fact there is truly nothing there! To put this another way, if we perform one hundred tests at a 5% significance level, and in reality all null hypotheses are true, we will have about five of those tests come back as significant. If some of the null hypotheses are not true, we will still have about five tests erroneously come back as significant along with the test that do return genuinely significant results. This is acceptable in some scenarios but in the majority of cybersecurity related problems, where potentially tens of thousands of computers are being tested, a response that hundreds of them are compromised is cause for an entire network shut down (which would be very costly for the organisation responsible for the network). These problems were not obvious in the initial design of hypothesis testing. Single hypothesis testing was designed in the 1920s and 1930s - before the Computer Age and the possibility to analyse thousands of factors simultaneously. We need to use more robust techniques to effectively handle the problems that quickly crop up with single hypothesis testing.

3.3 Family Wise Error Rates

The Family Wise Error Rate is set out well in Bradley Efron's 2010 book 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9] on pages 34 and 35. First, we associate a series of p-values $p_1, p_2, ..., p_N$ with a series of null hypotheses $H_{01}, H_{02}, ..., H_{0N}$. This leads us to the following definition:

Definition 15. *Family Wise Error Rate (FWER): The probability of making at least one false rejection in a family of hypothesis-testing problems. i.e.* $FWER = P(\text{Rejecting any true } H_{0i})$ *with* $i \in \{1, 2, ..., N\}$.

Definition 16. *FWER Control Procedure: Any procedure that receives a series of p-values $p_1, p_2, ..., p_N$ and tells us if each corresponding null hypotheses $H_{01}, H_{02}, ..., H_{0N}$ is rejected subject to the FWER being lower than a predetermined level, α.*

In a study we will have data, x, that we can use to test $H_{01}, H_{02}, ..., H_{0N}$, the family of hypotheses. Based on this data we will also have the FWER test controlled at a particular α level, denoted by $\text{FWER}_\alpha(x)$. This leads us to the concept of adjusted p-values.

Definition 17. *Adjusted p-value:*

$$\tilde{p_i}(x) = \inf_\alpha \{H_{0i} \text{ rejected by } FWER_\alpha(x)\}$$

There exist a variety of methods designed to control the FWER at a certain level.

3.3.1 Bonferroni Bound

The Bonferroni bound, as outlined in Bradley Efron's 2010 book 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9] on page 35 gives us a simple FWER control procedure allowing us to reject interesting[1] null hypotheses.

Algorithm 1. *Reject H_{0i} if*

$$p_i \leq \frac{\alpha}{N}$$

Verifying this is a quick task. Let I_0 index true hypotheses, there being N_0 true hypotheses.

[1] Efron [9] uses the term "interesting" to describe hypotheses that fall within the critical region as opposed to "significant", a convention I intend to continue in this dissertation.

Observe that:

$$\text{FWER} = P\left(\bigcup_{i\in I_0}\left(p_i \leq \frac{\alpha}{N}\right)\right)$$

Before we can proceed, we need to define define Boole's Inequality. The following definition is taken, without alteration, from Ryan Ta's 2019 class, 'MATH 149A', for the University of California, Riverside [11].

Definition 18. *Let $\{C_n\}$ be an arbitrary sequence of events. Then*

$$P\left(\bigcup_{n=1}^{\infty} C_n\right) \leq \sum_{n=1}^{\infty} P(C_n).$$

Now, by Boole's Inequality:

$$P\left(\bigcup_{i\in I_0}\left(p_i \leq \frac{\alpha}{N}\right)\right) \leq \sum_{i\in I_0} P\left(p_i \leq \frac{\alpha}{N}\right)$$

As $P\left(p_i \leq \frac{\alpha}{N}\right) = \frac{\alpha}{N}$ and given there are N_0 true hypotheses,

$$\sum_{i\in I_0} P\left(p_i \leq \frac{\alpha}{N}\right) = N_0\frac{\alpha}{N} \leq \alpha. \quad \square$$

Example 1. *Suppose we have a network where every edge has one hundred pieces of data transferred per minute. The traffic is distributed according to the Poisson distribution, i.e. $X \sim Pois(100)$. We wish to perform a routine check on the network. Breaking the network down into component edges, we obtain the following vector, $\boldsymbol{x}$, of traffic rates:*

$$\boldsymbol{x} = \begin{pmatrix} 57 \\ 87 \\ 111 \\ 124 \\ 125 \end{pmatrix}$$

The following vector, $\boldsymbol{p}$, contains the p-values (rounded to 3 d.p. for presentation) corresponding to each value of $\boldsymbol{x}$,

$$\boldsymbol{p} = \begin{pmatrix} 1.000 \\ 0.896 \\ 0.126 \\ 0.009 \\ 0.007 \end{pmatrix}$$

As we have five edges in this network, $N = 5$. If we set $\alpha = 0.05$ we have

$$\frac{\alpha}{N} = \frac{0.05}{5} = 0.01$$

Testing if $p_i \leq 0.01$ we note that x_4 and x_5 are detected as 'interesting' when tested with the Bonferroni bound. Hence we may want to consider shutting down (or otherwise further investigating said edges).

3.3.2 Šidák's Procedure

An alternative to the Bonferroni bound is to use Šidák's procedure as the guiding rule for rejecting interesting null hypotheses. Šidák's procedure, as laid out on page 36 of Efron's textbook 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9] is as follows:

Algorithm 2. *Reject H_{0i} if*

$$p_i \leq 1 - (1 - \alpha)^{1/N}$$

This is derived by first considering what it means to control FWER at a particular level, α. We actually find a formula for the adjusted p-value, $\tilde{p_i}$, in doing this. Following a proof modified from Ben Powell's 2020 lecture series, 'Probability and Statistics II: Statistical Inference 2 and Linear Models' [12] we verify the above equation.

Proof.

$$\begin{aligned}\alpha &= \tilde{p_i}(x)\\ &= \inf_{\alpha}\{H_{0i} \text{ rejected by } \text{FWER}_{\alpha}(x)\}\\ &= P(\text{Reject any true null})\\ &= 1 - P(\text{Reject no true null})\\ &= 1 - \prod_{i=1}^{N} P(H_{0i} \text{ is not rejected}|H_{0i} \text{ is true})\\ &= 1 - \prod_{i=1}^{N}(1-p_i)\\ &= 1-(1-p_i)^N\end{aligned}$$

This gives the adjusted p-value as

$$\tilde{p_i} = 1-(1-p_i)^N$$

Working with the above proof effectively backwards gives us

$$\begin{aligned}1-(1-p_i)^N &= \alpha\\ 1-p_i &= (1-\alpha)^{1/N}\\ p_i &= 1-(1-\alpha)^{1/N}\end{aligned}$$

□

An important consideration with Šidák's procedure is that it is only controls the FWER at the specified α level if all the p-values are independent of one another. In realistic situations, it is fairly unlikely they are all independent of one another (traffic on an edge often has a knock on effect on nearby edges) but Šidák's procedure does improve on Bonferroni by making it easier to reject every H_{0i} at any given level. In other words, if all p-values are independent of one another, Šidák's procedure is more powerful than the Bonferroni bound.

Example 2. *Suppose we again have a network where every edge has one hundred pieces of data transferred per minute. As in Example 1 the traffic is distributed according to the Poisson distribution, i.e.* $X \sim Pois(100)$. *The vector,* $\boldsymbol{x}$, *of traffic along each of the network's edges is*

$$\boldsymbol{x} = \begin{pmatrix} 90 \\ 13 \\ 124 \\ 82 \\ 75 \\ 125 \end{pmatrix}$$

The following vector, $\boldsymbol{p}$, contains the p-values (rounded to 3 d.p. for presentation) corresponding to each value of $\boldsymbol{x}$,

$$\boldsymbol{p} = \begin{pmatrix} 0.829 \\ 0.001 \\ 0.009 \\ 0.963 \\ 0.995 \\ 0.007 \end{pmatrix}$$

As we have six edges in this network, $N = 6$. If we set $\alpha = 0.05$ we can calculate the threshold value

$$1 - (1 - \alpha)^{1/N} = 1 - (1 - 0.05)^{1/6} = 0.008$$

Testing if $p_i \leq 0.008$ we note that x_2 and x_5 are detected as interesting when tested with the Šidák procedure. Hence we may want to consider shutting down (or otherwise further investigating) edges x_2 and x_5.

3.3.3 Holm's Procedure

While the Šidák procedure can't be considered a general improvement over Bonferroni, what is known as the Holm's procedure can be. On page 36 of Efron's textbook, 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9], Holm's procedure is laid out.

Algorithm 3. *This FWER control procedure begins by ordering the p-values in ascending order, i.e.:*

$$p_{(1)} \leq p_{(2)} \leq ... \leq p_{(N)}$$

We reject the $H_{0(i)}$ associated with $p_{(i)}$ if

$$p_{(j)} \leq \frac{\alpha}{N-j+1} \textit{ for } j = 1, 2, ..., i$$

The adjusted p-value for Holm's procedure is

$$\tilde{p}_{(i)} = \max_{j \leq i}\{\min\{(N-j+1)p(j), 1\}\}$$

Holm's procedure has a larger rejection region than the Bonferroni bound and is thus more powerful. An additional benefit is that it does not require the independence assumption Šidák does. This means we can take a more relaxed view towards knock-on effects edges have on each other.

Example 3. *Suppose we have a network where every node sends out three hundred pieces of data per minute. The traffic is distributed according to the Poisson distribution, $X \sim Pois(300)$. During a routine monitoring period of one minute, the amount of data sent out by nodes is recorded in the vector, $\boldsymbol{x}$, below*

$$\boldsymbol{x} = \begin{pmatrix} 301 \\ 269 \\ 355 \\ 331 \end{pmatrix}$$

The following vector, $\boldsymbol{p}$, contains the p-values (rounded to 3 d.p. for presentation) corresponding to each value of $\boldsymbol{x}$,

$$\boldsymbol{p} = \begin{pmatrix} 0.462 \\ 0.963 \\ 0.001 \\ 0.036 \end{pmatrix}$$

Ordering the p-values, we have

$$p_3 < p_4 < p_1 < p_2$$

As we have four nodes in this network, $N = 4$. We set $\alpha = 0.05$. Starting at the largest p-value, p_2, we set $j = 4$ (as p_2 is fourth in our ranking of the p-values) and calculate

$$\frac{\alpha}{N-j+1} = \frac{0.05}{4-4+1} = 0.05 < 0.963$$

As p_2 is greater than the threshold value we move on to test the next p-value down. We continue on with this procedure until we find a p-value which is lower than it's corresponding threshold value - for this data, this is p_3 with

$$\frac{\alpha}{N-j+1} = \frac{0.05}{4-1+1} = 0.0125 > 0.001$$

(Note j is set to 1 as p_3 is ranked last in our ordering.)

Thus we note x_3 is an 'interesting' result when tested with the Holm procedure and hence would investigate or shut down node x_3.

3.3.4 Hochberg's Procedure

Hochberg and Benjamini outlines in their 1995 paper included in the Journal of the Royal Statistical Society on pages 293 and 294 [13] a method to control the FWER.

Algorithm 4. *We begin by ordering the p-values*

$$p_{(1)} \leq p_{(2)} \leq ... \leq p_{(i)} \leq ... \leq p_{(N)}$$

Let k equal the maximum i such that

$$p_{(i)} \leq \frac{i\alpha}{N+1-i}.$$

Reject all H_{0i}, $i = 1, 2, ..., k$.

The adjusted p-value for Hochberg's procedure is

$$\tilde{p}_i = \frac{p_{(i)}(N+1-i)}{i}$$

The proof is a simple matter of rearrangement:

Proof.

$$\frac{i\alpha}{N+1-i} = p_{(i)}$$

$$i\alpha = p_{(i)}(N+1-i)$$

$$\alpha = \frac{p_{(i)}(N+1-i)}{i}$$

And hence

$$\tilde{p}_i = \frac{p_{(i)}(N+1-i)}{i}$$

□

As noted in the paper, Hochberg's procedure returns at least a bound equivalent to Bonferroni as when $i = 1$, $p_{(1)} = \frac{\alpha}{N}$.

Example 4. *Suppose we have a network of five nodes where every node sends out an average, λ, of 37.5 pieces of data per minute. The traffic is distributed according to the Poisson distribution. The vector, $\boldsymbol{x}$, records the amount of data sent out by each node during a period of two minutes.*

$$\boldsymbol{x} = \begin{pmatrix} 56 \\ 99 \\ 75 \\ 101 \\ 69 \end{pmatrix}$$

As the records are for two minutes, we double the average rate λ, to get 75 pieces of data every two minutes. Hence, we wish to test if $X \sim Pois(75)$. The following vector, $\boldsymbol{p}$, contains the p-values (rounded to 3 d.p. for presentation) corresponding to each value of $\boldsymbol{x}$,

$$p = \begin{pmatrix} 0.987 \\ 0.003 \\ 0.469 \\ 0.002 \\ 0.734 \end{pmatrix}$$

Ordering the p-values, we have

$$p_4 < p_2 < p_3 < p_5 < p_1$$

As we have five nodes in this network, $N = 5$. We set $\alpha = 0.05$. Starting at the largest p-value, p_1, we set $i = 5$ (as p_1 is fifth in our ranking of the p-values) and calculate

$$\frac{i\alpha}{N+1-i} = \frac{5 \times 0.05}{5+1-5} = 0.25 < 0.987$$

As p_1 is greater than the threshold value we move on to test the next p-value down. We continue on with this procedure until we find a p-value which is lower than it's corresponding threshold value - for this data, this is p_2 with

$$\frac{i\alpha}{N+1-i} = \frac{2 \times 0.05}{5+1-2} = 0.025 > 0.003$$

(Note i is set to 2 as p_2 is ranked second in our ordering.)

Thus we reject both x_2 and x_4 as $p_4 < p_2$ with the Hochberg procedure. We hence further investigate (or shut down) nodes x_2 and x_4.

3.4 Alternatives to FWER

3.4.1 Lehman and Romano's *k*-FWER Criteria

Lehman and Romano in their 2005 paper, Generalizations of the Familywise Error Rate [14] proposed an alternative to the FWER concept in the form of the '*k*-FWER Criteria'. In contrast to FWER where the probability of making at least one false rejection in a family of hypothesis-testing problems is considered, *k*-FWER considers making at least *k* false rejections.

Definition 19. *k-FWER: The probability of making k false rejections in a family of hypothesis-testing problems.*
i.e. k-FWER $= P($Rejecting k true $H_{0i})$ with $i \in \{1, 2, ..., N\}$.

Definition 20. *k-FWER Control Procedure: Any procedure that receives a series of p-values $p_1, p_2, ..., p_N$ and tells us if each corresponding null hy-*

potheses $H_{01}, H_{02}, ..., H_{0N}$ is rejected subject to the k-FWER being lower than a predetermined level, α.

Lehman and Romano propose a k-FWER control procedure in Theorem 2.1 of their paper [14]. The notation and style have been somewhat modified to keep it within the scope of this paper.

Theorem 1. *Associate the null hypotheses $H_{01}, H_{02}, ..., H_{0i}, ..., H_{0N}$ with p-values $p_1, p_2, ..., p_i, ..., p_N$. Suppose p_i satisfies $P(p_i \leq u) \leq u, u \in (0,1)$. The procedure that rejects any H_{0i} for which $p_i \leq \frac{k\alpha}{N}$ controls the k-FWER, so that $P(Rejecting\ k\ true\ H_{0i})$ holds.*

In order to prove the above, we first need to define Markov's Inequality. The below definition is modified from the one given on page 130 of Irwin Miller and Marylees Miller's textbook 'John E. Freund's Mathematical Statistics with Applications' [15]:

Definition 21. *Markov's Inequality:*

$$P(X \geq a) \leq \frac{E[X]}{a}$$

The following proof is adapted from Bradley Effron's textbook, 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction', page 44 [9].

Proof. Let I_0 index true hypotheses, there being N_0 true hypotheses. Let V be the hypotheses falsely declared interesting. We have by Markov's inequality that

$$P(V \geq k) \leq \frac{E[V]}{k}.$$

Hence

$$\frac{E[V]}{k} = \sum_{i \in I_0} \frac{P\left(p_i \leq \frac{k\alpha}{N}\right)}{k}.$$

As $P\left(p_i \leq \frac{k\alpha}{N}\right) = \frac{k\alpha}{N}$ we can write

$$\begin{aligned} \sum_{i \in I_0} \frac{P\left(p_i \leq \frac{k\alpha}{N}\right)}{k} &= \sum_{i \in I_0} \frac{\alpha}{N} \\ &= \frac{N_0 \alpha}{N} \\ &\leq \alpha \end{aligned}$$

□

Chapter 4

False Discovery Rates

4.1 Two Groups Model

In a review of a network's security, we take the stance for all N edges, the edge is either secure or compromised. In the language of hypothesis testing, this means a test would return either null or non-null as a result. Bradley Efron, on page 18 of his textbook, 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9] sets out this concept. These events have prior probabilities as well as z-values having densities. This is summarised below:

Prior Probability	z-Value Density
$\pi_0 = P(\text{null})$	$f_0(z) = \text{density if null}$
$\pi_1 = P(\text{non-null})$	$f_1(z) = \text{density if non-null}$

In normal cases (i.e. the network is operating normally and there hasn't been a mass wave of attacks), we would expect $\pi_0 >> \pi_1$ and moreover for $f_0(z)$ to follow a standard normal distribution under the null whilst $f_1(z)$ will follow an alternative distribution.

As touched upon in section 3.1 we want to derive a small collection of interesting results from a much larger set of possible hypotheses. We are only interested in when the alternative hypothesis has a non-trivial chance of being true.

4.2 The Bayes False Discovery Rate

Bradley Efron defines a continuous probability distribution with relation to z-values on page 18 of his 2010 book 'Large-Scale Inference: Empirical Bayes

Methods for Estimation, Testing and Prediction' [9].

Definition 22. *Continuous Probability Distribution: Let $F(Z)$ corresponding with $f(z)$ be the Continuous Probability Distribution defined as*

$$F(Z) = \int_Z f(z)dz$$

where Z is a subset of the real numbers.

In most practical settings, Z usually denotes either the region $(-\infty, z)$ or (z, ∞).
We then come to defining several key terms within the topic of false discovery rates. Suppose we have the two distributions as set out in section 4.1. Within the context of cybersecurity, the z-value for edges/nodes in a network have probability distributions $F_0(z)$ and $F_1(z)$ with densities $f_0(z)$ and $f_1(z)$. When sampling z values for the distribution related to the entire network, the distribution will of course be a 'mixture' of values derived from the secure edges/nodes (under the null distribution) and the compromised edges/nodes (under the alternate distribution). Efron [9] defines the mixture density and probability distribution as:

Definition 23. *Mixture Density:*

$$f(z) = \pi_0 f_0(z) + \pi_1 f_1(z)$$

Definition 24. *Mixture Probability Distribution:*

$$F(z) = \pi_0 F_0(z) + \pi_1 F_1(z)$$

The next definition tells us the probability of making a false discovery given z is in the region Z but we report z as non-null (i.e. the probability of saying the node/edge is compromised when it is not).

Definition 25. *Bayes False Discovery Rate for Z:*

$$Fdr(Z) \equiv P(null|z \in Z) = \frac{\pi_0 F_0(Z)}{F(Z)}$$

This is of particular interest to us - we don't want to shut down our entire network over a false discovery. This gives us a way of checking ourselves and assessing how confident we are in our claims.

4.3 Estimating Bayes FDR Empirically

In Chapter 2, Section 3 of Bradley Efron's book, 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9], he argues that we can assume the distribution of $f_0(z)$ from the null hypothesis and that in most experiments we can assume $\pi_0 = 1$. This makes sense in our cybersecurity context. We are very likely know $f_0(z)$ a priori as we can gather historic data to model our network. As was noted in section 2.1, hackers employ ingenious tricks and subtly is one of them. Only a few nodes/edges might be corrupted, making them harder to spot (if very many of them were corrupted, it would likely be obvious on the surface level of the network). Hence we can assume, like Efron, that π_0 does equal 1. This only leaves us with determining the distribution for $f_1(z)$ as knowing it before hand is unlikely - we can't realistically expect to know what a virus might do to the specific nodes/edges of a network.
Efron argues that we opt for using an empirical method to estimate this quantity. We start by defining the proportion of z_i values observed in the region Z. Let N be the total number of z_i values and $N_+(Z)$ be the number of z_i values in the region Z[1].

Definition 26. *Empirical Distribution of the N z_i-values:*

$$\bar{F}(Z) = \frac{N_+(Z)}{N}$$

[1] Note that these are not the same! N counts all z_i values while $N_+(Z)$ only counts those within the region of Z, a subset of the real line.

We can use this to to estimate Fdr(Z) by substituting this into definition 25 to give:

Definition 27. *Bayes False Discovery Rate Empirical Estimate:*

$$\overline{Fdr}(Z) = \frac{\pi_0 F_0(Z)}{\overline{F}(Z)}$$

Now split $N_+(Z)$ into $N_0(Z)$, the number of null $z_i \in Z$ and $N_1(Z)$, the number of non-null $z_i \in Z$ such that

$$N_+(Z) = N_0(Z) + N_1(Z).$$

It's not possible to observe $N_0(Z)$ so instead we need to estimate it. Thankfully, we can with the following formula.

$$e_0(Z) = E[N_0(Z)] = N\pi_0 F_0(Z)$$

The derivation of this estimate is omitted due to it being beyond the scope of this paper.
By rearranging the formula for $e_0(Z)$ we have

$$\pi_0 = \frac{e_0(Z)}{NF_0(Z)}$$

and thus

$$\begin{aligned}\overline{\text{Fdr}}(Z) &= \frac{\pi_0 F_0(Z)}{\overline{F}(Z)} \\ &= \frac{e_0(Z)}{N\overline{F}(Z)}\end{aligned}$$

Then substituting in definition 26 we have

$$\begin{aligned}\overline{\text{Fdr}}(Z) &= \frac{e_0(Z)}{N(N_+(Z)/N)} \\ &= \frac{e_0(Z)}{N_+(Z)}\end{aligned}$$

Example 5. *Suppose we have a network of 7834 edges and we are looking for 'interesting' activity. We want to calculate an estimate for the false discovery rate if we set the region* Z *to be* $(2.5, \infty)$. *We find 201* z_i *exceed* 2.5, *that is, fall in the region* $(2.5, \infty)$. *Assuming* $pi_0 = 1$, *we can calculate the expected number of null* z_i *in the region* Z.

$$\begin{aligned} e_0(z) &= N\pi_0 F_0(Z) \\ &= 7834 \times 1 \times (1 - \Phi(2.5)) \\ &= 48.6465 \ (4 \ d.p.) \end{aligned}$$

The false discovery rate is hence

$$\begin{aligned} \overline{Fdr}(Z) &= \frac{e_0(Z)}{N_+(Z)} \\ &= \frac{e_0(z)}{201} \\ &= 0.2420 \ (4 \ d.p.) \end{aligned}$$

That is, there's roughly a 24% *chance of making a false discovery.*

Note that `R` *code accompanying this calculation can be found in appendix A.*

4.4 Benjamini-Hochberg's FDR Control Algorithm

In any experiment involving multiple hypotheses, we will have N null hypotheses:

$$H_{01}, H_{02}, ..., H_{0N}$$

We test these hypotheses using a data set X. A decision rule D then classifies these hypotheses "null" or "non-null". Table 4.1 summarises the various outcomes.

		Decision		
		Null	Non-null	Total
Actual	Null	U	V	N_0
	Non-Null	T	S	N_1
	Total	$N - R$	R	N

Table 4.1: D declared $N-R$ hypotheses null, of which, U were actually null and T were not. In contrast, D declared R tests non-null, of which V were null and S were not. The notation used is a merger between what's found in Benjamini's and Hochberg's original paper on page 291 of the 'Journal of the Royal Statistical Society' [13] and Bradley Effron's notation in his book 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9].

Note that the probability of $V > 0$ is the family wise error rate and further note that in the case $N = 1$ we have the traditional hypothesis situation with $P(V = 1|N_0 = 1) = \alpha$, α the size of the test and $P(S = 1|N_1 = 1) = \beta$, β the power. Indeed, it can be shown that

$$E\left[\frac{V}{N_0}\right] = \bar{\alpha}$$

and

$$E\left[\frac{S}{N_1}\right] = \bar{\beta}$$

for multiple hypothesis testing.
In the classic sense originally envisaged by Fisher, a distribution for the null is assumed which allows us to look at α and calculate the size of the test. With Neyman-Pearson Theory, a distribution is specified for the non-null case, allowing us to examine power, β. It's when we enter large scale testing that we can 'work vertically'.
The Benjamini-Hochberg (BH) algorithm, as laid out by Efron on pages 48 and 49 of his book 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9]

Algorithm 5. *Assume under a decision rule, D, each null Hypothesis, H_{0i}, has a uniformly distributed p-value, p_i, produced by D if H_{0i} is correct. i.e. under D, if H_{0i} is true,*

$$H_{0i} : p_i \sim U(0,1)$$

We order our p-values as follows

$$p_1 \leq p_2 \leq ... \leq p_i \leq ... \leq p_N$$

For a fixed $q \in (0,1)$*, let* j *be the largest index such that*

$$p_i \leq \frac{iq}{N}$$

reject all H_{0i} *corresponding to* p_i *with* $i \leq j$*. Accept* H_{0i} *otherwise.*

Consider the following example:

Example 6. *Suppose we have a network of servers (nodes) transmitting sensitive financial data where every server transfers six hundred pieces of data per hour, with a standard deviation of thirty pieces per hour. The traffic is distributed according to the Normal distribution, i.e.* $X \sim N(600, 30^2)$*. Breaking the network down into component edges, we obtain the following vector,* x*, of traffic rates from a monitoring period:*

$$x = \begin{pmatrix} 672 \\ 663 \\ 680 \\ 669 \\ 644 \\ 612 \end{pmatrix}$$

We suspect the network may be compromised with a virus affecting certain servers, causing them to send out corrupted information. A compromised server causes a loss of all revenue (around \$700 every hour) for as long as it's allowed to still be 'live'. However, shutting down a server causes the revenue generated from it (around \$700 per hour) to be lost along with the cost of hiring a professional to investigate the situation (\$150 per hour) for three hours. Put simply, it's best to not to unnecessarily shut down a server over a false alarm. We use the Benjamini-Hochberg to investigate the network.

The following vector, p, contains the p-values (rounded to 3 d.p. for presentation) corresponding to each value of x,

$$p = \begin{pmatrix} 0.008 \\ 0.018 \\ 0.004 \\ 0.011 \\ 0.071 \\ 0.345 \end{pmatrix}$$

Ordering the p-values, we have

$$p_3 < p_1 < p_4 < p_2 < p_5 < p_6$$

As we have six nodes (servers) in this network, $N = 6$. We set $q = 0.1$. Starting at the largest p-value, p_6, we set $i = 6$ (as p_6 is sixth in our ranking of the p-values) and calculate

$$\frac{iq}{N} = \frac{6 \times 0.1}{6} = 0.1 < 0.345$$

As p_6 is greater than the threshold value we move on to test the next p-value down, p_5. We set $i = 5$ (as p_5 is fifth in our ranking of the p-values) and calculate

$$\frac{iq}{N} = \frac{5 \times 0.1}{6} > 0.071$$

p_5 is less than the threshold value and as $p_3 < p_1 < p_4 < p_2 < p_5$ we have x_1, x_2, x_3, x_4 and x_5 detected as interesting when tested with the Benjamini-Hochberg Algorithm. We hence shut down servers x_1, x_2, x_3, x_4 and x_5 to investigate them.

Efron, on page 49 of ‘Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction’ [9], highlights an interesting theorem associated with the Benjamini-Hochberg algorithm. We first start by defining the False Discovery Proportion for D.

Definition 28. *False Discovery Proportion for D: The proportion of rejected cases that are actually null.*

$$Fdp_D = \frac{\alpha_D}{R_D}$$

where α_D is the number of null hypotheses declared non-null under D and R_D is the total number of 'discoveries' under D.

Theorem 2. *If the p-values corresponding to the correct null hypothesis are independent of each other, then the rule BH(q) based on the BH algorithm controls the expected false discovery proportion at q,*

$$E[Fdp_{BH(q)}] = \pi_0 q \leq q \textit{ where } \pi_0 = \frac{N_0}{N}$$

The proof of this theorem, while interesting, is beyond the scope of this paper. The proof can however be found in Bradley Efron's textbook 'Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction' [9] on pages 51 and 52.

Theorem 2 essentially allows us to put a cap on the proportion of false discoveries we make when using the BH algorithm. This is attractive from a cybersecurity stand point. As was highlighted in example 6, shutting down a node/edge of a network can cause a loss of revenue for a company - we do not want to go on several wild-goose chases.

Efron [9] raises several valid concerns with the Benjamini-Hochberg procedure however. We assume the p-values are independently distributed in order to access the attractive statement of theorem 2 - this isn't necessarily realistic in cybersecurity. Edges/nodes may very well depend on one another, even when working correctly (i.e. the null hypothesis). Further more, we are controlling an error rate *expectation* rather than a *probability.* Is it genuinely good enough to only control the expectation? Efron also points out that we calculate the p-values for the BH procedure based on an assumed null distribution. This assumption can break down when N grows large (as it frequently does when we have massive networks with thousands of nodes/edges!) Finally, there is the question of what level do we set q at? Efron suggests $q = 0.1$ based on the existing literature but concedes the consensus is not yet concrete. A nagging worry remains that we have fell into the trap outlined in section 2.5 - is q just another “arbitrary” level to argue over and potentially abuse? These questions can’t be answered within the scope of this paper as their answers are lengthy and complex, but would provide the basis for further research.

Chapter 5

Conclusion

This paper has looked at the issue of family wise error rates (FWER) and false discovery rates (FDR) and how to best combat them. We have covered a wide variety of family wise error rate procedures; namely the Bonferroni bound as well as the Šidák, Holm and Hochberg procedures. We have also extensively looked at the Benjamini-Hochberg (BH) algorithm. The examples given for these procedures have used very small data sets to keep things simple. Real-life scenarios almost certainly involve thousands of inputs for the data. Appendix B contains an `R` script implementing the BH algorithm, Bonferroni bound, Hochberg procedure and Holm procedure which could be easily modified to take on a vector of thousands of inputs if needed.

It is worth noting this research is applicable far beyond it's motivating example of cybersecurity. The concepts and procedures have applications in biology, manufacturing, data science and many other fields. How best to implement the methods to said fields could be the subject of many strands of research.

The extensive list of FWER procedures compared with the single procedure controlling FDR[1] given in this paper begs the question of further exploring FDR theory and potentially coming up with a new method that requires more realistic assumptions than those of the BH algorithm.

[1] There does exist other methods such as the Benjamini–Yekutieli procedure which has been omitted from this paper.

Appendix A

R Code Accompanying Example 5

This code accompanies example 5 by showing how to implement the calculations in R.

```
#Set variables
N=7834
Np=201
pi0=1

# Calculate expected number of null z_i in the region Z
e0 = N*pi0*(1-pnorm(2.5))

# Calculate expected FDR
FDR = e0/Np

# Print outputs
e0
FDR
```

Appendix B

R Code to use Several Procedures as Tests

This code is modified from Sal Mangiafico's response to a question on how to compute p-values for the Bonferroni bound and Benjamini-Hochberg's algorithm [16]. I have included the Holm and Hochberg procedures as laid out in sections 3.3.3 and 3.3.4 respectively. In this code, I assume we are testing some very bland sample data, namely the integers between 14 and 21 inclusive. We pretend these were drawn from a random variable X. We test to see if any of these samples contradict the null hypothesis $X \sim Pois(10)$. In reality, we would have thousands of samples and may need to change the distribution for the null hypothesis.

```
#Set values for lambda
lambda <- 10

#Example input to be switched out for real test data
observed.values <- c(14,15,16,17,18,20,21)

#Generate a list of p-values
p.values <- ppois(observed.values, lambda,
lower.tail = F)

#Calculate adjusted p-values
p.BH = p.adjust(p.values, method="BH")
p.B = p.adjust(p.values, method="bonferroni")
p.Hoch = p.adjust(p.values, method="hochberg")
p.Holm = p.adjust(p.values, method="holm")
```

```
#Make things pretty
p.values = round(p.values, 3)
p.BH = round(p.BH, 3)
p.B = round(p.B, 3)
p.Hoch = round(p.Hoch, 3)
p.Holm = round(p.Holm, 3)

#Test
S.BH = p.BH < 0.05
S.B = p.B < 0.05
S.Hoch = p.Hoch < 0.05
S.Holm = p.Holm < 0.05

#Create data frame to store results
Data = Data = data.frame(observed.values, p.values,
p.BH, S.BH, p.B, S.B, p.Hoch, S.Hoch, p.Holm, S.Holm)

#Print data. If TRUE, reject the null hypothesis
#associated with the observed value
Data
```

Bibliography

[1] Nickolai Zeldovich (2014) 1. Introduction, Threat Models. MIT OpenCourseWare, MIT 6.858 Computer Systems Security, Fall 2014. `https://www.youtube.com/watch?v=GqmQg-cszw4`. Access Date: 10.11.19

[2] James Mickens (2014) 2. Control Hijacking Attacks. MIT OpenCourseWare, MIT 6.858 Computer Systems Security, Fall 2014. `https://www.youtube.com/watch?v=r4KjHEgg9Wg` . Access Date: 11.11.19

[3] Nick Heard (2018), Data Science in Cyber-Security and Related Statistical Challenges. Microsoft Research. `https://www.microsoft.com/en-us/research/video/data-science-in-cyber-security-and-related-statistical-challenges/` . Access Date: 5.11.19

[4] Gustav W. Delius (2017), Introduction to Probability & Statistics Lecture Notes. The University of York, MAT00004C, page 33

[5] Ben Powell (2019) Probability and Statistics II: Statistical Inference 2 and Linear Models. The University of York, MAT00035I

[6] Steffen Lauritzen (2004), Statistical Inference. University of Oxford, BS2. `http://www.stats.ox.ac.uk/~steffen/teaching/bs2siMT04/si11c.pdf`. Access Date: 13.6.2020

[7] Ronald L. Wasserstein & Nicole A. Lazar (2016) The ASA Statement on p-Values: Context, Process, and Purpose. The American Statistician, 70:2, 129-133, DOI: 10.1080/00031305.2016.1154108

[8] Bradley Efron (2007) Size, Power and False Discovery Rates. The Annals of Statistics. 2007, Vol. 35, No. 4, 1351-1377. DOI: 10.1214/009053606000001460. Institute of Mathematical Statistics, 2007

[9] Bradley Efron (2010) Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction. Cambridge University Press. ISBN 978-0-521-19249-1.

[10] Juliet Popper Shaffer (1995) Multiple Hypothesis Testing. Annu. Rev. Psychol 1995 46:561-84

[11] Ryan Ta (2019) MATH 149A. The University of California, Riverside, page 1. `http://math.ucr.edu/~ryanta/teaching/math149a_fall2019/boole%27s_inequality`. Access Date: 10.6.2020

[12] Ben Powell (2020) Probability and Statistics II: Statistical Inference 2 and Linear Models. The University of York, MAT00035I, slides 156-157

[13] Yoav Benjamini, Yosef Hochberg (1995) Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, Journal of the Royal Statistical Society, Series B (Methodological), Vol. 57, No. 1 (1995), 290-294

[14] E. L. Lehmann and Joseph P. Romano (2005) Generalizations of the Familywise Error Rate, The Annals of Statistics, 2005, Vol. 33, No. 3, 1138–1154, DOI:10.1214/009053605000000084 Institute of Mathematical Statistics, 2005

[15] Irwin Miller, Marylees Miller (2014), John E. Freund's Mathematical Statistics with Applications (International Edition), Eight Edition. Pearson. ISBN-10: 0-321-90440-0. ISBN-13: 978-0-321-90440-9.

[16] Sal Mangiafico (https://stats.stackexchange.com/users/166526/sal-mangiafico), Bonferroni bound and FDR: compute p-values, URL (version: 2017-12-19): `https://stats.stackexchange.com/q/319583`. Access Date: 21.5.2020