Estimating dose and time of exposure from a protein-based radiation biomarker

YILUN CAI

Estimating dose and time of exposure from a protein-based radiation biomarker

Yilun Cai

Master by Research (MRes) -Mathematical Sciences

Department of Mathematical Sciences

Durham University

2024

Abstract

In order to analyze the potential damage to the human body caused by exposure to ionizing radiation, one needs to have an estimation of the dose of radiation received by the individual. We present here a new method that, unlike the approaches that produce the estimation with data collected at a predetermined time after exposure, allows us to estimate the dose at any time within a reasonable time interval after exposure, as well as determine the time of exposure if needed. Namely, we take existing calibration curves and generalize them using the decay mechanism of γ-H2AX foci to build a model that describes the functional relationship between the count of γ-H2AX foci in exposed blood cells and the time and dose of exposure. This model is illustrated using both real and simulated data.

Contents

Chapter 1

Acknowledgement

I would like to give special thanks to my supervisor, Professor Jochen Einbeck, for his guidance and support during this project.

Chapter 2

Introduction

In modern society, individuals are at risk of potentially harmful ionizing radiation exposure, such as occupational radiation workers (nuclear workers, medical staff, radiographers etc.) and also patients undergoing diagnostic or therapeutic radiation treatments to detect or treat cancers and other related health conditions (orthopaedics, cardiology etc.). Therefore, following a potential over-exposure to ionizing radiation in a mass-casualty radiological accident or incident, one needs a tool such as biodosimetry (biological dosimetry, the use of physiological, chemical or biological markers of exposure of human tissues to ionizing radiation for the purpose of reconstructing doses to individuals or populations) to estimate the dose of the exposure to inform on medical intervention and/or triage of patients for treatment.

2.1 The Dicentric Chromosome Assay

Ionizing radiation exposure causes DNA double strand breaks (DSBs). The repairing of chromosomes can result in several types of chromosome aberrations during DNA replication. Specifically, dicentric chromosomes are the result of the joining of two chromosome ends each containing a centromere. Upon mitosis, this often leads to an anaphase bridge and breakage[24]. Dicentric chromosomes very rarely occur spontaneously (around 1 dicentric chromosome in 1000 cells), are highly characteristic of ionising radiation, and are sensitive to acute, recent exposure to ionising radiation[1], which are all desirable traits for a biomarker in biological dosimetry[2]. Currently, the dicentric chromosome assay (DCA) is well established and is regarded as the ‘gold standard’ method for accurate dose estimation[10][13][4][9] which is based on the counts of dicentric chromosomes. This method benefits from being robust to inter-individual[3] and inter-laboratory variations[26], but there are clear drawbacks such as being too costly[18], and too slow for emergencies since from the moment of exposure, lymphocytes need to be cultured for at least two days until chromosomes reach the metaphase stage of mitosis[27] before the dose estimation.

2.2 The γ-H2AX assay

A chromatin is a combination of DNA and proteins (mainly histones) in the interphase nucleus, before it condenses into mitotic chromosomes. A nucleosome is the fundamental repeating unit of chromatin, which consists of about 147 base-pairs of DNA wrapped around a histone: octamer. The octamer is built from two copies of each of four different families of histone proteins: H2A, H2B, H3, and H4. Specificallly, the H2A family has subfamilies: H2A1/2, H2AZ, and H2AX[25]. The H2AX histone responses to DNA damage and coordinates its repair[15]. Upon occurrence of DSB, protein kinases (ATM, ATR, DNA-PKcs) rapidly phosphorylate the C-terminal Ser139 of H2AX, producing the γ-H2AX, which spreads over megabase-scale regions flanking the break, creating a platform to recruit mediator proteins (MDC1), ubiquitin ligases (RNF8/RNF168), and downstream effectors (53BP1, BRCA1), building a visible multiprotein hub, which is called a γ-H2AX focus. In practice, a blood sample (no more than 0.1ml of blood that can be collected through a finger prick sample) is collected[21][17] from individuals. Antibodies that specifically bind to γ-H2AX (and not to unmodified H2AX) are then applied to each of a number of cells in the sample, before applying secondary antibodies carrying a fluorescent dye which binds the first type of antibody, so every γ-H2AX focus lights up as a bright dot (figure 2.2) under a fluorescence microscope (figure 2.1), where each fluorescent dot (roughly) corresponds to one DSB (or a small cluster of breaks) in a nucleus. These dots are then counted using manual, automated or semi-automated scoring techniques[27][22][6]. This practice is called immunofluorescence microscopy[12], which can be done minutes after ionizing radiation exposure[28], where the γ-H2AX foci reach the highest amount at around $1h$ after exposure then slowly decay until almost disappearing at around $96h$ after exposure[7][14].

A relatively new method that is able to produce results faster than the DCA, the γ-H2AX assay uses the γ-H2AX foci as the biomarker. Common approaches utilizing this biomarker

Figure 2.1: immunofluorescence microscopy (source: bfs.de)

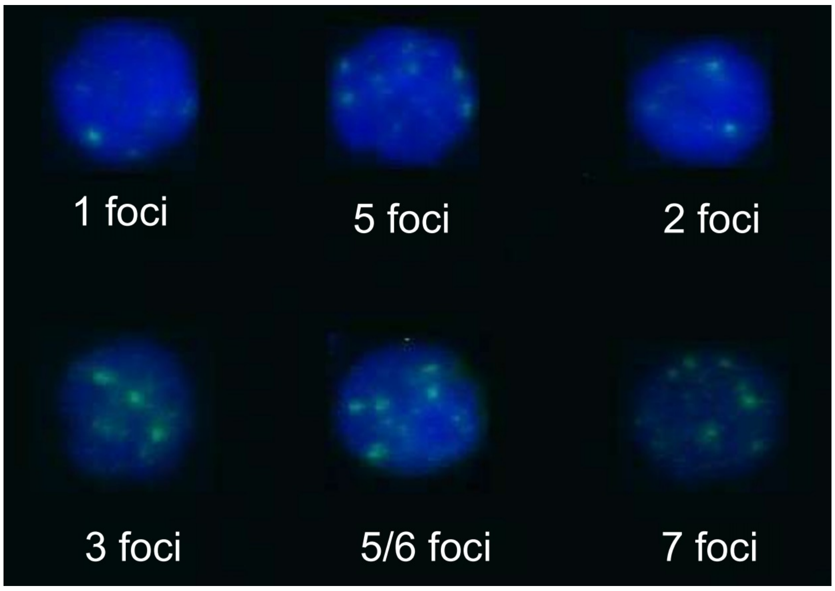


Figure 2.2: γ-H2AX foci (source: UKHSA, Radiation Effects Department)

involve exposing blood samples collected from individuals and exposing them to design doses of ionizing radiation (e.g. 0.5/1/2/4 Gy), then counting the number of γ-H2AX foci in a set number of cells in each sample at predetermined times after exposure such as 1, 2, 4, 24 hours[12][5][27]. We will hence refer to the mean count (which corresponds to the average or the expectation depending on the context) of γ-H2AX foci in a sample at a time after exposure as 'Yield', and denote it as Y. Linear calibration curves are fitted that describe the functional relationship between Yield and design doses at each time after exposure.

These approaches, while having multiple advantages over DCA such as allowing the dose estimation to take place much sooner after exposure, are however, still limited by their time dependence. For instance, if a potentially irradiated individual were to come in to the lab at say 6 hours after exposure, the lab would have no choice but to ask this individual to wait for 18 hours to take the sample and have a dose estimation. Further, if this individual is unsure about when the time of the possible exposure, then there would be no way to estimate the dose. To address this issue, we present here a new method that allows us to estimate ionizing radiation dose at any time within a reasonable time interval after exposure, as well as determine the time of exposure if needed. To do this, we take existing calibration curves and generalize them using the decay mechanism of γ-H2AX foci to build a model that describes the functional relationship between Yield, dose, and time since exposure. To apply this model, we simply take γ-H2AX foci count measurements twice, i.e. at different time points, to infer dose and time since exposure.

Chapter 3

The Basic Model

The idea of this model in its essence is to generalize calibration curves so as to remove their time dependence, and we use the decay mechanism of γ-H2AX foci to achieve this.

3.1 The Decay Mechanism

We denote D as the dose of ionizing radiation received in Gy and denote t as the time after ionizing radiation exposure in hours. Recall that we mentioned in the introduction that the number of γ-H2AX foci decreases between $1h$ and $96h$ after exposure[7][14]. We refer to this as γ-H2AX foci decay. Denote Y_t as the Yield at time t hours after ionizing radiation exposure. Previous study[14] shows that the Yield follows

$$Y_t = D(11.92e^{-0.3495t} + 3.552e^{-0.01843t}) \tag{3.1}$$

between $1h$ and $96h$ after exposure. We use this as the decay mechanism of γ-H2AX foci in the human body since the character of this mechanism, taking the biexponential form, is a physical property and is presumably intra- and inter-lab as well as intra- and inter-individual independent. Therefore, we extract the form of the decay mechanism to get

$$Y_t = D(Ae^{ut} + Be^{vt})$$

that we refer to as the basic decay mechanism model. Specifically, A, B are positive parameters and u, v are negative parameters. All four of these parameters will be determined later.

3.2 Fitting Calibration Curves

Recall that we mentioned in the Introduction that past literature describes the functional relationship between Yield and dose at each time after exposure by fitting a linear calibration curve[12][5][27]. We will illustrate here how the calibration curves are fitted in practice by looking at the case from [12] as an example.

We define, for fixed t,

$$Y_t = a_t D + b_t,$$

where a_t and b_t are parameters to be determined, as the basic calibration curve model.

Recall from the Introduction that we defined the 'Yield': Y, as the average count of γ-H2AX foci in a number of cells from a sample at a time after exposure. Specifically, we will denote the number of cells being used for foci counting in each sample as n and denote the number of samples we consider as N. For example, in the case of [12], at least $N = 16$ samples where used for each dose and the foci counts in $n = 500$ 'cells' were recorded in each sample.

Here it makes sense to

The generalized linear model function `glm` in R was employed to generate the calibration curves, namely, to determine the values of the parameters a_t and b_t. Here we used Poisson GLM with identity link, so we are only fitting a simple linear WLS (refer to section 4.2.2 for details). However, instead of a model of the average foci count of n cells in a sample, namely, Y_t, the sum of the n counts in a sample, namely nY_t was modelled. Therefore, we have

$$nY_t = a_t nD + nb_t,$$

which is a 'Quasi-Poisson' generalized linear model with identity link and covariates: nD and n[19]. The 'Quasi-Poisson' model refers to the response variables being modelled by an overdispersed version of the Poisson model. We will later explain this further in section 4.2.1.

An argument could be made that instead of modelling the sum of foci counts nY_t, we could simply model the Yield Y_t. However, in this case the model would no longer be a model of counts and become a linear model, which leads to the response variables being assumed to have Normal distributions instead of Poisson distributions or overdispersed versions of it. The resulting values of a_t and b_t would likely have a minute change that is overall inconsequential. However, the uncertainty estimations would be heavily effected. Therefore, it is inadvisable to model Y_t instead of nY_t.

3.3 The Basic Model

Recall that in the last section, we defined

$$Y_t = a_t D + b_t,$$

where a_t and b_t are parameters to be determined, as the basic calibration curve model. Here, a_t relates heavily to the decay mechanism as it is multiplied by D. However, b_t should have very little to do with the decay mechanism since ideally we should be seeing a zero Yield when $D = 0$, and so b_t likely comes from other factors in play in the environment such as laboratory conditions, individual difference, etc. (i.e. the background yield of foci when no exposure has taken place). Therefore, we assume that a_t changes according to the basic decay mechanism model while b_t simply changes linearly with t as we do not wish to do much with it, and so the calibration curves takes the form of

$$Y_t = at + b + D(Ae^{ut} + Be^{vt}) \tag{3.2}$$

We refer to this as the basic model.

At the moment, for simplicity of argument, we assume that we know the values of parameters a, b, A, B, u, v. We will determine these values later.

If an individual in question is aware of the time of exposure, we take one sample. In this case the only unknown in the model would be D, which we can easily solve for:

$$D = \frac{Y_t - at - b}{Ae^{ut} + Be^{vt}}$$

In the event of the individual in question not knowing the time of exposure, we take two samples at two different times. We denote t_1 and t_2 as the time in hours after ionizing radiation exposure when we take the first and second sample, and define $\Delta = t_2 - t_1$. In this case, we would not know t_1 or t_2, but we would know Δ. We have:

$$Y_{t_1} = at_1 + b + D(Ae^{ut_1} + Be^{vt_1})$$

and

$$Y_{t_1+\Delta} = a(t_1 + \Delta) + b + D(Ae^{u(t_1+\Delta)} + Be^{v(t_1+\Delta)})$$

where we have two unknowns, namely t_1 and D and two equations. However, this time we can not as easily find an explicit expression of t_1 and D. Therefore, we employ numerical methods to acquire solutions that are "sufficiently good". Specifically, we use the `Nsolve` function of Mathematica. For our specific problem, NSolve computes a numerical Gröbner basis using an efficient monomial ordering, then uses eigensystem methods to extract numerical roots.

To acquire the values of the parameters a, b, A, B, u, v, the parameters u, v can reasonably be assumed to be determined by the nature of the physical decay mechanism due to their being part of the power of e and are directly multiplied by t so we fix them to $u = -0.3495$ $v = -0.01843$ from (3.1) and A, B are related to the absolute magnitude of foci observed, and hence are likely to be lab-specific (e.g. depending on the filter settings of the foci detection software). We solve for parameters a, b, A, B by equating the model with the relative terms of two calibration curves. These can be any two calibration curves as long as they are acquired with the same laboratory conditions.

Specifically, supposing that we have calibration curves at times t_{1c} and t_{2c}:

$$Y_{t_{1c}} = a_{t_{1c}} + b_{t_{1c}} D$$

and

$$Y_{t_{2c}} = a_{t_{2c}} + b_{t_{2c}} D,$$

where we equate terms by

$$at_1 + b = a_{t_{1c}}$$

$$a(t_1 + \Delta) + b = a_{t_{2c}}$$

$$Ae^{-0.3495t_1} + Be^{-0.01843t_1} = b_{t_1c}$$

$$Ae^{-0.3495(t_1+\Delta)} + Be^{-0.01843(t_1+\Delta)} = b_{t_{2c}}$$

to acquire the values of the parameters.

To illustrate with an example, we look at [12]. Here, we have a pair of calibration curves fitted to a data set that contains at least 16 samples with 500 cells counted in each sample for 1 hour after exposure:

$$Y_1 = 0.131 + 12.559D$$

and 24 hours after exposure:

$$Y_{24} = 0.179 + 1.937D.$$

Equate the relative terms and we have

$$at_1 + b = 0.131$$

$$a(t_1 + \Delta) + b = 0.179$$

$$Ae^{-0.3495t_1} + Be^{-0.01843t_1} = 12.559$$

$$Ae^{-0.3495(t_1+\Delta)} + Be^{-0.01843(t_1+\Delta)} = 1.937$$

where $t_1 = 1$, $t_2 = 24$, $\Delta = 23$.
Solve for a, b, A, B to get $a = 0.002087$, $b = 0.129$, $A = 13.6222$, $B = 3.0095$, and we have

$$Y_{t_1} = 0.129 + 0.002087t_1 + D(13.6222e^{-0.3495t_1} + 3.0095e^{-0.01843t_1})$$

$$Y_{t_1+\Delta} = 0.129 + 0.002087(t_1 + \Delta) + D(13.6222e^{-0.3495(t_1+\Delta)} + 3.0095e^{-0.01843(t_1+\Delta)})$$

as the model, and we use `Nsolve` to acquire the estimations for D, t_1.

3.4 Calibration Curves in Abundance

In the previous section, there were only two calibration curves available to us, and so we used them to infer a, b, A, B. However, if we had three or four calibration curves, we would in theory be able to infer u or/and v using `Nsolve` instead of taking them as given. However, attempts at this were unsuccessful. As long as there are at least three total variables including at least one of u or v, Mathematica will run until RAM fills up or crash in the process.

Chapter 4

Testing with Simulated Data

In this chapter we will test our model with simulated data. Specifically, we setup a simulation model where we generate N samples in each test with each sample containing $n = 500$ 'cells' and therefore 500 simulated foci counts (unless when stated otherwise). We need a large enough N to sufficiently reduce the effect of randomness, so we use $N = 100$ in all testings in this chapter.

4.1 The Suitability of `Nsolve`

First, we would like to see if `Nsolve` indeed works in our scenario and we do so using Poisson distributed simulation data.

4.1.1 The Poisson Simulation Model

Define design Yield as the Yield we would like to simulate and denote $Y_{D,t}$ as the design Yield at time t hours after "exposure" to design dose D Gy of ionizing radiation. We will use the example from the last chapter as our framework, therefore we substitute $D = 1$ and $D = 4$ into the calibration curves from the example, namely

$$Y_1 = 0.131 + 12.559D$$

and

$$Y_{24} = 0.179 + 1.937D$$

to acquire the design Yields: $Y_{1,1}, Y_{1,24}, Y_{4,1}, Y_{4,24}$ for design doses $D = 1$ and $D = 4$ at time $t_1 = 1$ and $t_2 = 24$. We then generate $n = 500$ count data for each design Yield: $C^{(1)}_{D,t}, C^{(2)}_{D,t}, .., C^{(n)}_{D,t}$, with

$$C^{(1)}_{D,t}, C^{(2)}_{D,t}, .., C^{(n)}_{D,t} \overset{\text{iid}}{\sim} Poisson(Y_{D,t})$$

with each combination of D and t. And so we have

$$\sum_{k=1}^{n} C^{(k)}_{D,t} \overset{\text{iid}}{\sim} Poisson(nY_{D,t}),$$

$k = 1, 2, ..., n$, and the Mean of these $n = 500$ data: $\overline{C}^{(k)}_{D,t}$ would effectively be Y_1 or Y_{24}, and

$$\overline{C}^{(k)}_{D,t} \sim \frac{Poisson(nY_{D,t})}{n}$$

$$E[\overline{C}^{(k)}_{D,t}] = \frac{nY_{D,t}}{n} = Y_{D,t},$$

$k = 1, 2, ..., n$. To acquire the estimations, we substitute into the same model from the example in the last chapter for each design dose, namely

$$\overline{C}^{(k)}_{D,1} = 0.129 + 0.002087t_1 + D(13.6222e^{-0.3495t_1} + 3.0095e^{-0.01843t_1})$$
$$\overline{C}^{(k)}_{D,24} = 0.129 + 0.002087(t_1 + 23) + D(13.6222e^{-0.3495(t_1+23)} + 3.0095e^{-0.01843(t_1+23)}),$$

$k = 1, 2, ..., n$. Solving the equations yields an estimated D: $\hat{D}$ and an estimated t_1: $\hat{t}$ for each design dose $D = 1$ and $D = 4$. We repeat this $N = 100$ times to simulate 100 samples and acquire $\hat{D}^{(1),(2),...,(N)}$ and $\hat{t}^{(1),(2),...,(N)}$ for each $D = 1$ and $D = 4$.

4.1.2 Setup

We display the simulation results with Box Whisker Plots. We will also display the Bias of $\hat{D}$ and $\hat{t}$ denoted by $Bias_{\hat{D}}$ and $Bias_{\hat{t}}$ that we define as

$$Bias_{\hat{D}} = \frac{\sum\limits_{k=1}^{N} \hat{D}^{(k)}}{N} - D$$

$$Bias_{\hat{t}} = \frac{\sum_{k=1}^{N} \hat{t}^{(k)}}{N} - t_1,$$

$k = 1, 2, ..., N$, as well as Variance of $\hat{D}$ and $\hat{t}$ denoted by $Var_{\hat{D}}$ and $Var_{\hat{t}}$ that we define as

$$Var_{\hat{D}} = \frac{\sum_{k=1}^{N} \left(\hat{D}^{(k)} - \frac{\sum_{k=1}^{N} \hat{D}^{(k)}}{N} \right)^2}{N-1}$$

$$Var_{\hat{t}} = \frac{\sum_{k=1}^{N} \left(\hat{t}^{(k)} - \frac{\sum_{k=1}^{N} \hat{t}^{(k)}}{N} \right)^2}{N-1},$$

$k = 1, 2, ..., N$.

To further illustrate the results, we define the Absolute Bias as some what of an combination of Bias and Variance of $\hat{D}$ and $\hat{t}$ denoted $AbsBias_{\hat{D}}$ and $AbsBias_{\hat{t}}$ that we define as

$$AbsBias_{\hat{D}} = \frac{\sum_{k=1}^{N} \left| \hat{D}^{(k)} - D \right|}{N}$$

$$AbsBias_{\hat{t}} = \frac{\sum_{k=1}^{N} \left| \hat{t}^{(k)} - t_1 \right|}{N},$$

$k = 1, 2, ..., N$.

4.1.3 Results

We display the results with Box Whisker Plots in figures 4.1 and 4.2, as well as the relative statistics in table 4.1. It seems from the results that `Nsolve` is indeed able to generate accurate and precise estimations.

4.2 Performance under Overdispersion

We see that the model performs well with Poisson data, but the Poisson Model is not well suited for real world data. In fact the Variance of the data can be as high as 50 to 60 times the Mean[12]. Therefore, we employ the Negative Binomial Distribution to simulate an overdispersed Poisson data set following a 'Quasi-Poisson' Distribution.

Table 4.1: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Poisson distributed simulation data

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$D = 1$, $t_1 = 1$	-0.000644197	0.000959545	0.0229637
$D = 4$, $t_1 = 1$	0.00537652	0.00452254	0.0522639

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$D = 1$, $t_1 = 1$	-0.00186132	0.0150187	0.092421
$D = 4$, $t_1 = 1$	0.00767098	0.00407422	0.0507677

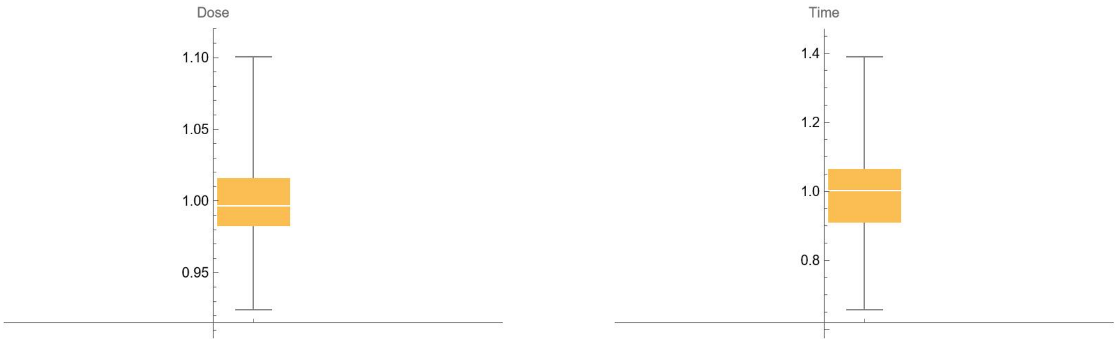


Figure 4.1: $\hat{D}$ and $\hat{t}$ from Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

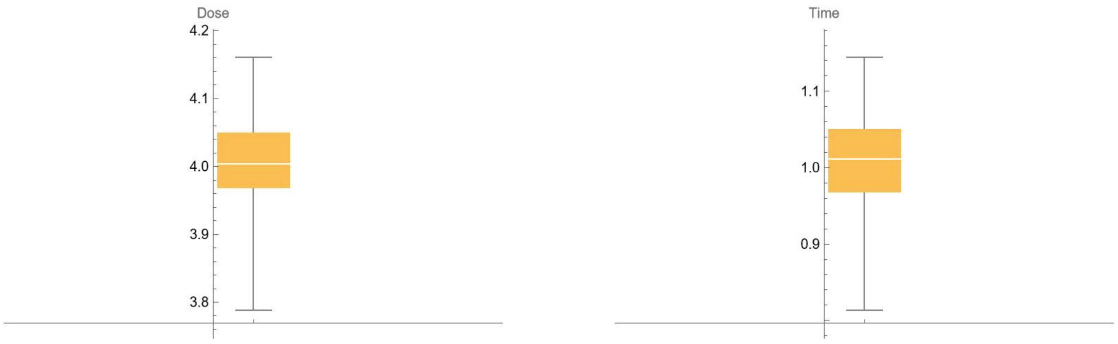


Figure 4.2: $\hat{D}$ and $\hat{t}$ from Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

4.2.1 The Quasi-Poisson Simulation Model

Recall from section 4.1.1 that we had

$$C_{D,t}^{(1)}, C_{D,t}^{(2)}, .., C_{D,t}^{(n)} \overset{\text{iid}}{\sim} Poisson(Y_{D,t})$$

Here we will instead define

$$C_{D,t}^{*(1)}, C_{D,t}^{*(2)}, .., C_{D,t}^{*(n)} \overset{\text{iid}}{\sim} QuasiPoisson(Y_{D,t}, \phi)$$

where $Y_{D,t}$ is still the mean and we define ϕ as the dispersion parameter following

$$\phi = \frac{Var(C_{D,t}^{*(k)})}{Y_{D,t}},$$

$k = 1, 2, ..., n$. Here, $Var(C_{D,t}^{*(k)})$ denotes the Variance of $C_{D,t}^{*(k)}$ and we still have

$$\overline{C^*}_{D,t}^{(k)} \sim \frac{QuasiPoisson(nY_{D,t})}{n}$$

$$E[\overline{C^*}_{D,t}^{(k)}] = \frac{nY_{D,t}}{n} = Y_{D,t}$$

We set $\phi = 50$.

The Quasi-Poisson distribution is not a real distribution, we therefore employ the Negative Binomial distribution to simulate data. Therefore, we have

$$C_{D,t}^{*(1)}, C_{D,t}^{*(2)}, .., C_{D,t}^{*(n)} \overset{\text{iid}}{\sim} NegativeBinomial(r_{D,t}, p_{D,t}),$$

where $NegativeBinomial(r_{D,t}, p_{D,t})$ describes the distribution of the number of failures in a sequence of trials with success probability $p_{D,t}$ before $r_{D,t}$ successes occur. We have

$$Y_{D,t} = \frac{r_{D,t}(1 - p_{D,t})}{p_{D,t}}$$

and

$$\phi Y_{D,t} = Var(C_{D,t}^{*(k)}) = \frac{r_{D,t}(1 - p_{D,t})}{p_{D,t}^2},$$

$$p_{D,t} = \frac{1}{\phi}$$

and

$$r_{D,t} = \frac{p_{D,t}}{1 - p_{D,t}} Y_{D,t}.$$

To acquire the estimations, we have as we did similarly in 4.1.1

$$\overline{C*}_{D,1}^{(k)} = 0.129 + 0.002087t_1 + D(13.6222e^{-0.3495t_1} + 3.0095e^{-0.01843t_1})$$

$$\overline{C*}_{D,24}^{(k)} = 0.129 + 0.002087(t_1 + 23) + D(13.6222e^{-0.3495(t_1+23)} + 3.0095e^{-0.01843(t_1+23)}),$$

$k = 1, 2, ..., n$, which yields estimates $\hat{D}$ and $\hat{t}$ for each design dose $D = 1$ and $D = 4$ and we repeat this $N = 100$ times to simulate 100 samples and acquire $\hat{D}^{(1),(2),...,(N)}$ and $\hat{t}^{(1),(2),...,(N)}$ for each $D = 1$ and $D = 4$.

4.2.2 GLM

To avoid over complicating notations, the notations in this section 4.2.2 are defined for this section only and do not carry over to other sections unless otherwise specified.

GLM, generalized linear model, is a variant of LM, linear model. Suppose we have a data set with n samples with Y_i being the response, $i = 1, ..., n$. With LM, we assume the response variables Y_i are independently drawn from normal distributions, and are linearly dependent on the predictors. With GLM, we assume the response variables are independently drawn from a chosen distribution the user deems suitable, and a function of Y_i: $g(Y_i)$: is linearly dependent on the predictors. Here, $g(\cdot)$ is the link function and is again a chosen function the user deems suitable. GLM is used to model data that violates the normal assumption of LM (for example, in our case, count data clearly isn't normal) and is able to capture nonlinearity using prexisting knowledge of the data set (by using a suitable link function).

We mentioned in section 3.2 that we fit a Poisson GLM with identity link, which is the same as fitting a linear WLS. Here the weights are the inverse of the variance of the response, which by the Poisson assumption is the same as it's estimate, which is the mean. This is of course not true as the Quasi-Poisson model assumption is that the variance is ϕ times the mean. But since we are not doing inference here, our only concern is the parameter estimations which remain the same as the weights we used to fit the line is the weights of the Quasi-Poisson model multiplied by the same scaler ϕ at each point. A minor point perhaps worth mentioning is that unlike WLS where the estimates are computed analytically, `glm` uses fisher scoring to find the parameters, so in this sense it is the same as IRLS, though this again has no effect on the estimations so is not our concern.

To show this, we first look at the Poisson GLM, suppose we again have a data set with n samples. We assume Y_i follows Poisson distribution with mean μ_i, $i = 1, ..., n$. In our case, the Y_is are the sum of foci count in each sample consisting of 500 cells. We use the identity link, so the link function:

$$g(\mu_i) = x_i^T \beta = \sum_{j=0}^{1} \beta_j x_{ij}.$$

By the Poisson assumption,

$$P(Y_i = k) = \frac{\mu_i^k e^{-\mu_i}}{k!}.$$

Written in exponential distribution form,

$$P(Y_i = k) = \exp(k \log \mu_i - \mu_i) \frac{1}{k!}.$$

For overdispersed Y_i, we assume, in addition, that the variance of Y_i is $\phi\mu_i$, so

$$P(Y_i = k) = \exp(\frac{k \log \mu_i - \mu_i}{\phi}) \frac{1}{k!}.$$

Here, ϕ remains the dispersion parameter, and the Poisson distribution is a special case with $\phi = 1$. So Likelihood

$$L = \prod_{i=1}^{n} \exp(\frac{y_i \log \mu_i - \mu_i}{\phi}) \frac{1}{y_i!},$$

and log Likelihood

$$l = \sum_{i=1}^{n} (\frac{y_i \log \mu_i - \mu_i}{\phi} - \log y_i!).$$

Therefore, score equation

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{1}{\phi \mu_i} (y_i - \mu_i) x_{ij}.$$

This is equivalent to a WLS with weights $w_i = \frac{1}{\phi \mu_i}$

4.2.3 Setup

The rest of the setup of this simulation are the exact same as what we had in section and we still display the simulation results with Box Whisker Plots, as well as the Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$.

Table 4.2: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$D = 1$, $t_1 = 1$	-0.00487288	0.0644764	0.20121
$D = 4$, $t_1 = 1$	0.0506653	0.249688	0.410283

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$D = 1$, $t_1 = 1$	-0.0619906	1.06874	0.806279
$D = 4$, $t_1 = 1$	0.0332338	0.233304	0.38811

4.2.4 Results

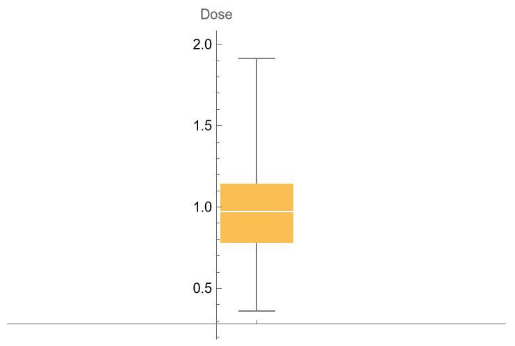
We display the results with Box Whisker Plots in figures 4.3 and 4.4, as well as the relative statistics in table 4.2. It seems from the results that our method still holds relatively well with overdispersed data except when estimating t_1 when $D = 1$, where the bias and variance are a bit higher compared to the other estimations. This, however, should not be too much of a problem since we would like some information about the time of exposure but we are a lot more interested in get a good dose estimation than a time estimation.

4.3 The Choice of n

4.3.1 Setup

Recall that n is denoted as the number of cells, or the number of foci counts we generate in each sample. We would like to give some insight into the choice of n, namely, the effect of different values of n on the reliability of the result. We test 4 different values of n: 200, 500, 1000, and 2000. We do not seek to test for values of n larger than 2000 due to the large computational cost.

The rest of the setup of this simulation are the exact same as what we had in section 4.2 and



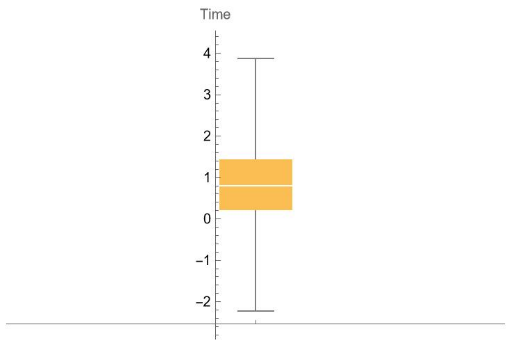
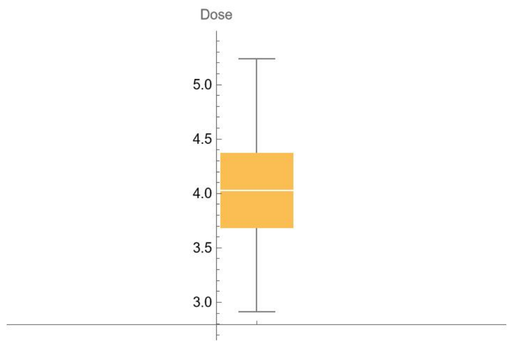


Figure 4.3: $\hat{D}$ and $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$



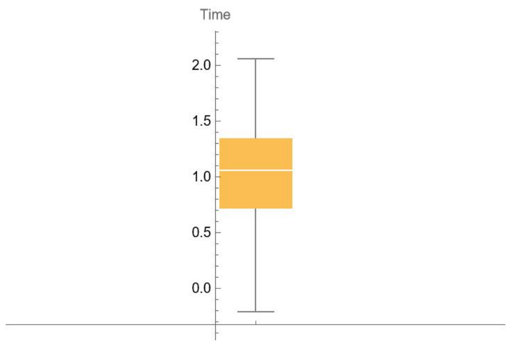


Figure 4.4: $\hat{D}$ and $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

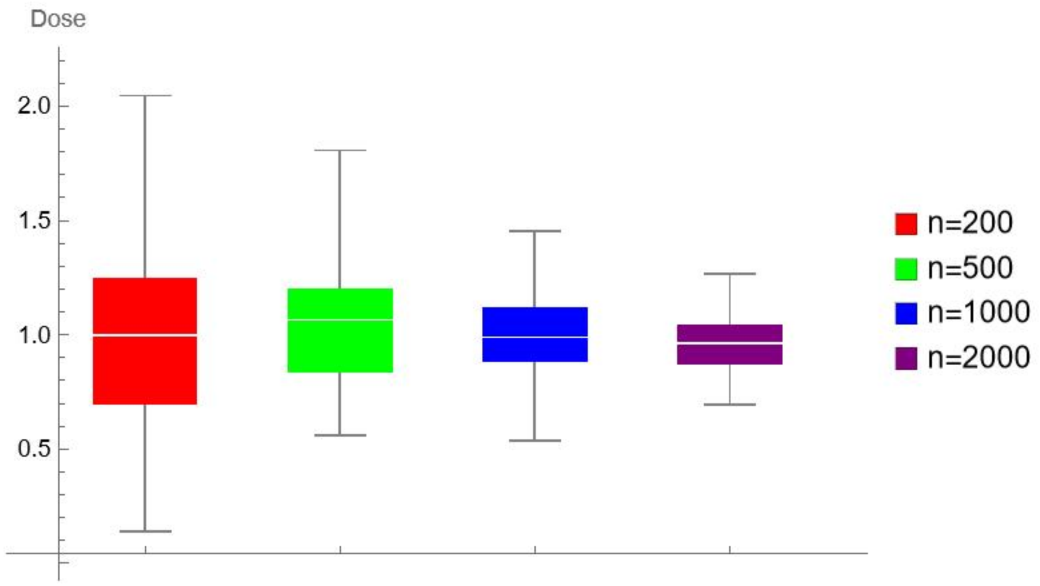


Figure 4.5: $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

we still display the simulation results with Box Whisker Plots, as well as the Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$.

4.3.2 Results

We display the results with Box Whisker Plots and the relative statistics in figures 4.5 to 4.20. We see from these results that the Variance and the Absolute Bias decreases as n increases, which is to be expected. The Bias seems to fluctuate with no visible pattern. However, when looking at the magnitude of the bias, it is generally quite small when compared to the values of D and t_1, indicating that the accuracy of the model is quite good even with small n. Overall, having $n = 200$ is certainly not good enough, and there is reason to argue for increasing n from 500 if there is an abundance of time and resources.

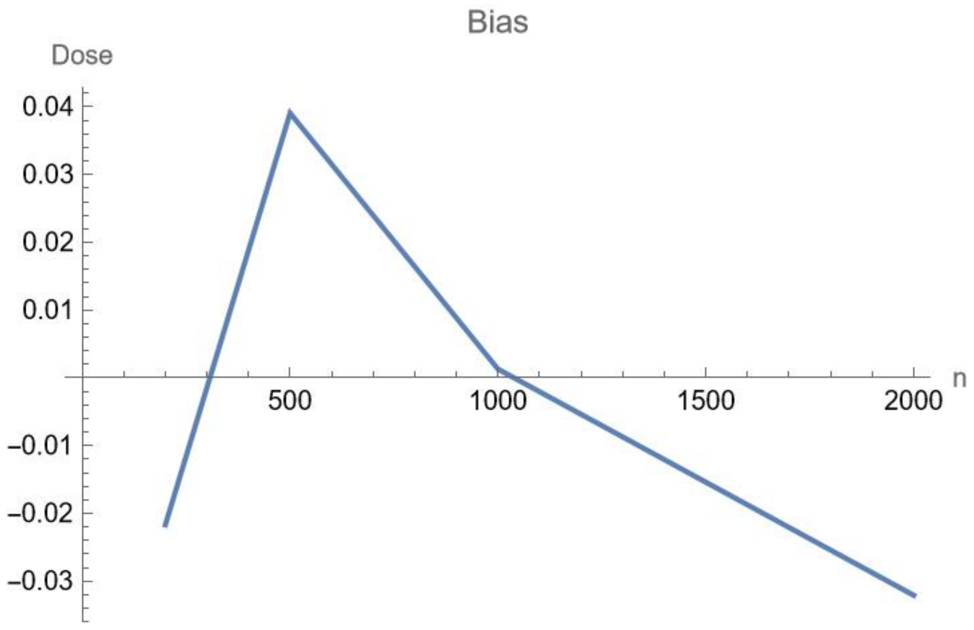


Figure 4.6: Bias of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

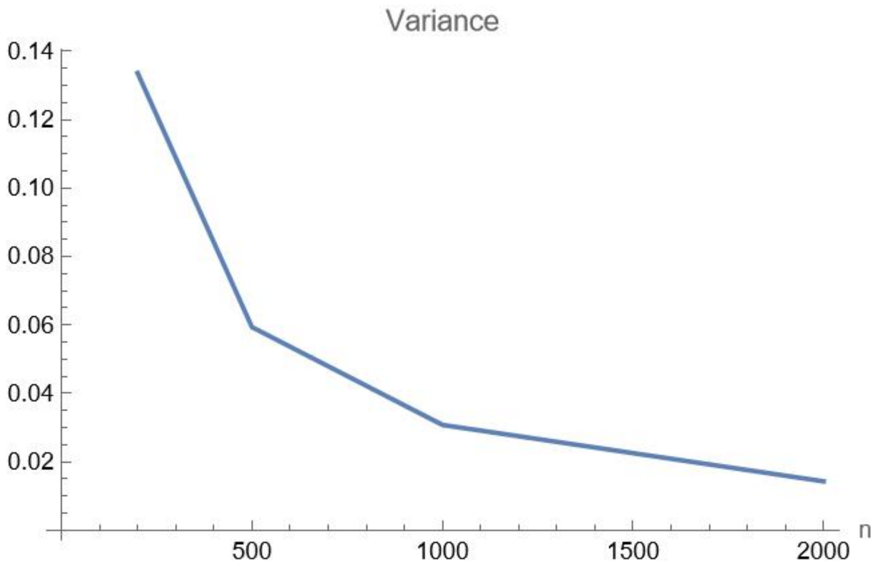


Figure 4.7: Variance of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

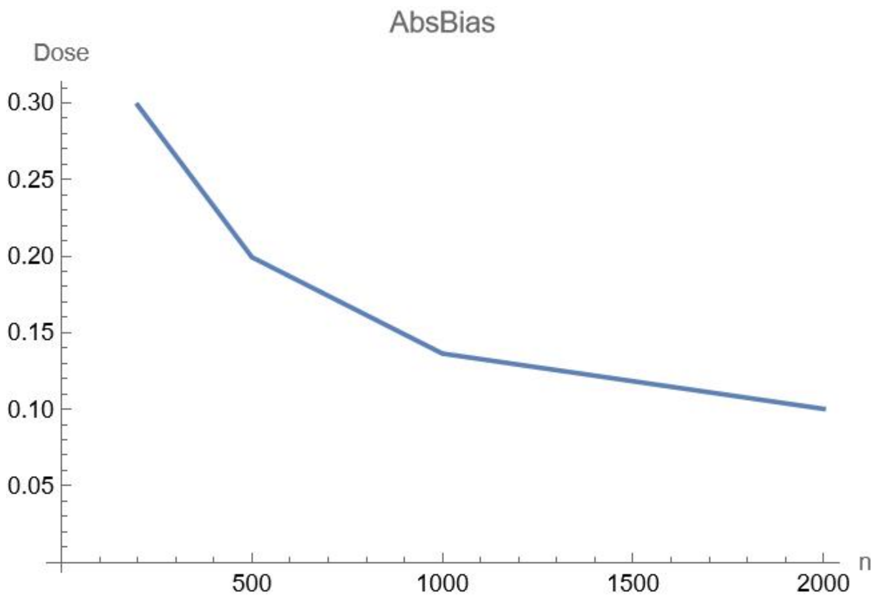


Figure 4.8: Absolute Bias of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

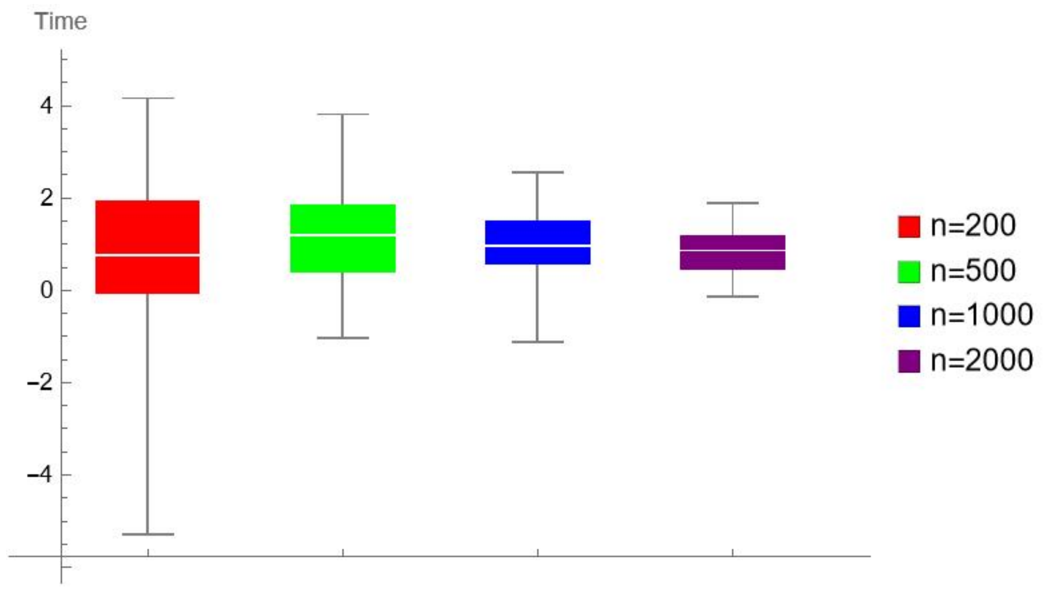


Figure 4.9: $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

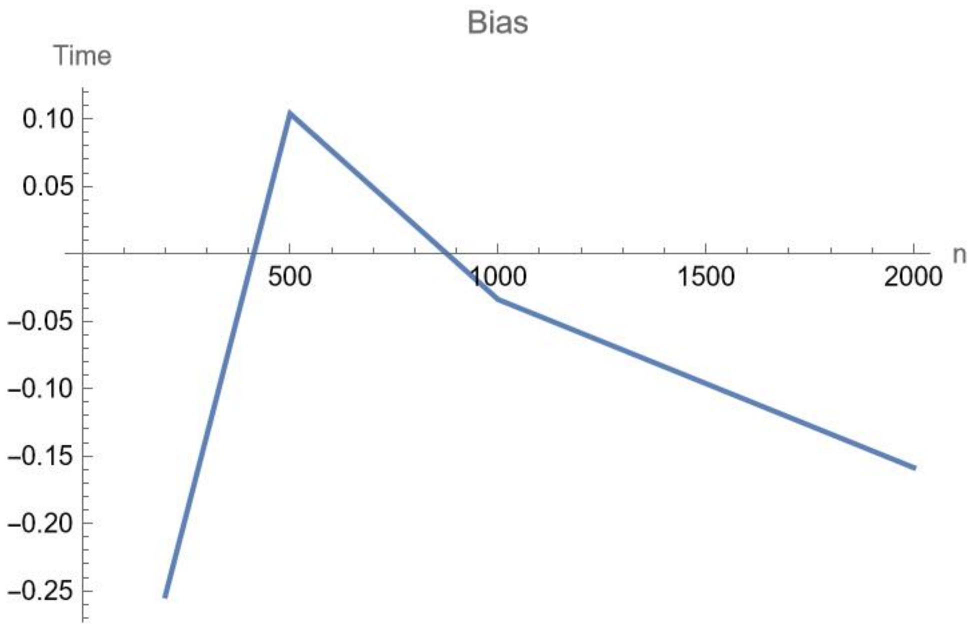


Figure 4.10: Bias of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

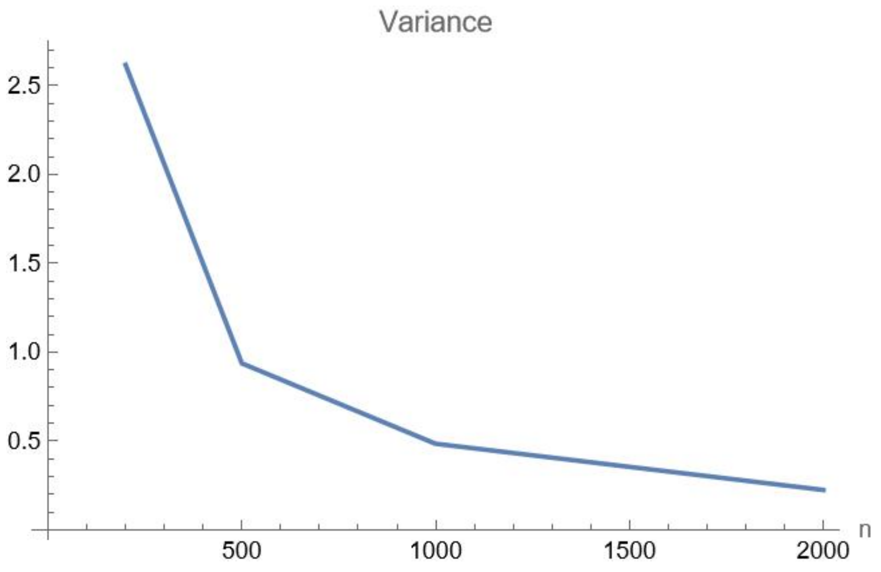


Figure 4.11: Variance of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

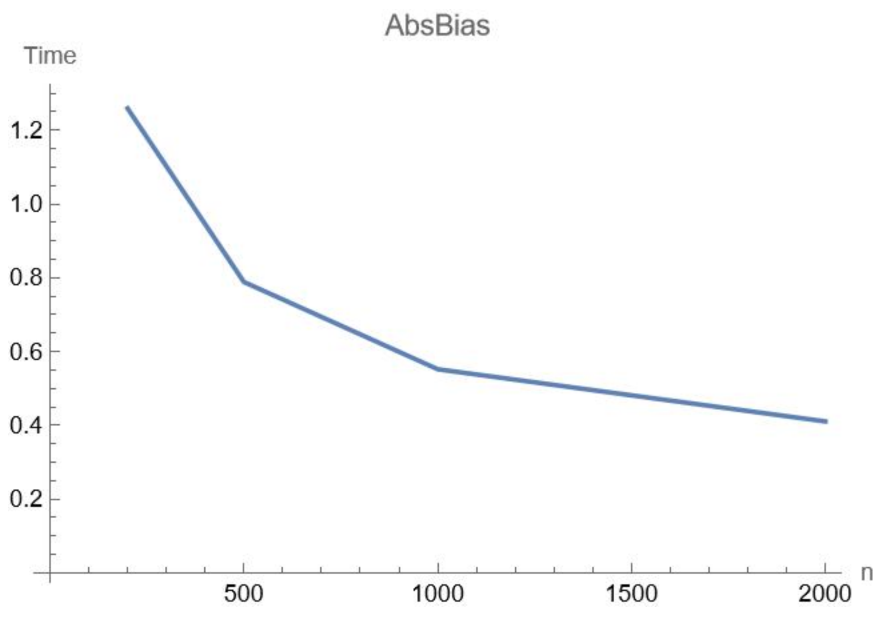


Figure 4.12: Absolute Bias of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1$

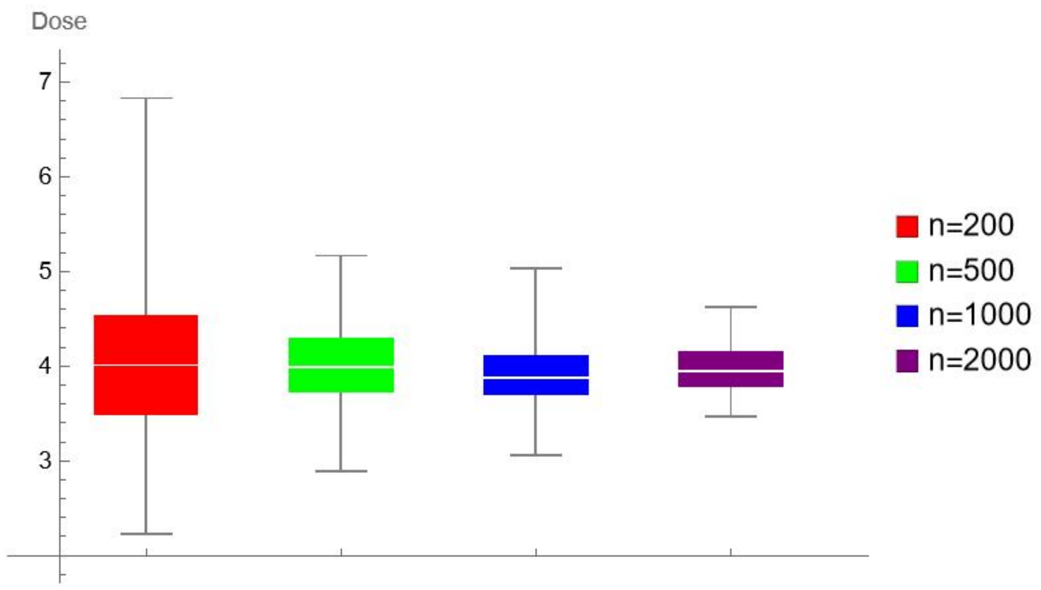


Figure 4.13: $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

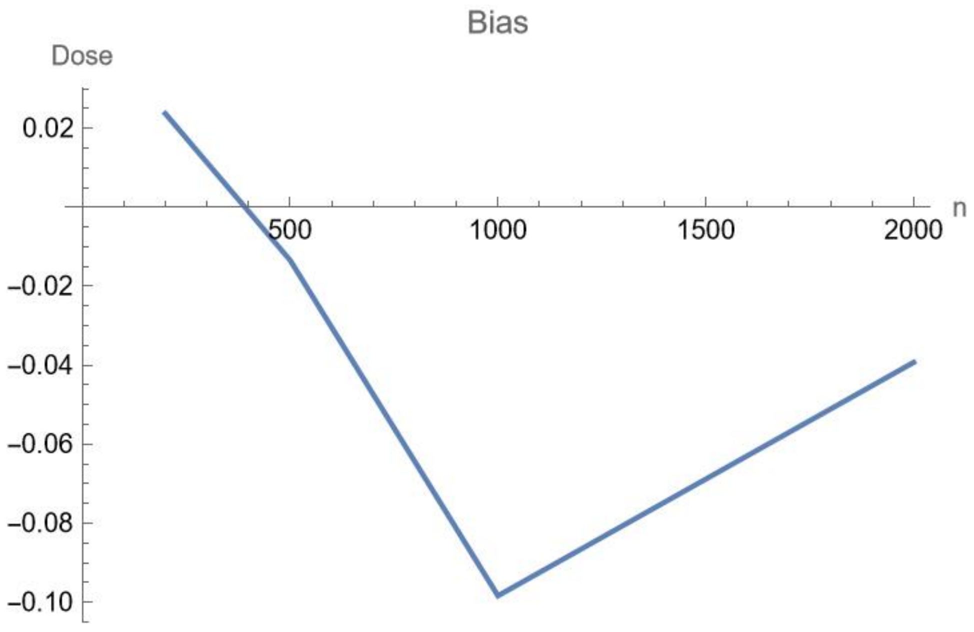


Figure 4.14: Bias of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

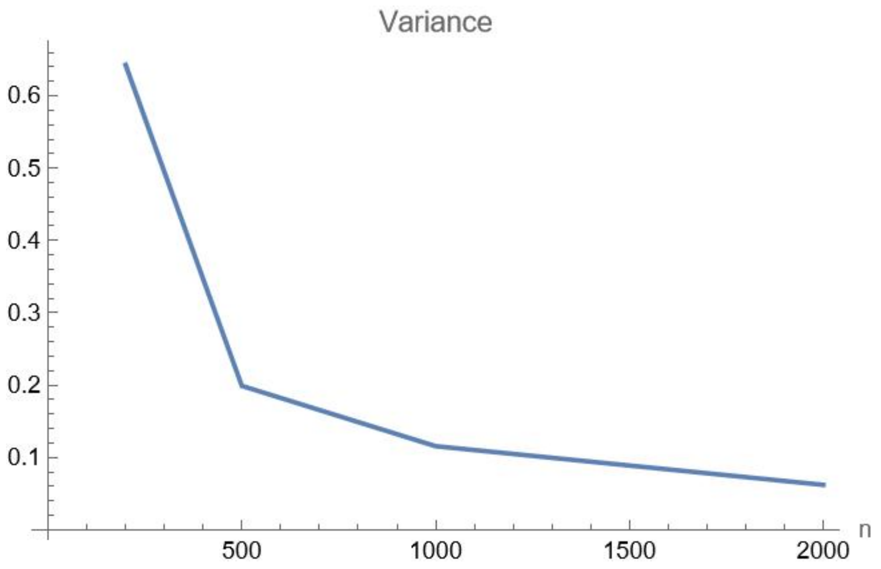


Figure 4.15: Variance of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

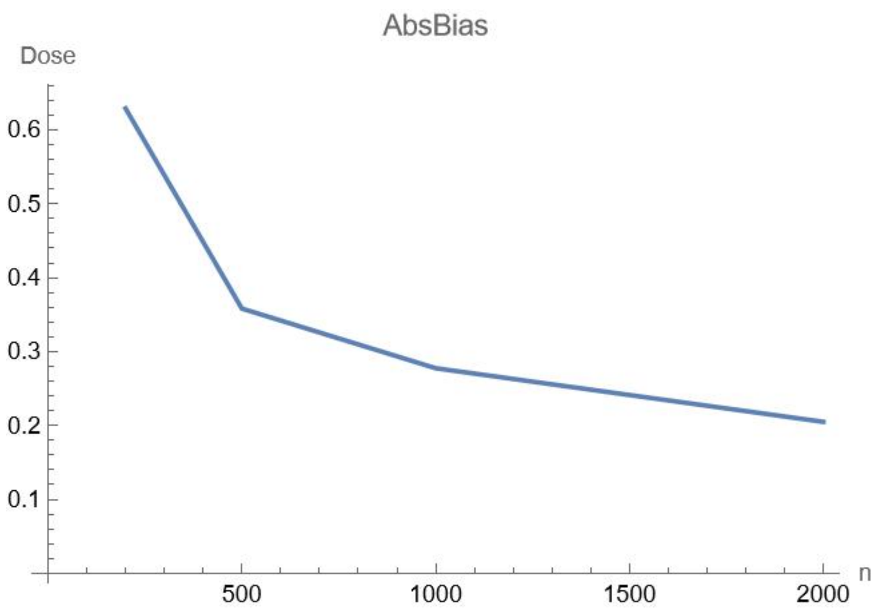


Figure 4.16: Absolute Bias of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

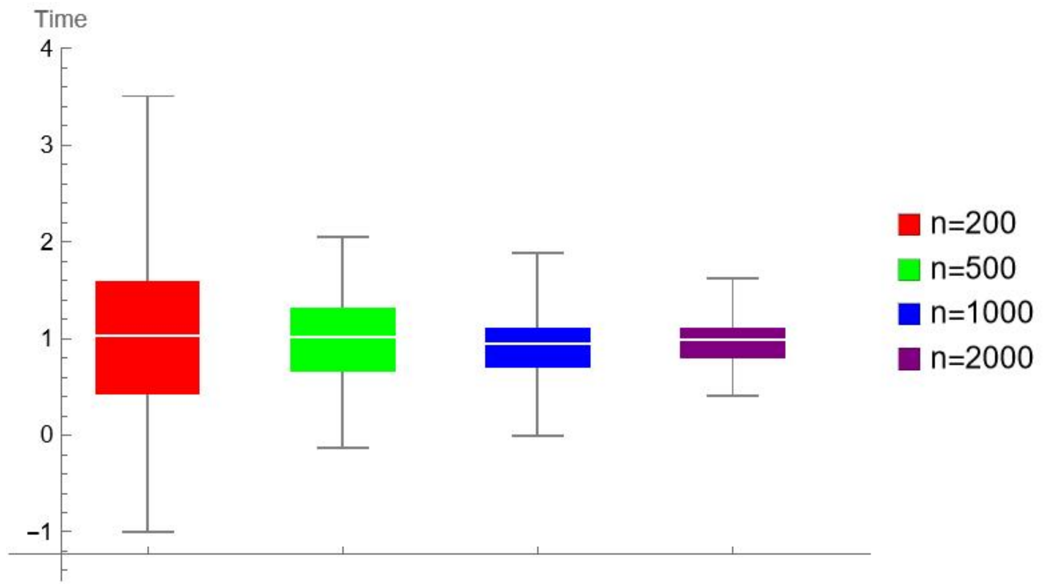


Figure 4.17: $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

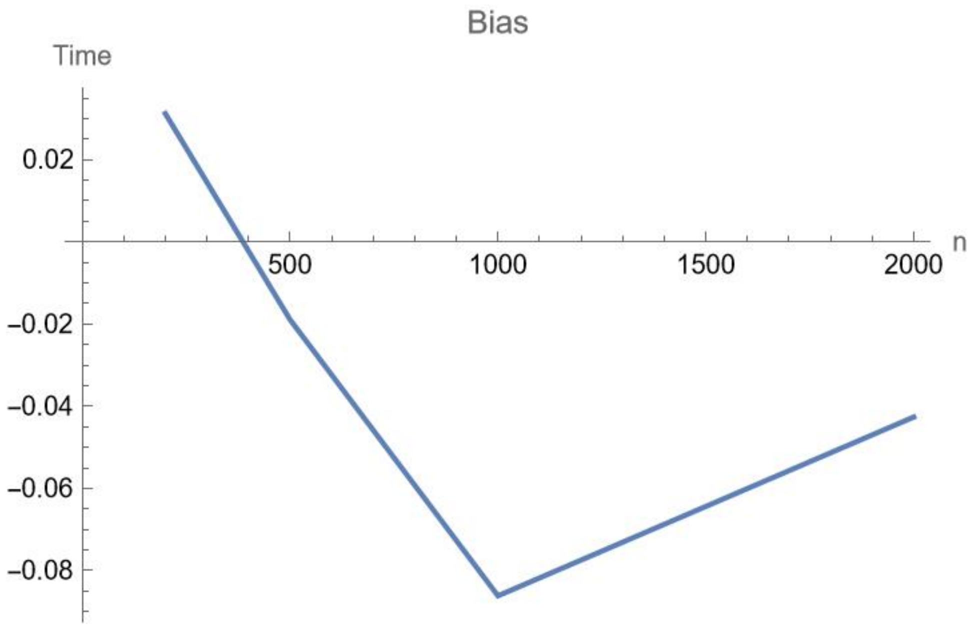


Figure 4.18: Bias of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

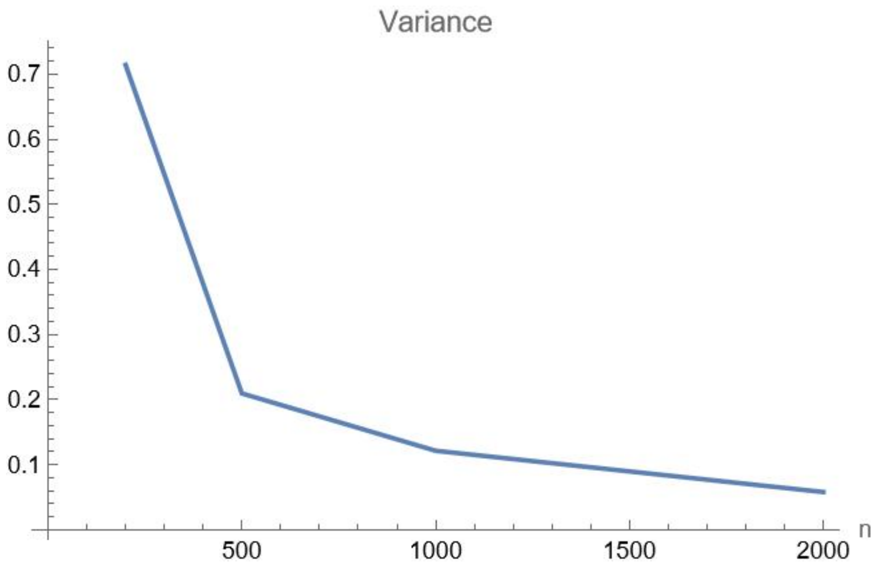


Figure 4.19: Variance of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

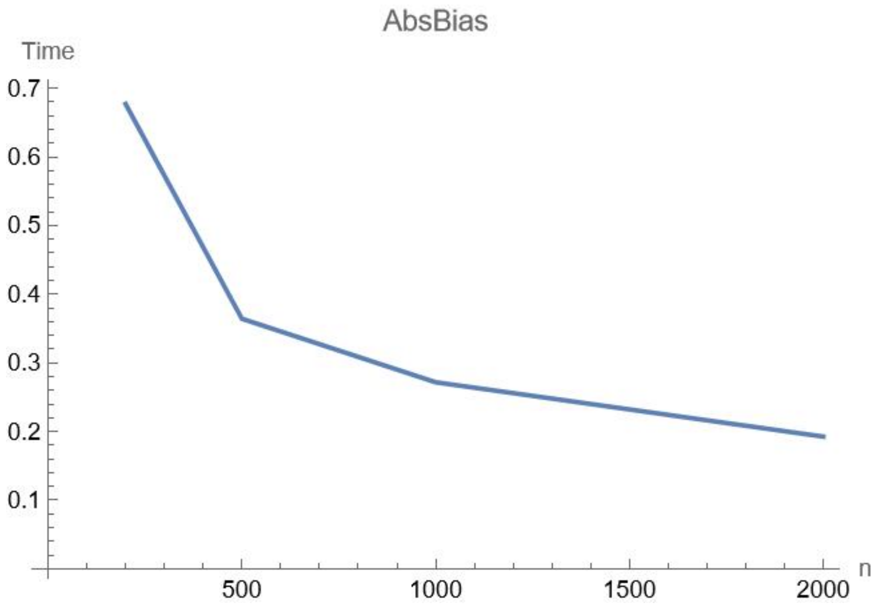


Figure 4.20: Absolute Bias of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1$

4.4 The Choice of Δ

4.4.1 Setup

Ideally, we would like our estimation to be as quick as possible and so we would like to have t_1 and t_2 relatively close to each other. However, it is obvious that the Bias, Variance and Absolute Bias of both $\hat{D}$, $\hat{t}$ will blow up if t_1 and t_2 are too close, in fact, we see in our testings that when the values of t_1 and t_2 are too close, we have extreme values of Bias, Variance and Absolute Bias that reaches as high as the magnitude of 10 to the power of a few hundred. So, how small can Δ get while the model remains reasonably reliable? Further, how does the reliability of the model change when we have bigger or smaller D or t_1? To answer these questions, we present 'maps' using different values of D, t_1 and t_2 in this section so that we can have a holistic understanding of how the choices of D, t_1 and Δ will effect the Bias, Variance, and Absolute Bias of both $\hat{D}$, $\hat{t}$.

We continue using the overdispersion framework from section 4.2 and we will examine the cases of $D = 0.5, 1, 2, 4$. In each case, we record the Bias, Variance and Absolute Bias of both $\hat{D}$ and $\hat{t}$ when $t_1 = 1, 2, ..., 24$ and $t_2 = 1, 2, ..., 24$.

4.4.2 Presentation of Results

We seek to display the results in 3D plots. Namely, we have t_1 and t_2 on the x and y axis and the values of the absolute value of Bias or Variance or Absolute Bias on the z axis. Note that we are displaying the absolute value of the bias of $\hat{D}$ and $\hat{t}$ here that we define as

$$\left|Bias_{\hat{D}}\right| = \left|\frac{\sum\limits_{k=1}^{N} \hat{D}^{(k)}}{N} - D\right|$$

$$\left|Bias_{\hat{t}}\right| = \left|\frac{\sum\limits_{k=1}^{N} \hat{t}^{(k)}}{N} - t_1\right|,$$

where $k = 1, 2, ..., N$, which is not the same as the Absolute Bias that we defined as

$$AbsBias_{\hat{D}} = \frac{\sum\limits_{k=1}^{N} \left|\hat{D}^{(k)} - D\right|}{N}$$

$$AbsBias_{\hat{t}} = \frac{\sum\limits_{k=1}^{N} \left|\hat{t}^{(k)} - t_1\right|}{N},$$

where $k = 1, 2, ..., N$. For example, take a point (i, j, Z) in the plot, Z would be the Bias or Variance or Absolute Bias with $t_1 = Min(i, j)$ and $t_2 = Max(i, j)$. However, due to the extremely large values in the results, we can not display the Bias or Variance or Absolute Bias as it is since we would only be able to see a thin "ridge" near $t_1 = t_2$ and "plains" everywhere else. To manage this, we instead show the log of the Bias, Variance and Absolute Bias with base 10 to give us a better idea of the behavior of these data instead of the exact values. We decided to display the absolute value of Bias instead of the original for the reason that the 3D plots would be very difficult to read with both positive and negative values of Z. It is safe to assume the values of the Bias, Variance and Absolute Bias of both $\hat{D}$, $\hat{t}$ go to infinity (or negative infinity in the case of Bias) when $t_1 \to t_2$, therefore, for each point in the plot with $t_1 = t_2$, we assign to its z coordinate the z coordinate of the larger z coordinate of the adjacent points with $t_1 + 1 = t_2$. For example, if we have point (1,2,8) and point (2,3,10) in the plot, then the point at $t_1 = t_2 = 2$ will have coordinates (2,2,10).

The 3D plot gives us some idea of the behaviors of the Bias, Variance and Absolute Bias, however, by itself it is not enough to answer our questions above. In fact, we would like a "one stop" guide on which pairs of t_1 and t_2 will likely produce reliable results. Therefore, we present a 2D "activation map" of the Absolute Bias which gives us a direct idea of the performance of different pairs of t_1 and t_2.

We define a feasible pair in terms of $\hat{D}$ or $\hat{t}$ as a pair of t_1 and t_2 with Absolute Bias of $\hat{D}$ or $\hat{t}$ less than a certain cutoff. Any other combinations would be defined as infeasible combinations. For the estimation of D, we set the cutoff to be 1, that is we consider a pair of t_1 and t_2 as "good enough" to produce reliable results if the Absolute Bias is less than 1. For the estimation of t_1 we set a more generous cutoff of 48 as, again, we would like to have some idea of the time of exposure but we mostly care about the dose. In the activation map, we display the infeasible combinations of t_1 and t_2 with white blocks. For the feasible combinations, we display darker blocks that are darker with lower values of Absolute Bias. Therefore, a feasible pair with Absolute Bias close to 0 would be assigned an almost black block while a feasible pair with Absolute Bias close to 1 would be assigned an almost while block and all infeasible pairs would be assigned a pure white block. By looking at this activation map, we can immediately have an idea of how likely we will have reliable results with a pair of t_1 and t_2.

4.4.3 Results

3D Plots

We will first look at the 3D plots in figures 4.21 to 4.32. The higher values in the plots would have colors corresponding to lower frequency lights with red being the most extreme and the lower values in the plots have colors corresponding to higher frequency lights with purple being the most extreme. The plots look like "mountains" with their "ridge" near $t_1 = t_2$ as we expected. The "grey patches" visible on some of the plots correspond to the highest values in the plots, which are supposed to look red, but due to a bug in the plotting facilities of Mathematica, they were given a "null" color that leads to them being grey.

Activation Maps

We display the activation maps in figures 4.33 to 4.40.

As detailed in section 4.4.2, the X and Y axis of the activation maps are t_1 and t_2 and the corresponding block of a combination of t_1 and t_2 being shaded means they are feasible by

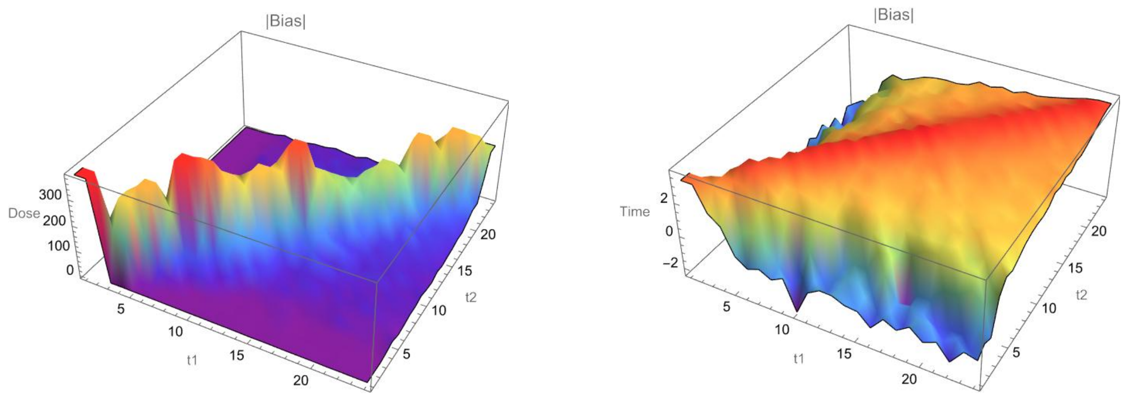


Figure 4.21: log (base 10) Absolute value of Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

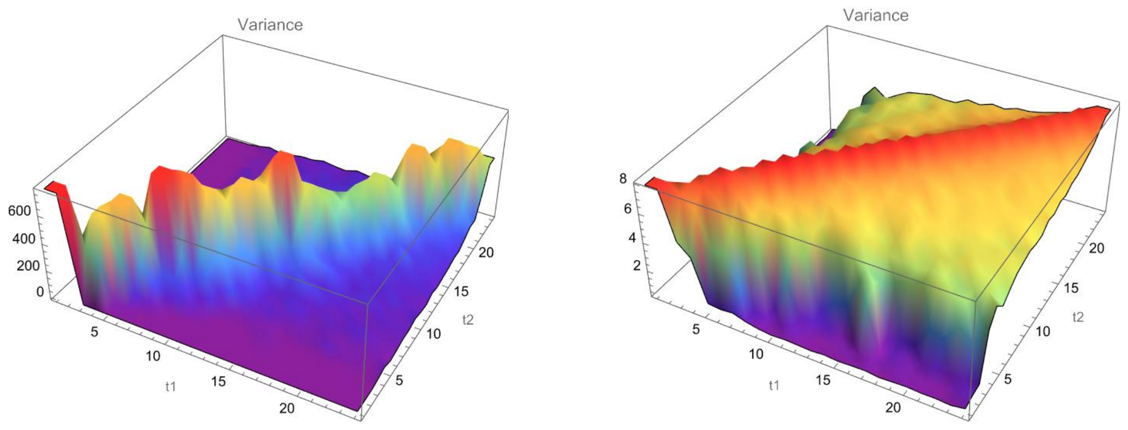


Figure 4.22: log (base 10) Variance of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

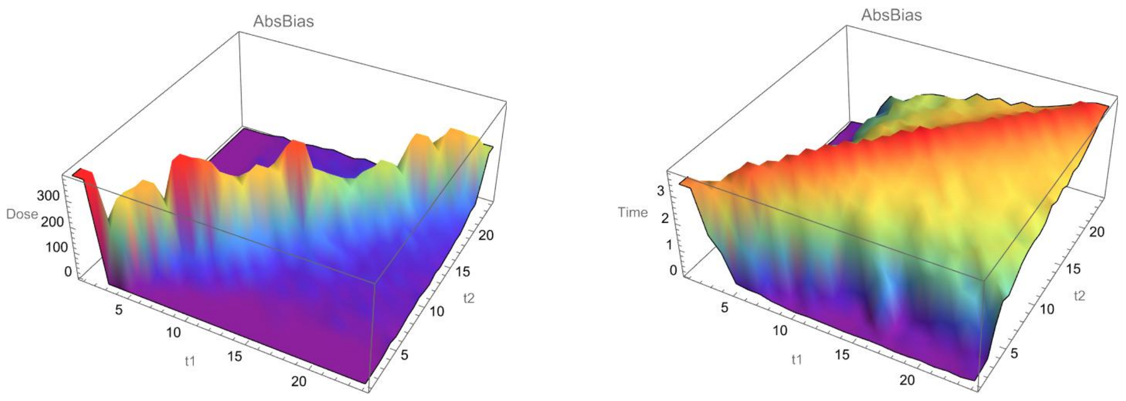


Figure 4.23: log (base 10) Absolute Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

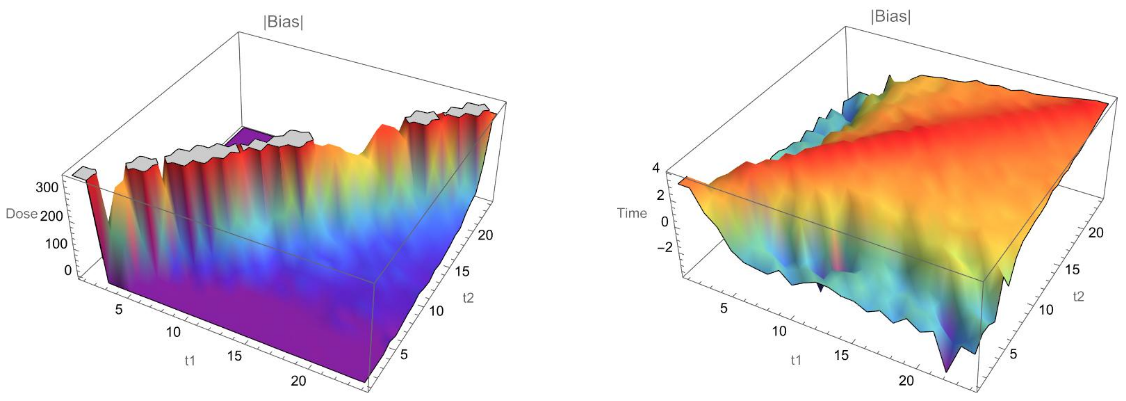


Figure 4.24: log (base 10) Absolute value of Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

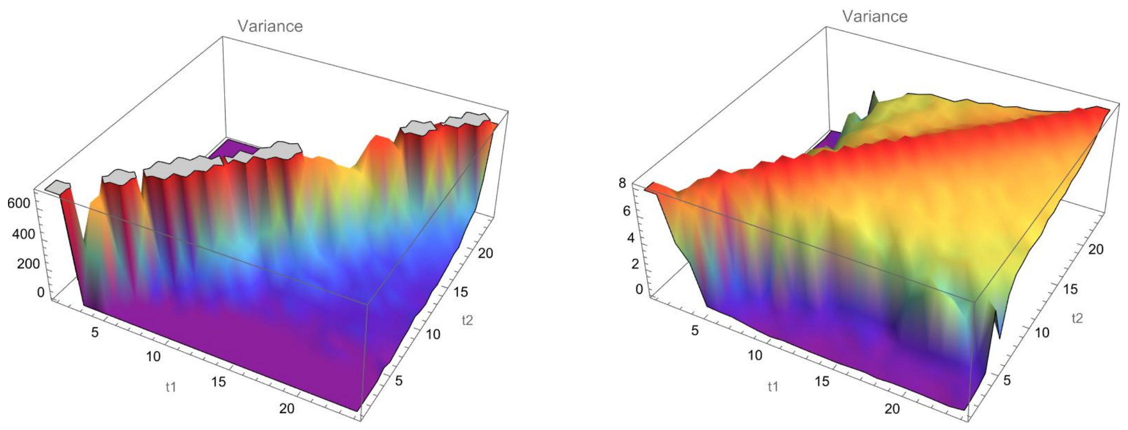


Figure 4.25: log (base 10) Variance of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

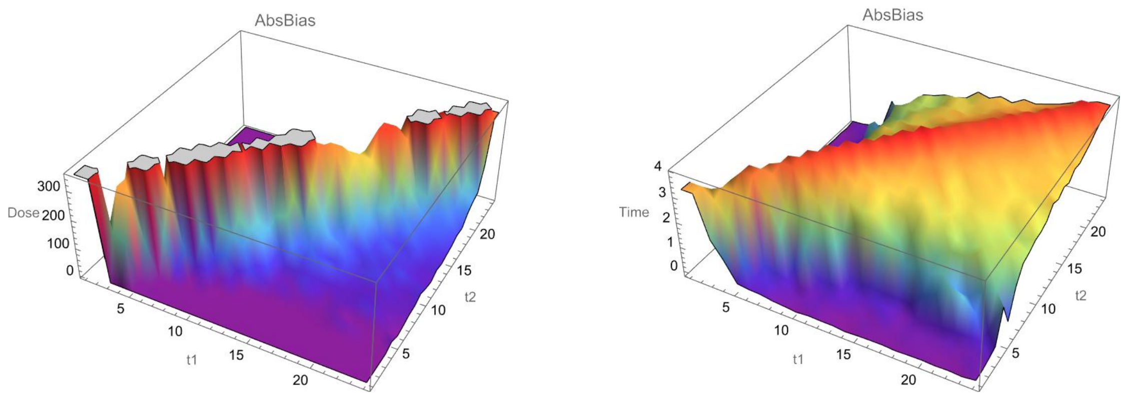


Figure 4.26: log (base 10) Absolute Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

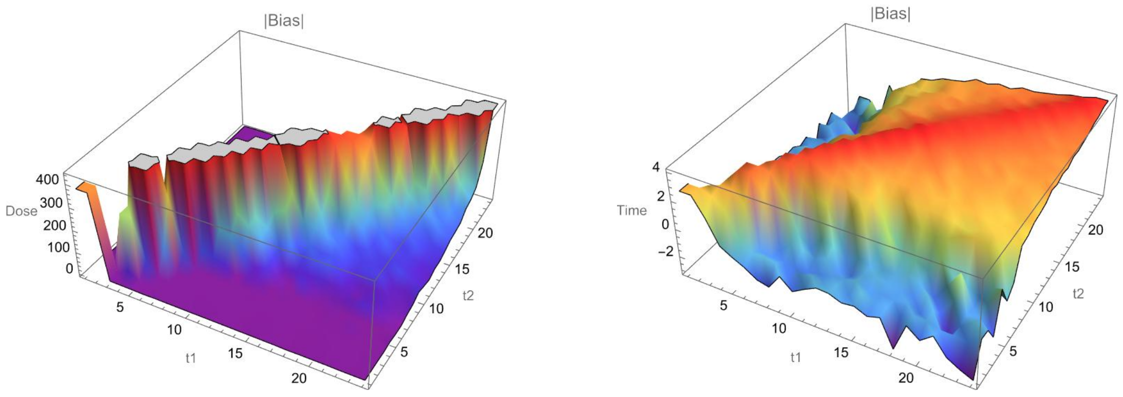


Figure 4.27: log (base 10) Absolute value of Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

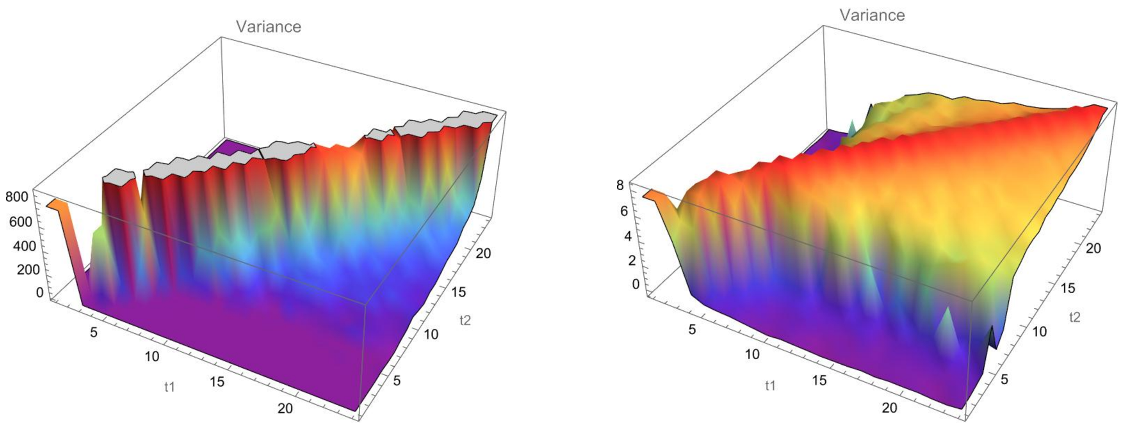


Figure 4.28: log (base 10) Variance of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

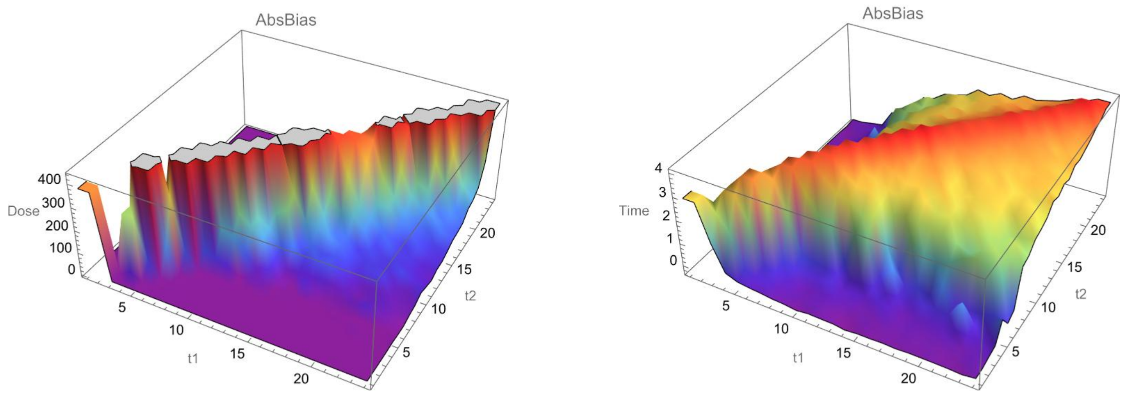


Figure 4.29: log (base 10) Absolute Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

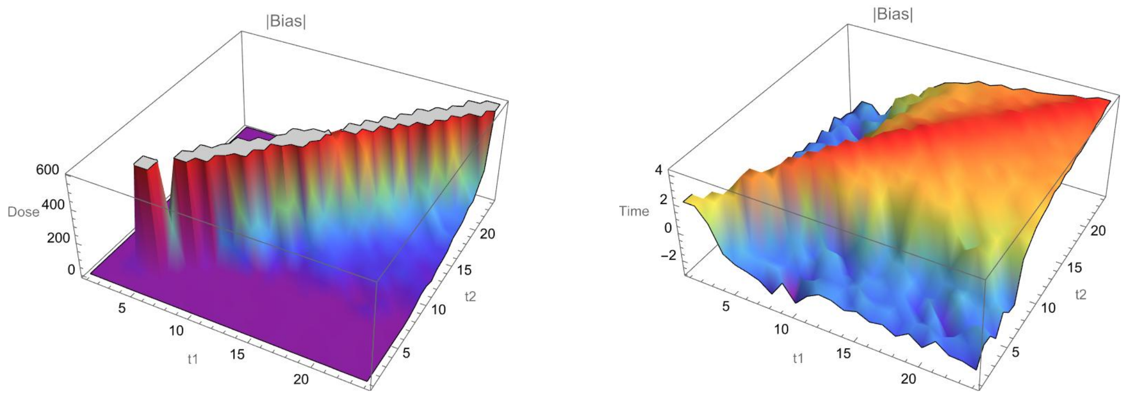


Figure 4.30: log (base 10) Absolute value of Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

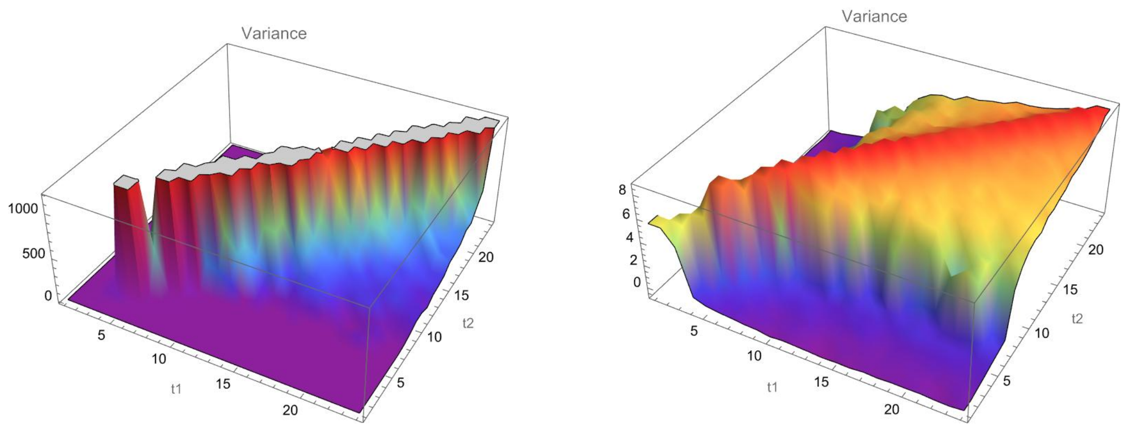


Figure 4.31: log (base 10) Variance of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

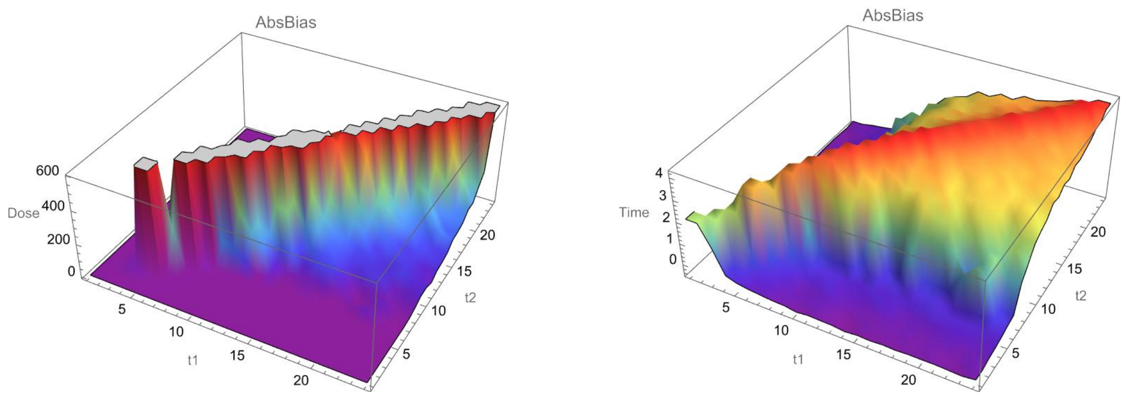


Figure 4.32: log (base 10) Absolute Bias of $\hat{D}$ (left) and $\hat{t}$ (right) from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

our definition. For example, the block with coordinates (5,10) in figure 4.40 is shaded. This suggests that the combination of taking blood samples 5h and 10h after exposure is feasible by our definition when dose is $4Gy$.

We will mostly be interested in using the activation maps of $\hat{D}$ as reference for real life estimations. We can see that the black blocks take up more space for both $\hat{D}$ and $\hat{t}$ as the dose increases. This is to be expected as bigger values of D will allow the Yield at each time point to have a bigger difference, which would likely allow the program to produce a better solution of the model.

Overall, a bigger Δ leads to a better estimation, but we can get away with a smaller Δ when the dose is higher. We should use any knowledge of the potential time and dose of exposure alongside the activation map when selecting Δ for a dose estimation.

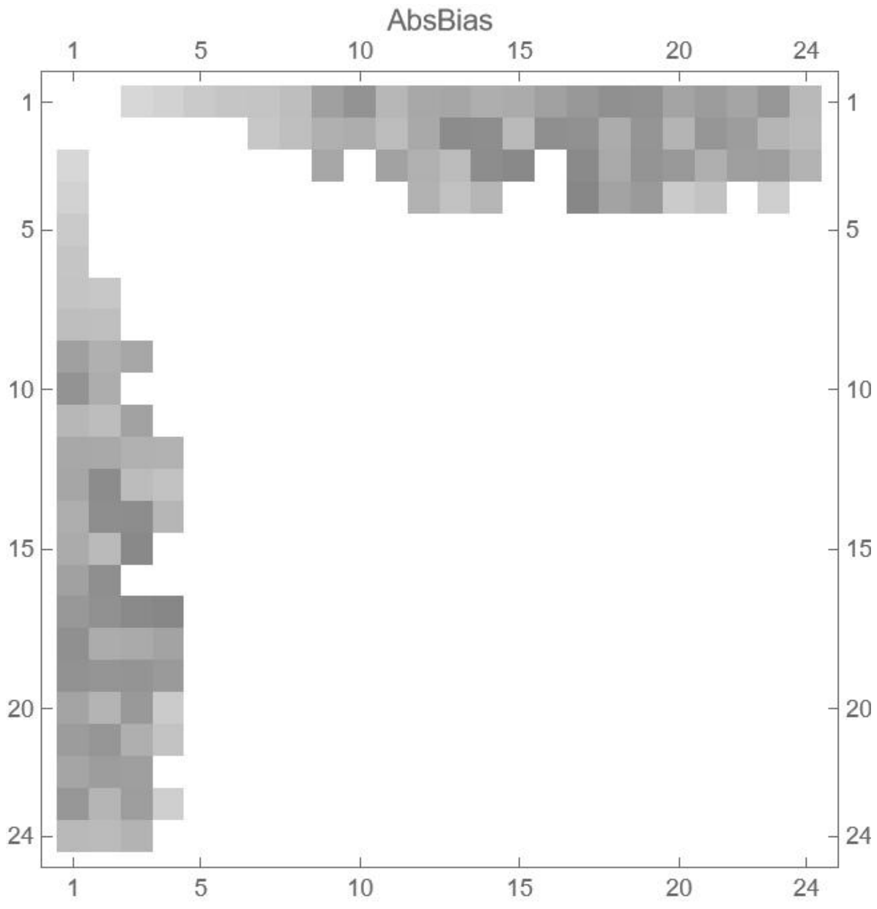


Figure 4.33: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

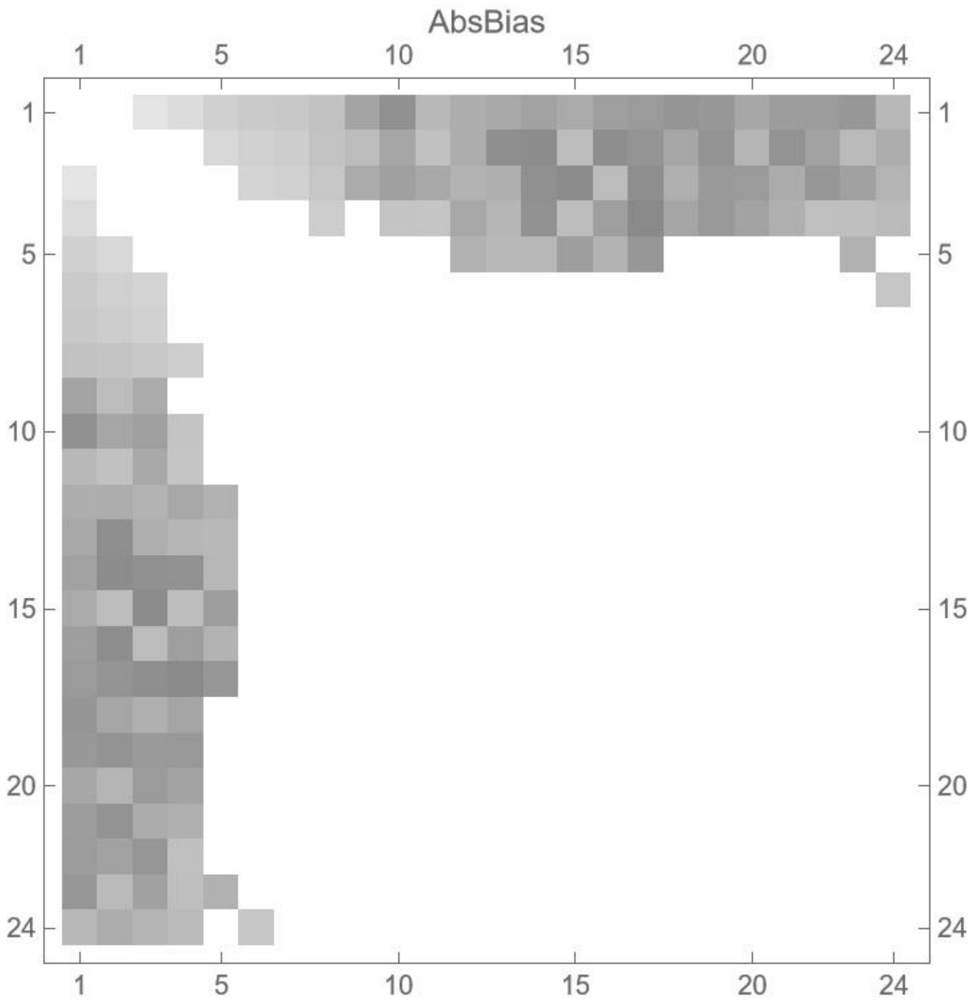


Figure 4.34: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

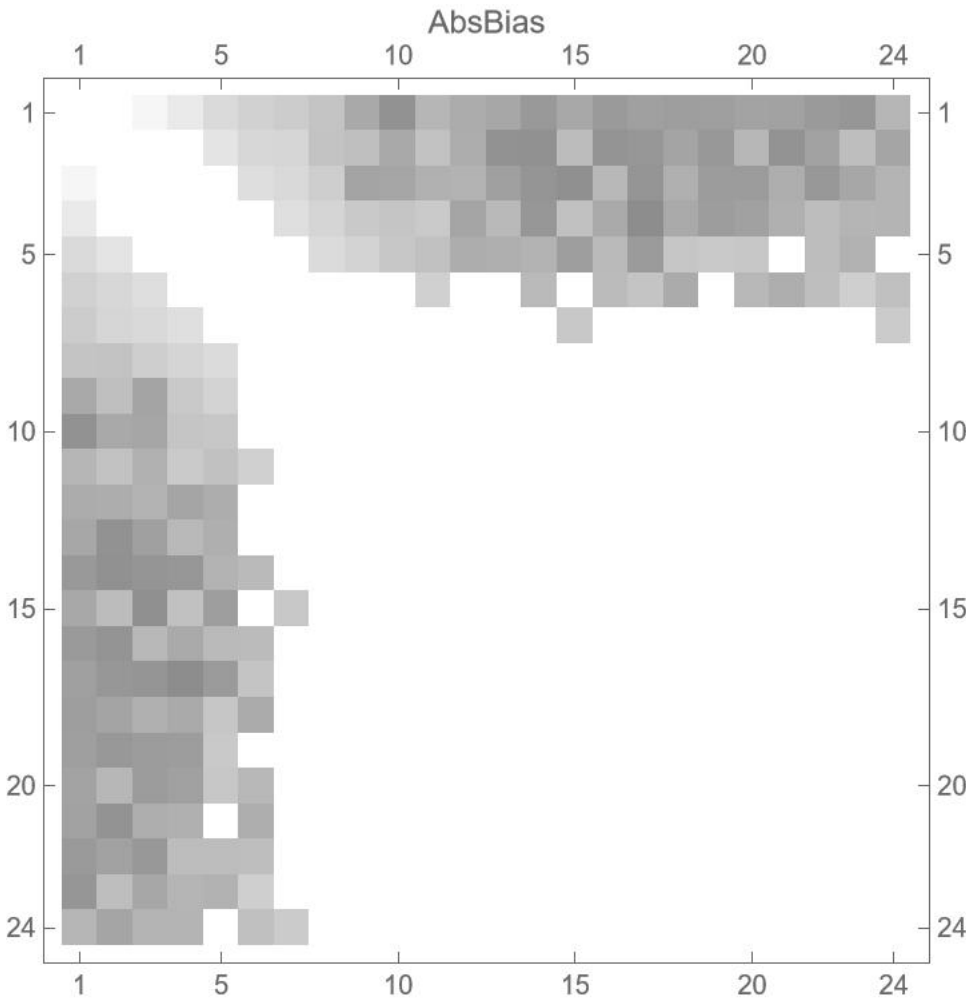


Figure 4.35: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

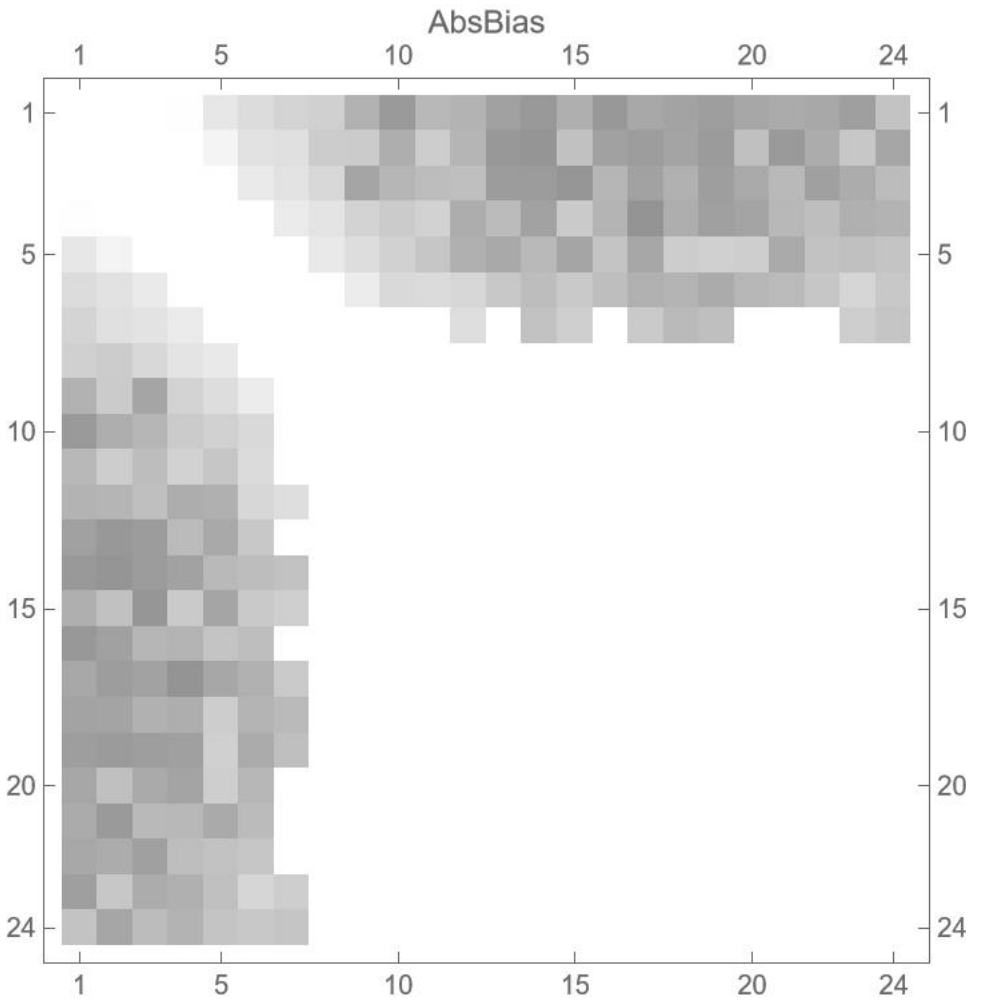


Figure 4.36: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

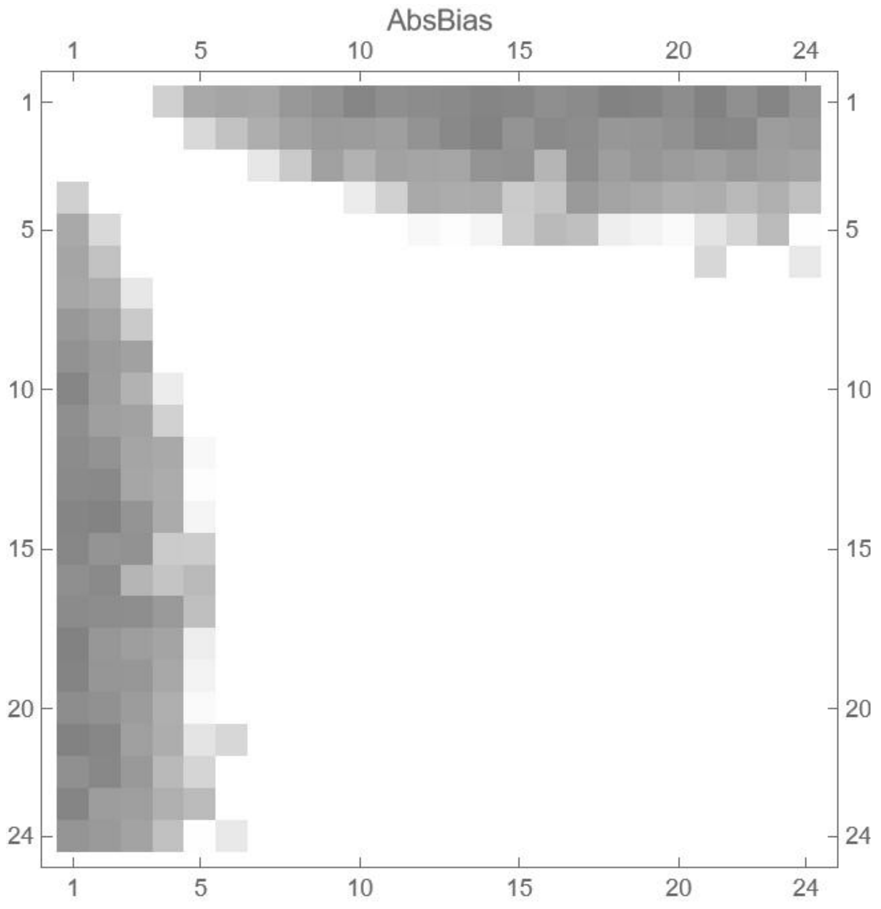


Figure 4.37: Activation Map with cutoff at 48 of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

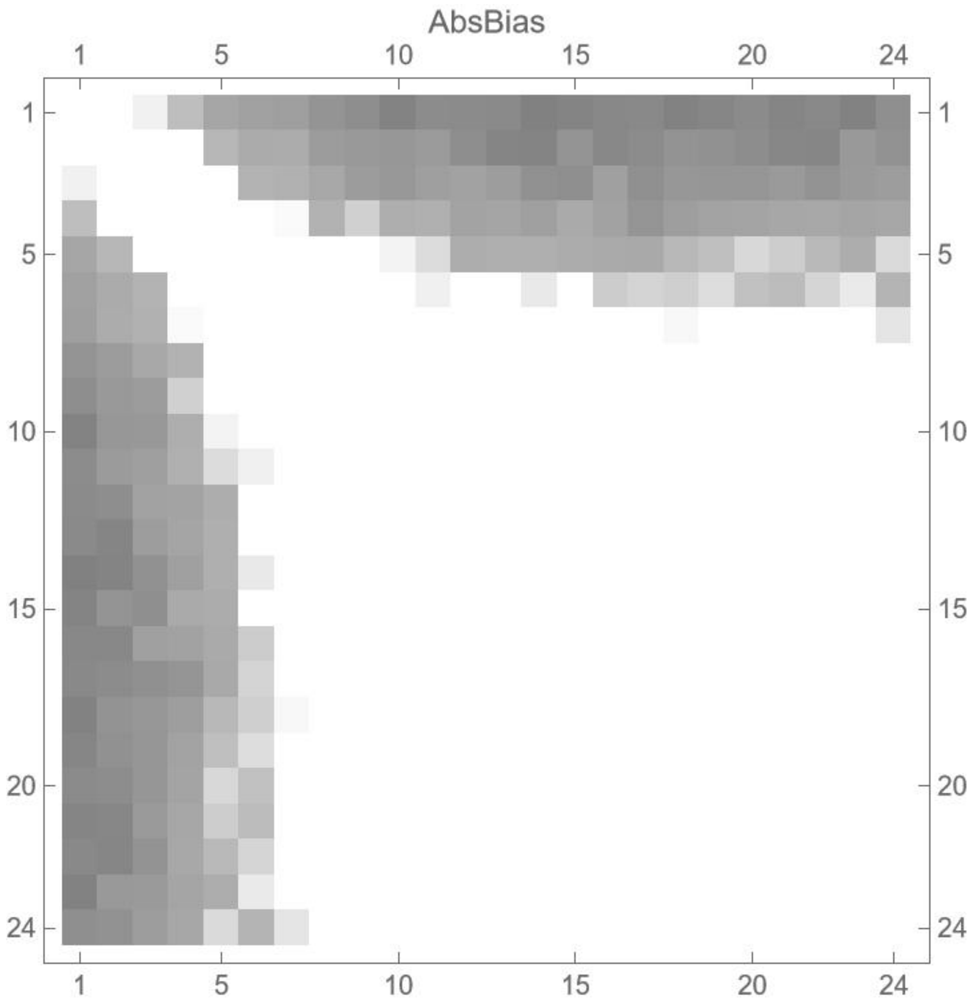


Figure 4.38: Activation Map with cutoff at 48 of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 1$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

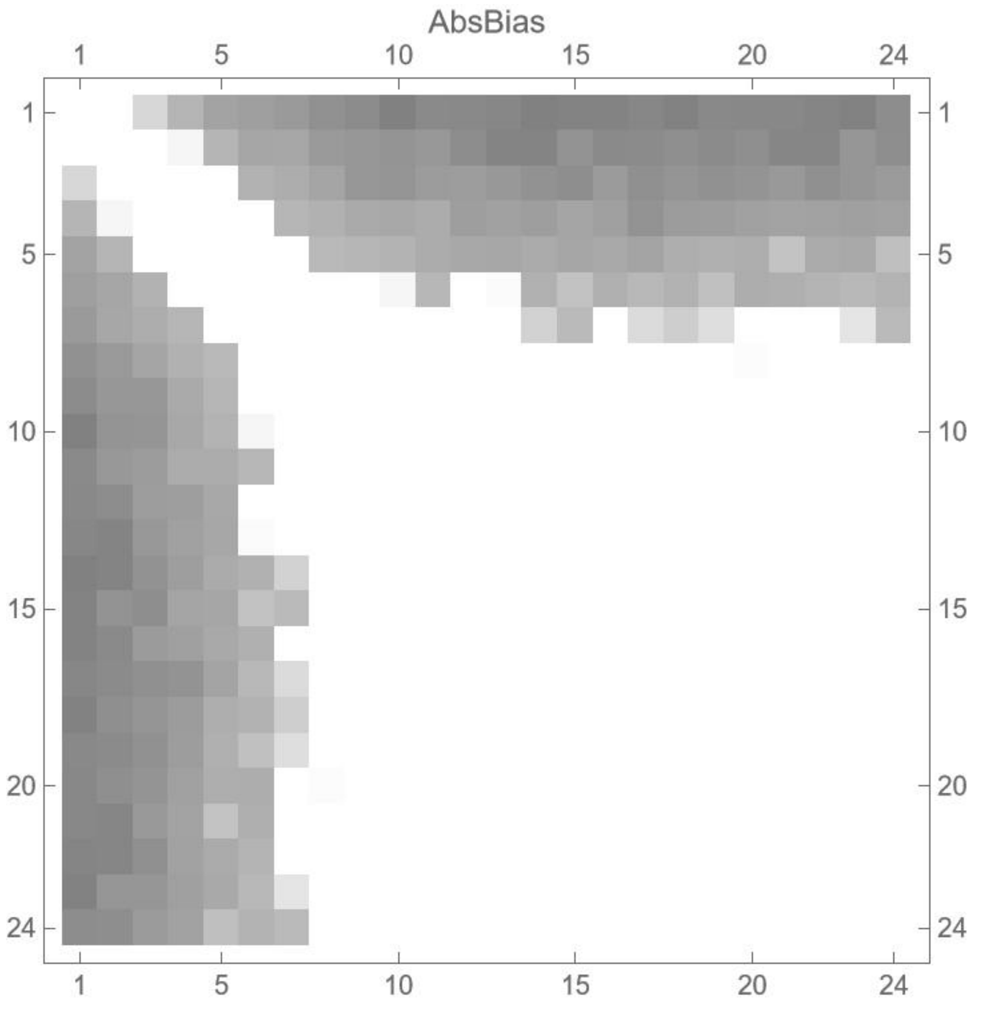


Figure 4.39: Activation Map with cutoff at 48 of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

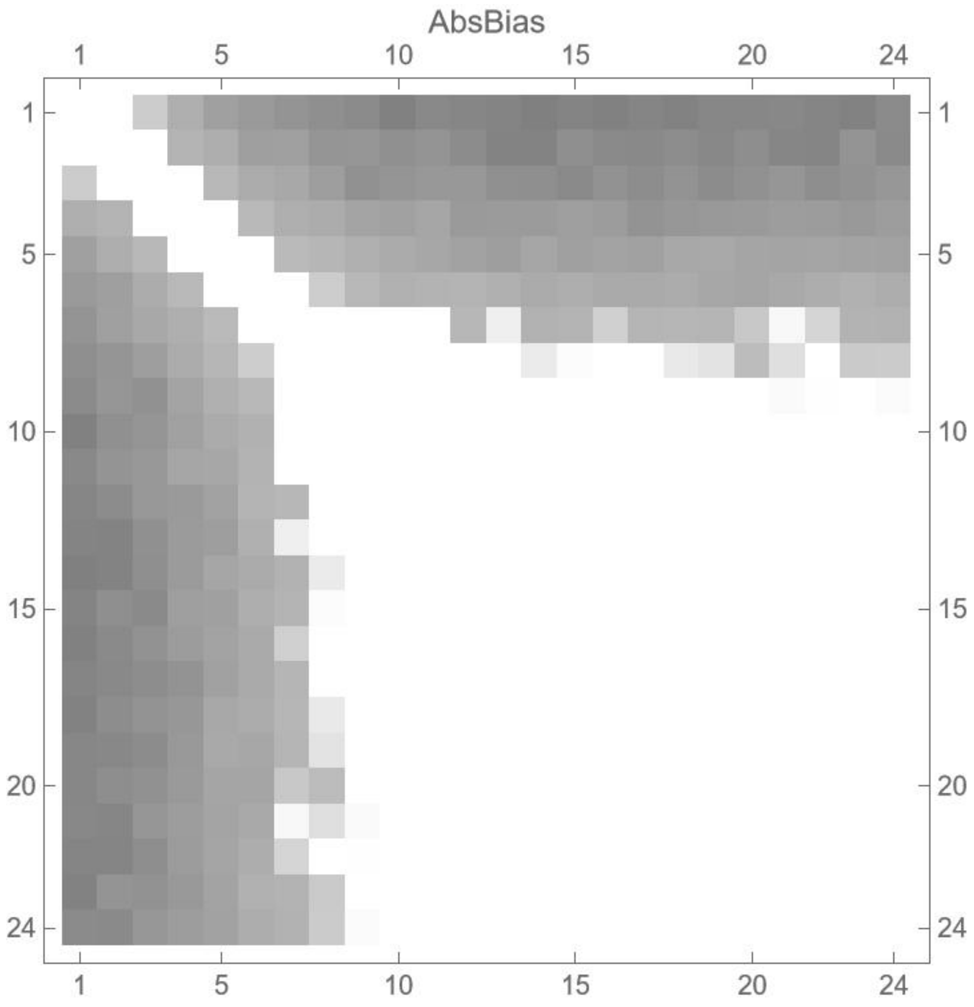


Figure 4.40: Activation Map with cutoff at 48 of $\hat{t}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 4$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

Chapter 5

Testing with Real Data

5.1 The Data

The methods discussed above are applied to a real data set produced at the chromosome dosimetry unit at UKHSA. Four blood samples from a healthy volunteer were exposed *in vitro* to ionizing radiation with design doses $0, 0.5, 1, 2$. For each sample, the foci count in $n = 200$ cells were taken at time points $1h, 2h, 4h, 24h, 48h$ after exposure. Table 5.1 displays the average foci count (Yield) at each individual design dose and time point.

Table 5.1: Laboratory data from UKHSA: Foci Yields in batches of 200 cells

	Average No. Foci			
Time Point (Hr)	0 Gy	0.5 Gy	1 Gy	2 Gy
1	0.675	2.265	4.62	5.905
2	0.165	2.045	3.79	6.04
4	0.15	1.74	3.17	4.48
24	0.265	0.545	1.12	2.025
48	0.17	0.635	0.76	1.285

From a calibration point of view, this is less than ideal, since we have previously shown in section 4.3 that choosing $n = 200$ significantly reduces the reliability of the results comparing to $n = 500$ or more, and we only have 1 sample for each design dose comparing to at least 16 samples from [12] in our simulations, but we have to work with what we have. In particular, we can immediately notice two anomalies highlighted in table 5.2. The average foci count with design dose $0.5Gy$ at time point $48h$ is larger than that of $24h$. This could be explained by an increase in Bias due to $n = 200$ being too small. Also, $0.5Gy$ is a relatively small dose, and would likely cause further bias as was shown in section 4.4 especially when the number of foci is low near the later time of $48h$. The average foci count with design dose $2Gy$ at time $1h$ is abnormally small, in fact, even smaller than that of $2h$. This could again be explained by the small n, but another cause of this could be that the number of foci at a relatively high dose of $2Gy$ at an early time of $1h$ being too high, which may cause foci to overlap more frequently in the counting process.

Table 5.2: Laboratory data from UKHSA: Foci Yields in batches of 200 cells

	Average No. Foci			
Time Point (Hr)	0 Gy	0.5 Gy	1 Gy	2 Gy
1	0.675	2.265	4.62	5.905
2	0.165	2.045	3.79	6.04
4	0.15	1.74	3.17	4.48
24	0.265	0.545	1.12	2.025
48	0.17	0.635	0.76	1.285

Hence, we remove the data with design dose $0.5Gy$ at time $48h$ and the data with design dose $2Gy$ at time $1h$. We end up with table 5.3.

Table 5.3: Laboratory data from UKHSA: Foci Yields in batches of 200 cells with two anomalies removed

	Average No. Foci			
Time Point (Hr)	0 Gy	0.5 Gy	1 Gy	2 Gy
1	0.675	2.265	4.62	
2	0.165	2.045	3.79	6.04
4	0.15	1.74	3.17	4.48
24	0.265	0.545	1.12	2.025
48	0.17		0.76	1.285

From the data, it is easy to see that the calibration curve we used in the examples from the last chapter will not work here. A quick verification of this using the applet from [12] shows that this data does not validate the calibration curves we have. This is due to the difference in conditions of different laboratories.

The applet we employed here is a web tool that produces dose estimates and confidence intervals using its own calibration curves, calibration curves provided by the user, or calibration curves generated from user data (reference samples) and plots the results. To ensure the validity of its estimates, the applet automatically checks reference samples against the user chosen calibration curves and gives warning if they do not align. This is what we used to invalidate the calibration curves we have.

When we introduced the basic model, we talked about needing two calibration curves from the same laboratory for the same reason. In fact, the calibration curves can be influenced by difference in laboratory[22], scoring mechanisms[27], technician performing the scoring[29], temperature at exposure[16], type of the cells where foci were counted[8], shipment[23]. There are reports of inter-individual and intra-individual variations[31][16] as well though this is accounted for by the overdispersion[12] that we have already considered.

The laboratory that produced our data did not provide any calibration curves. This is due to UKHSA using different calibration curves for different scenarios, which is similar to what we have done. Therefore, we will generate our calibration curves to test our model. Specifically, we use the average foci count at times $2h$ and $24h$ as our "training data" (table 5.4) to generate two calibration curves.

Table 5.4: The "training data" with which to create calibration curves

	Average No. Foci			
Time Point (Hr)	0 Gy	0.5 Gy	1 Gy	2 Gy
2	0.165	2.045	3.79	6.04
24	0.265	0.545	1.12	2.025

We use the average foci count at times $1h$, $4h$ and $48h$ along with the respective count in each $n = 200$ cells as our "testing data" (table 5.5). We are not overly concerned with what happens at $48h$ since one of the objectives of this project is to produce quicker dose estimations, but we will include it in our testings to make use of this data.

Table 5.5: The "testing data" with which to test the model

	Average No. Foci			
Time Point (Hr)	0 Gy	0.5 Gy	1 Gy	2 Gy
1	0.675	2.265	4.62	
4	0.15	1.74	3.17	4.48
48	0.17		0.76	1.285

5.2 Generating Calibration Curves and the Model

We go though the same process here as was done in section 3.2. We generate our calibration curves with the Generalized Linear Model function: `glm` in R as using the "training data" by fitting a Poisson generalized linear model with identity link and covariates: nD and n[19]. We do not need to use the Quasi-Poisson model for the response variable here because we do not produce uncertainty estimations here, which we will look at later in section 5.5. The resulting calibration curves are as follows:

2 hour after exposure:

$$Y_2 = 0.1954 + 3.2167D$$

24 hours after exposure:

$$Y_{24} = 0.2355 + 0.8609D.$$

Going through the same process as before, we get our model:

$$Y_{t_1} = 0.191755 + 0.00182273t_1 + D(3.87598e^{-0.3495t_1} + 1.33846e^{-0.01843t_1})$$

$$Y_{t_1+\Delta} = 0.191755 + 0.00182273(t_1+\Delta) + D(3.87598e^{-0.3495(t_1+\Delta)} + 1.33846e^{-0.01843(t_1+\Delta)})$$

5.3 Linear Vs. Quadratic

So far, we have taken for granted that the calibration curve should be linear simply due to the fact that it is what past research have done[12][5][27] as we mentioned in section 3.3. However, as we move to testings with real data, it is useful for us to address the reason why this is the case in order to make sense of our test results.

Define a quadratic calibration curve as

$$Y = a_qD^2 + b_qD + c_q.$$

The only difference between this and the linear calibration curve is the addition of a quadratic term. The argument for using this quadratic calibration curves come from a well documented 'saturation'[30][7] that occurs during γ-H2AX foci counting process. Namely, there is a chance that neighbouring foci will overlap with one another, causing the foci count to be lower than the actual number of foci in a sample. Naturally, this occurs more often when there are more foci in the sample, which is what prompted the use of a quadratic calibration curve. To show case this, we use a plot (figure 5.1) from [12] (which is the example we have been using to show case our model from section 3.3) with consent from the author.

We can see here that in both the $1h$ and $24h$ case, the quadratic curve is higher than the linear curve at a lower dose and lower than the linear curve at a higher dose. This would mean that

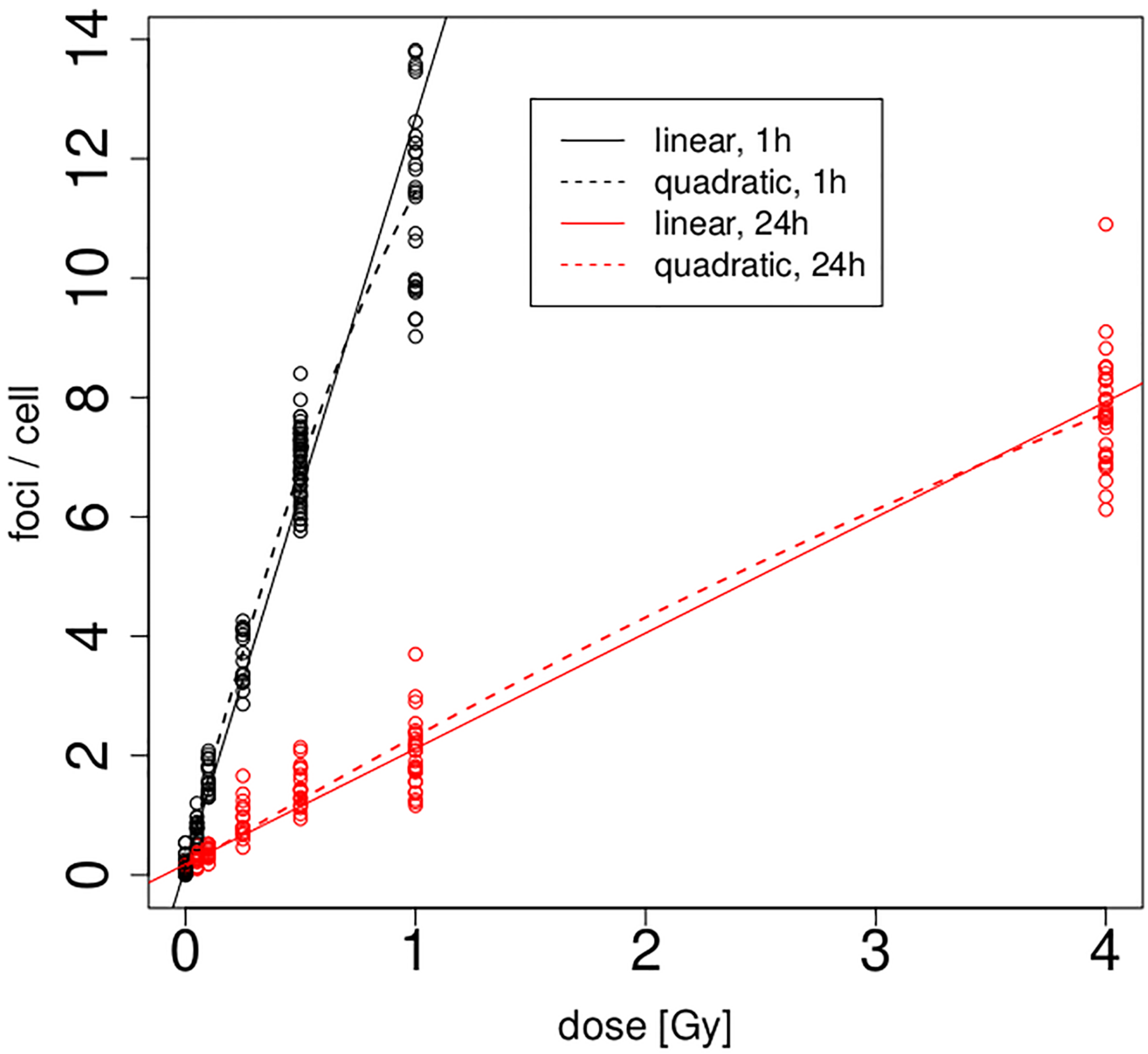


Figure 5.1: example of linear and quadratic calibration curves fitted to average foci counts (Yield) data[12]

when we make estimations with the linear calibration curve, the lower doses would be slightly overestimated while the higher doses would be slightly underestimated.

It seems the quadratic calibration curve captures the 'saturation', and is overall a more acurate representation of the data. Indeed the RSS are 26.045 and 5.9604 for the 2h and 24h respectively for the linear fit while they are 0.50639 and 3.1809 for the quadratic fit. However, the reason for the preference of the linear calibration curve is that the quadratic calibration curve introduces massive Variance to the estimation process[12], while the linear calibration curve introduces only a small bias, which is a worthy sacrifice for much better precision.

We examine our own data with a plot (figure 5.2) of similar settings to the example. We can see that the $2h$ calibration curves follow the same pattern as we discussed. However, the quadratic calibration curve fitted to the $24h$ data is very close to a straight line. This is unsurprising since the number of foci in the $24h$ samples is quite low that the foci overlapping does not happen very often. We would naturally expect the quadratic term (a_q) to be more significant when the average foci count is high but slowly disappears as the average foci count goes to 0. Still, we likely would have been be able to see a more quadratic curve had we more data which would likely improve precision.

The properties of the linear calibration curves will likely be carried over to our model, and will likely effect the results of our own testing in a similar way, that is the lower design doses would be slightly overestimated while the higher design doses would be slightly underestimated. We will briefly illustrate this by looking at the $2h$ calibration curves we generated (figure 5.3). When we have $Y_2 = 2.025$ (the green line), the dose estimation by the quadratic calibration curve is very close to 0.5 while the estimation by the linear quadratic calibration curve is slightly bigger, and when we have $Y_2 = 6.04$ (the blue line), the dose estimation by the quadratic calibration curve is very close to 2 while the estimation by the linear quadratic calibration curve is smaller. This effect will likely be more significant when we use data scored closer to exposure to produce estimations and less significant when we use data scored further to exposure to produce estimations.

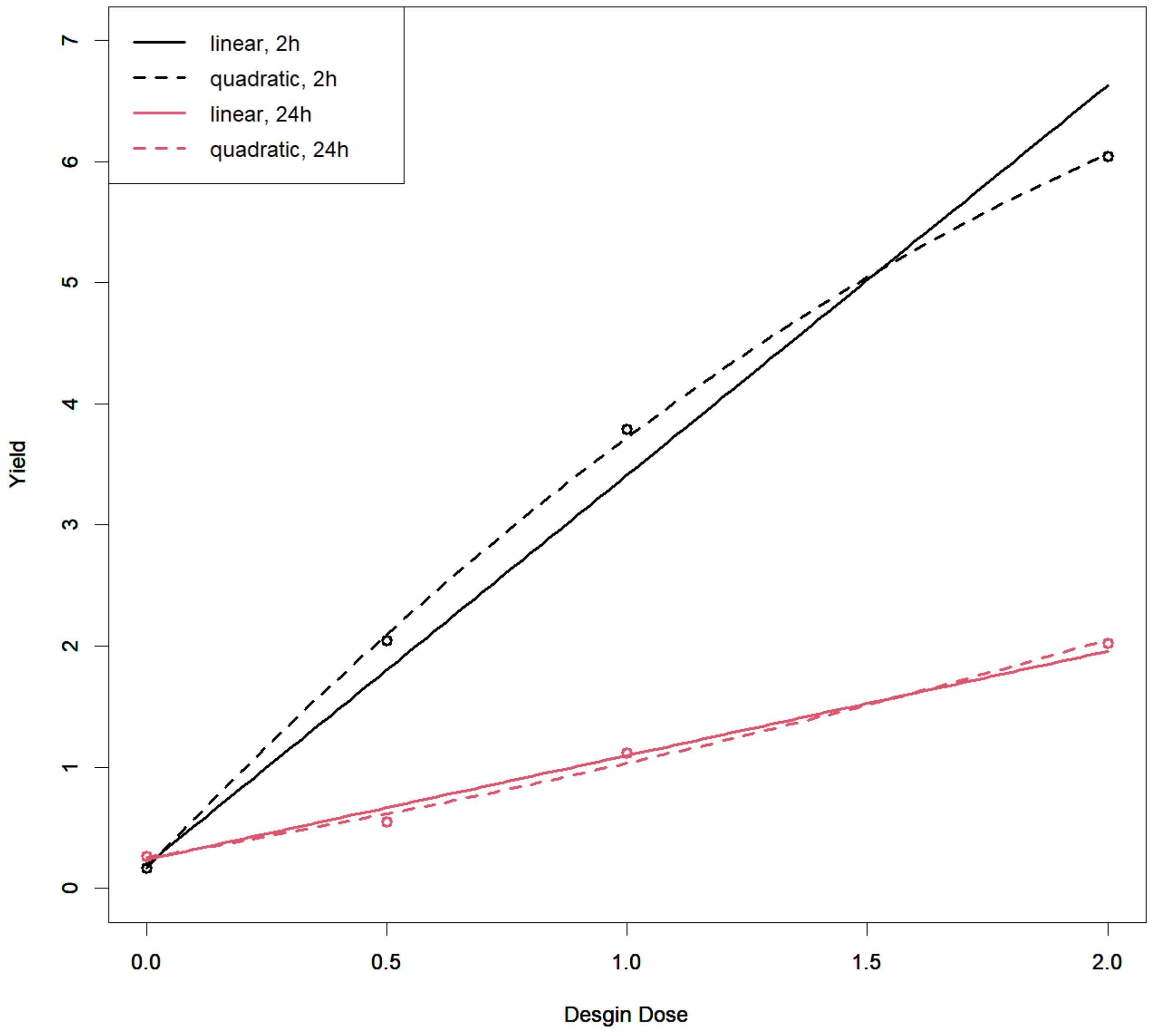


Figure 5.2: linear and quadratic calibration curves fitted to average foci counts (Yield) data

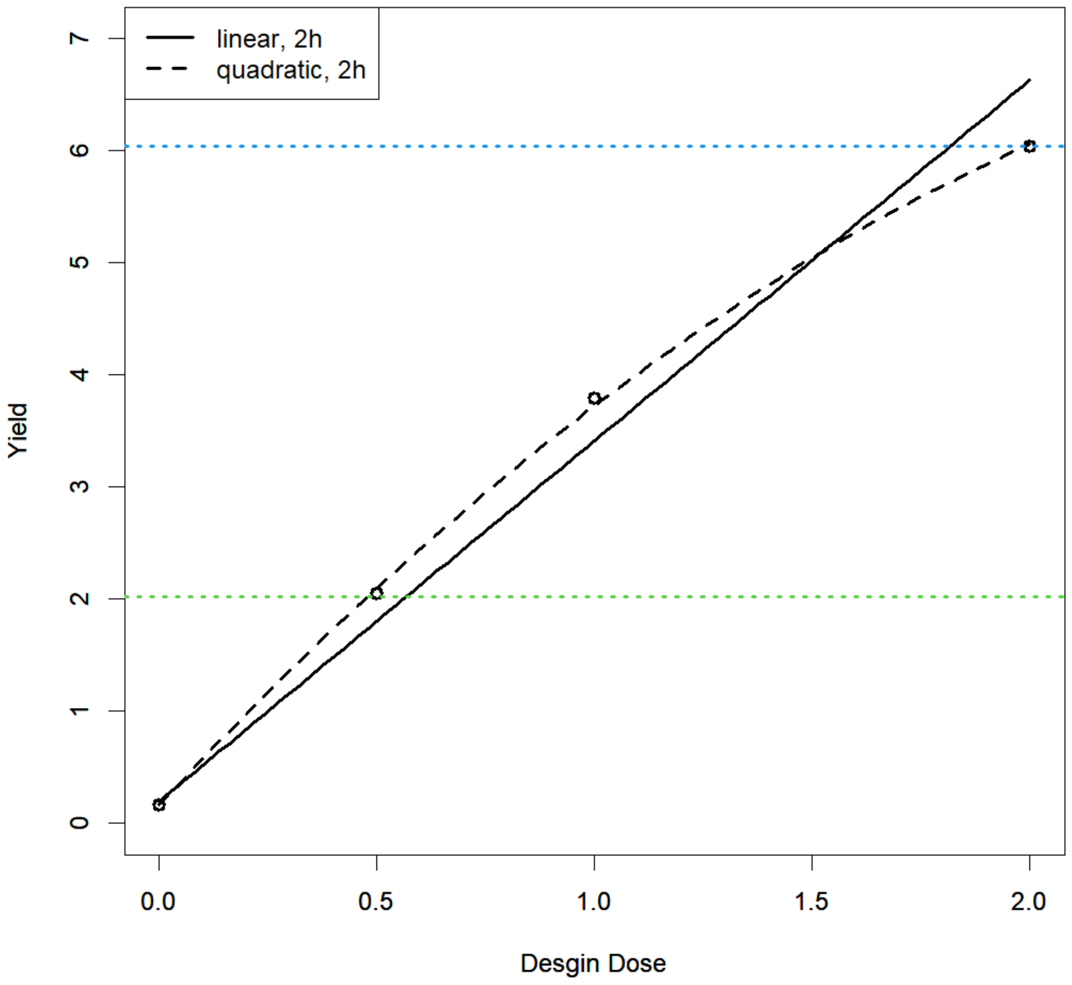


Figure 5.3: linear and quadratic calibration curves fitted to average foci counts (Yield) data

Table 5.6: Dose estimations for the testing data

		design dose (Gy)			
t_1, t_2	Δ	0	0.5	1	2
1, 4	3	0.00	1.12	1.90	
1, 48	37	0.00		0.86	
4, 48	34	0.00		0.84	1.81

5.4 Testing with the Average Foci Counts

5.4.1 Setup

We will first test the model on the average foci counts so as to have a straightforward idea of the performance of the model. For this section we do not distinguish between testing and training data but will test with all average foci counts except the two we removed, namely, the data in table 5.3.

Recall the model we generated for this data:

$$Y_{t_1} = 0.191755 + 0.00182273t_1 + D(3.87598e^{-0.3495t_1} + 1.33846e^{-0.01843t_1})$$

$$Y_{t_1+\Delta} = 0.191755 + 0.00182273(t_1 + \Delta) + D(3.87598e^{-0.3495(t_1+\Delta)} + 1.33846e^{-0.01843(t_1+\Delta)}).$$

For each design dose, we use every pair of t available as t_1 and $t_2 = t_1 + \Delta$ to estimate D and t_1 by substituting each corresponding Yield into the model.

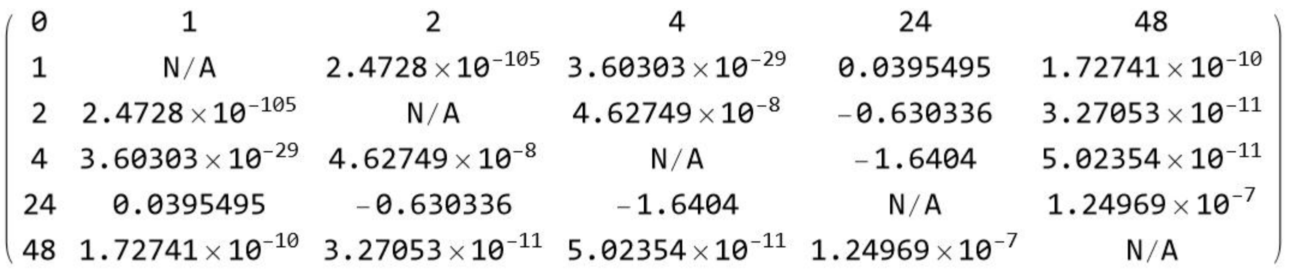
We present the results in matrices where the first row and column of all matrices will denote time points. If we denote one of these matrices R, then R_{ij} is $\hat{D}$ or $\hat{t}$ or the Bias of $\hat{D}$ or $\hat{t}$ at $t_1 = R_{Min(i,j)1} = R_{1Min(i,j)}$ and $t_2 = R_{Max(i,j)1} = R_{1Max(i,j)}$, $i, j > 1$.

5.4.2 Results

We display the dose and time estimations produced from the testing data (table 5.5) in tables 5.6 and 5.7.

Table 5.7: Time estimations for the testing data

		design dose (Gy)			
t_1, t_2	Δ	0	0.5	1	2
1, 4	3	-183	5.20	3.67	
1, 48	37	-58.9		0.04	
4, 48	34	-55.9		1.56	3.56



$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & 2.4728\times10^{-105} & 3.60303\times10^{-29} & 0.0395495 & 1.72741\times10^{-10} \\ 2 & 2.4728\times10^{-105} & \text{N/A} & 4.62749\times10^{-8} & -0.630336 & 3.27053\times10^{-11} \\ 4 & 3.60303\times10^{-29} & 4.62749\times10^{-8} & \text{N/A} & -1.6404 & 5.02354\times10^{-11} \\ 24 & 0.0395495 & -0.630336 & -1.6404 & \text{N/A} & 1.24969\times10^{-7} \\ 48 & 1.72741\times10^{-10} & 3.27053\times10^{-11} & 5.02354\times10^{-11} & 1.24969\times10^{-7} & \text{N/A} \end{pmatrix}$$

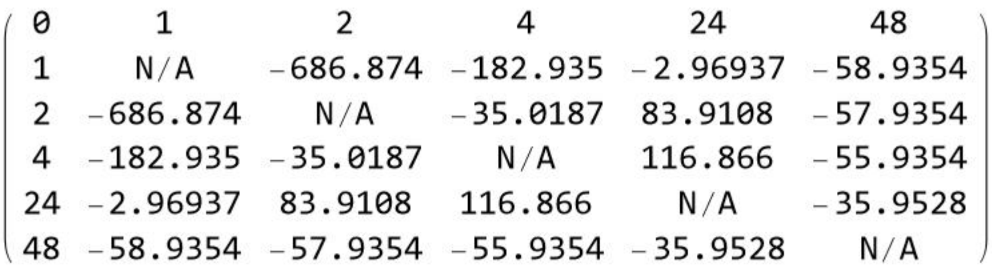
Figure 5.4: $\hat{D}$ and Bias of $\hat{D}$ from real data collected with design dose $D = 0$

Additionally, we display the results for each design dose in complimentary to tables 5.6 and 5.7. We display the dose estimations and its Bias in figure 5.4 and time estimations in figure 5.5 for design dose $D = 0$. Since the value of $\hat{D}$ and the Bias of $\hat{D}$ are the same here, they are displayed in the same figure.

To explain the presentation of data, the first row and column of matrices are time after exposure and the corresponding value to a combination of exposure times is the estimation. For example, the dose estimation generated with the corresponding Yield of $t_1 = 2$, $t_2 = 4$ at design dose $D = 0$, namely, $Y_2 = 0.165$, $Y_4 = 0.15$, is 4.62749×10^{-8}.

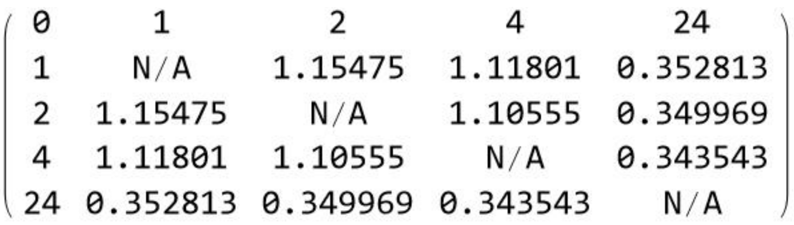
The results seem to indicate that the model is working well. For the dose estimations, we are either getting negative values or positive values very close to 0 with the largest value being 0.04 which is still quite small. The concept of time after exposure makes no sense here since there is no exposure so the time estimations are ignored.

We display the dose and time estimations as well as their Bias for design dose $D = 0.5$ in figures 5.6 to 5.9. The results here generally lines up with the predictions we made in section 5.3. For the dose estimations, there are large overestimations when $t_1, t_2 \leq 4$ and the estimations



$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & -686.874 & -182.935 & -2.96937 & -58.9354 \\ 2 & -686.874 & \text{N/A} & -35.0187 & 83.9108 & -57.9354 \\ 4 & -182.935 & -35.0187 & \text{N/A} & 116.866 & -55.9354 \\ 24 & -2.96937 & 83.9108 & 116.866 & \text{N/A} & -35.9528 \\ 48 & -58.9354 & -57.9354 & -55.9354 & -35.9528 & \text{N/A} \end{pmatrix}$$

Figure 5.5: $\hat{t}$ from real data collected with design dose $D = 0$

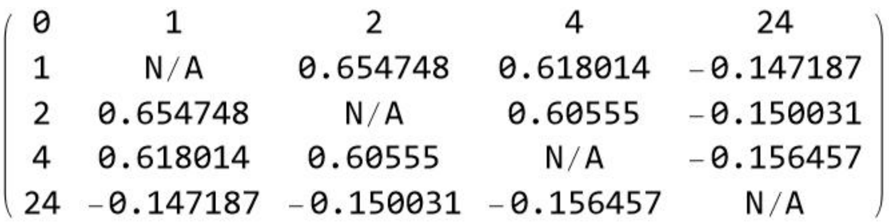


$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 \\ 1 & \text{N/A} & 1.15475 & 1.11801 & 0.352813 \\ 2 & 1.15475 & \text{N/A} & 1.10555 & 0.349969 \\ 4 & 1.11801 & 1.10555 & \text{N/A} & 0.343543 \\ 24 & 0.352813 & 0.349969 & 0.343543 & \text{N/A} \end{pmatrix}$$

Figure 5.6: $\hat{D}$ from real data collected with design dose $D = 0.5$

when $t_2 = 24h$ are quite good. This makes sense since both making estimations of a low design dose and making estimations with data from earlier times contributes to overestimation. It is also worth noting that $n = 200$ being relatively small is also liable to contribute to larger bias in the data. Despite this, if we separate the results into two groups of the results with $t_1, t_2 \leq 4$ and $t_2 = 24h$, we can see that the Variance is quite small at least 'locally'. The time estimations have relatively acceptable bias. Recall that the $48h$ data for $D = 0.5$ was removed and therefore does not feature here.

We display the dose and time estimations as well as their Bias for design dose $D = 1$ in figures 5.10 to 5.13. For the dose estimations, the same overestimation problem with $t_1, t_2 \leq 4$ is occurring here for likely the same reasons as with $D = 0.5$. The estimations seems to be



$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 \\ 1 & \text{N/A} & 0.654748 & 0.618014 & -0.147187 \\ 2 & 0.654748 & \text{N/A} & 0.60555 & -0.150031 \\ 4 & 0.618014 & 0.60555 & \text{N/A} & -0.156457 \\ 24 & -0.147187 & -0.150031 & -0.156457 & \text{N/A} \end{pmatrix}$$

Figure 5.7: Bias of $\hat{D}$ from real data collected with design dose $D = 0.5$

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 \\ 1 & \text{N/A} & 5.45319 & 5.19944 & -0.445586 \\ 2 & 5.45319 & \text{N/A} & 6.04305 & -0.0583648 \\ 4 & 5.19944 & 6.04305 & \text{N/A} & 0.567057 \\ 24 & -0.445586 & -0.0583648 & 0.567057 & \text{N/A} \end{pmatrix}$$

Figure 5.8: $\hat{t}$ from real data collected with design dose $D = 0.5$

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 \\ 1 & \text{N/A} & 4.45319 & 4.19944 & -1.44559 \\ 2 & 4.45319 & \text{N/A} & 4.04305 & -2.05836 \\ 4 & 4.19944 & 4.04305 & \text{N/A} & -3.43294 \\ 24 & -1.44559 & -2.05836 & -3.43294 & \text{N/A} \end{pmatrix}$$

Figure 5.9: Bias of $\hat{t}$ from real data collected with design dose $D = 0.5$

extremely accurate when using $24h$ as t_2, and the model is holding up well when using $48h$ as t_2. The time estimations seem to have an acceptable bias except a relatively big underestimation at $t_1 = 24, t_2 = 48$. This is not surprising as the Yield at $24h$ and $48h$ are likely too close for the model to be effective as simulations in section 4.4 showed. Again, we aim to make quick estimations so $48h$ is not much of our concern.

We display the dose and time estimations as well as their Bias for design dose $D = 2$ in figures 5.14 to 5.17. The dose estimations seem good across the board, but we are not seeing the underestimation when $t_1, t_2 \leq 24$ that we predicted in section 5.3. The only explanation here is that this reflects the Variance introduced by $n = 200$ being too small. The Bias at $t_1 = 24, t_2 = 48$ might be slightly bigger comparing to the rest with likely the same reason

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & 1.32166 & 1.90069 & 1.0227 & 0.857815 \\ 2 & 1.32166 & \text{N/A} & 2.10484 & 1.02038 & 0.855363 \\ 4 & 1.90069 & 2.10484 & \text{N/A} & 0.999597 & 0.83947 \\ 24 & 1.0227 & 1.02038 & 0.999597 & \text{N/A} & 0.696134 \\ 48 & 0.857815 & 0.855363 & 0.83947 & 0.696134 & \text{N/A} \end{pmatrix}$$

Figure 5.10: $\hat{D}$ from real data collected with design dose $D = 1$

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & 0.321661 & 0.900691 & 0.0227034 & -0.142185 \\ 2 & 0.321661 & \text{N/A} & 1.10484 & 0.0203818 & -0.144637 \\ 4 & 0.900691 & 1.10484 & \text{N/A} & -0.00040327 & -0.16053 \\ 24 & 0.0227034 & 0.0203818 & -0.00040327 & \text{N/A} & -0.303866 \\ 48 & -0.142185 & -0.144637 & -0.16053 & -0.303866 & \text{N/A} \end{pmatrix}$$

Figure 5.11: Bias of $\hat{D}$ from real data collected with design dose $D = 1$

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & 1.81741 & 3.66857 & 0.725423 & 0.038155 \\ 2 & 1.81741 & \text{N/A} & 5.84442 & 1.58975 & 0.842803 \\ 4 & 3.66857 & 5.84442 & \text{N/A} & 2.36853 & 1.56385 \\ 24 & 0.725423 & 1.58975 & 2.36853 & \text{N/A} & 8.88262 \\ 48 & 0.038155 & 0.842803 & 1.56385 & 8.88262 & \text{N/A} \end{pmatrix}$$

Figure 5.12: $\hat{t}$ from real data collected with design dose $D = 1$

$$\begin{pmatrix} 0 & 1 & 2 & 4 & 24 & 48 \\ 1 & \text{N/A} & 0.817412 & 2.66857 & -0.274577 & -0.961845 \\ 2 & 0.817412 & \text{N/A} & 3.84442 & -0.410249 & -1.1572 \\ 4 & 2.66857 & 3.84442 & \text{N/A} & -1.63147 & -2.43615 \\ 24 & -0.274577 & -0.410249 & -1.63147 & \text{N/A} & -15.1174 \\ 48 & -0.961845 & -1.1572 & -2.43615 & -15.1174 & \text{N/A} \end{pmatrix}$$

Figure 5.13: Bias of $\hat{t}$ from real data collected with design dose $D = 1$

$$\begin{pmatrix} 0 & 2 & 4 & 24 & 48 \\ 2 & \text{N/A} & 2.33665 & 2.10498 & 1.82029 \\ 4 & 2.33665 & \text{N/A} & 2.09647 & 1.80682 \\ 24 & 2.10498 & 2.09647 & \text{N/A} & 1.42657 \\ 48 & 1.82029 & 1.80682 & 1.42657 & \text{N/A} \end{pmatrix}$$

Figure 5.14: $\hat{D}$ from real data collected with design dose $D = 2$

$$\begin{pmatrix} 0 & 2 & 4 & 24 & 48 \\ 2 & \text{N/A} & 0.336648 & 0.104982 & -0.179705 \\ 4 & 0.336648 & \text{N/A} & 0.0964713 & -0.193177 \\ 24 & 0.104982 & 0.0964713 & \text{N/A} & -0.573426 \\ 48 & -0.179705 & -0.193177 & -0.573426 & \text{N/A} \end{pmatrix}$$

Figure 5.15: Bias of $\hat{D}$ from real data collected with design dose $D = 2$

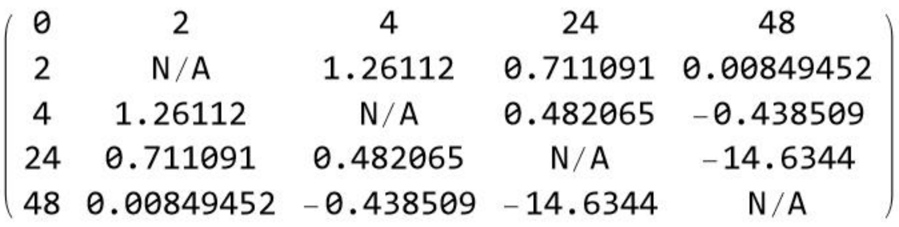
as before, the Yield at later hours being too small. The time estimations are quite accurate comparing to the lower design doses except when $t_1 = 24, t_2 = 48$, which is the same as we had with $D = 1$ with likely the same reason. Recall that the $1h$ data for $D = 2$ was removed and therefore does not feature here.

5.5 Testing with Bootstrap Resampling

We employ the Bootstrap[11] to estimate the uncertainty of the model on real data.

$$\begin{pmatrix} 0 & 2 & 4 & 24 & 48 \\ 2 & \text{N/A} & 3.26112 & 2.71109 & 2.00849 \\ 4 & 3.26112 & \text{N/A} & 4.48206 & 3.56149 \\ 24 & 2.71109 & 4.48206 & \text{N/A} & 9.36558 \\ 48 & 2.00849 & 3.56149 & 9.36558 & \text{N/A} \end{pmatrix}$$

Figure 5.16: $\hat{t}$ from real data collected with design dose $D = 2$



$$\begin{pmatrix} 0 & 2 & 4 & 24 & 48 \\ 2 & N/A & 1.26112 & 0.711091 & 0.00849452 \\ 4 & 1.26112 & N/A & 0.482065 & -0.438509 \\ 24 & 0.711091 & 0.482065 & N/A & -14.6344 \\ 48 & 0.00849452 & -0.438509 & -14.6344 & N/A \end{pmatrix}$$

Figure 5.17: Bias of $\hat{t}$ from real data collected with design dose $D = 2$

5.5.1 Setup

Recall the "testing data" (figure 5.5) from section 5.1 and the dose and time estimations produced from the testing data (table 5.6 and table 5.7).

We acquired this by removing the two abnormal values as well as removing the "training data" that we used to find two calibration curves. For each combination of time and design dose in consideration, we take 1000 Bootstrap samples of its n cells' foci count where each sample consists of n counts that was drawn from the original data with replacement. We take the average of each sample which we use as Yields to produce dose estimations. We present the results with Box Whisker Plots when applicable, as well as the Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$.

5.5.2 Results

We display the relative statistics for design dose $D = 0$ in table 5.8. Box Whisker Plots are not applicable here (the "box" is not visible in the plot) due to the high concentration of the data. The results for $\hat{t}$ are again ignored since it makes no sense when there is no exposure. It seems the model is quite good at detecting no exposure.

Table 5.8: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 0$

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	5.66493×10^{-8}	3.2091×10^{-12}	5.66493×10^{-8}
$t_1 = 1$, $t_2 = 48$	0.0000742503	5.1379×10^{-7}	0.0000742503
$t_1 = 4$, $t_2 = 48$	-0.00131261	0.000194166	0.00131559

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	N/A	1981.31	N/A
$t_1 = 1$, $t_2 = 48$	N/A	351.859	N/A
$t_1 = 4$, $t_2 = 48$	N/A	389.826	N/A

We display the relative statistics for design dose $D = 0.5$ in table 5.9.

Table 5.9: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 0.5$

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	1.94282	1552.39	1.94282

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	5.09965	214.62	5.09995

It might seem like both estimations for $\hat{D}$ and $\hat{t}$ have completely failed from the huge Variance, but this is not entirely true. Due to the high amount of Bootstrap samples we are taking from this relatively small data set with only 200 elements, it is not surprising that a very small amount samples might contain very skewed data, and cause a very small amount of estimations to have extremely large biases that completely mess up the statistics. Bearing this in mind, we have since examined the bootstrap samples and removed 2 extremely large outliers. We present the results from the remaining 998 samples with Box Whisker Plots in figure 5.18, as well as the relative statistics in table 5.10. There is no evidence in literature to back the removal of outliers in bootstrap samples, but doing so provides us with a better idea of what is most likely to happen. After all, the number of 2 outliers is relatively minuscule comparing to the number of 1000 samples. As for the results, we will not focus on the Bias as it comes from the data itself which we have already discussed in section 5.4. The main takeaway here is that the Variance is quite small after removal of outliers so we would most likely be able make good estimations with good calibration curves even when we have only a small n and dose.

Table 5.10: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 0.5$ with 2 outliers removed from 1000 samples

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	0.62664	0.0349509	0.62664

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	4.47408	3.7499	4.47438

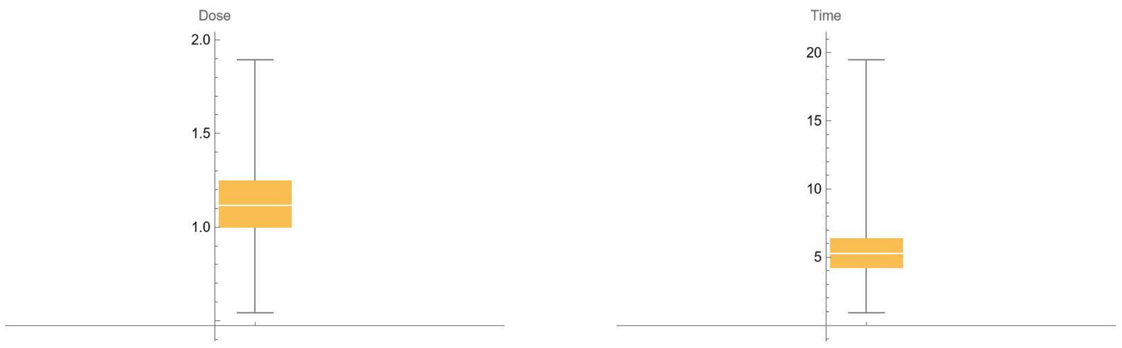


Figure 5.18: $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 0.5$, $t_1 = 1$, $t_2 = 4$ with 2 outliers removed from 1000 samples

We present the results for design dose $D = 1$ with Box Whisker Plots in figures 5.19 to 5.21, as well as the relative statistics in table 5.11.

Table 5.11: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 1$

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	0.923719	0.0618383	0.923719
$t_1 = 1$, $t_2 = 48$	-0.137661	0.0260605	0.176583
$t_1 = 4$, $t_2 = 48$	-0.155851	0.0252143	0.187146

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	2.77891	0.899378	2.77891
$t_1 = 1$, $t_2 = 48$	-0.976156	0.554721	1.04131
$t_1 = 4$, $t_2 = 48$	-2.45576	0.760185	2.45725

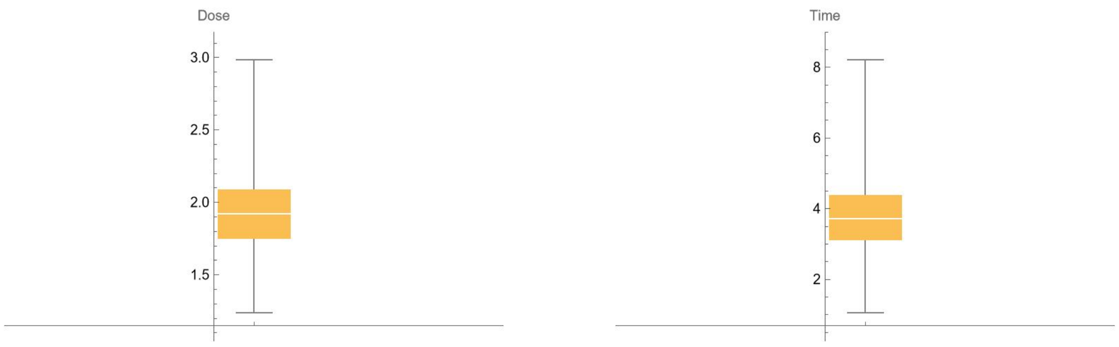


Figure 5.19: $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 1$, $t_1 = 1$, $t_2 = 4$

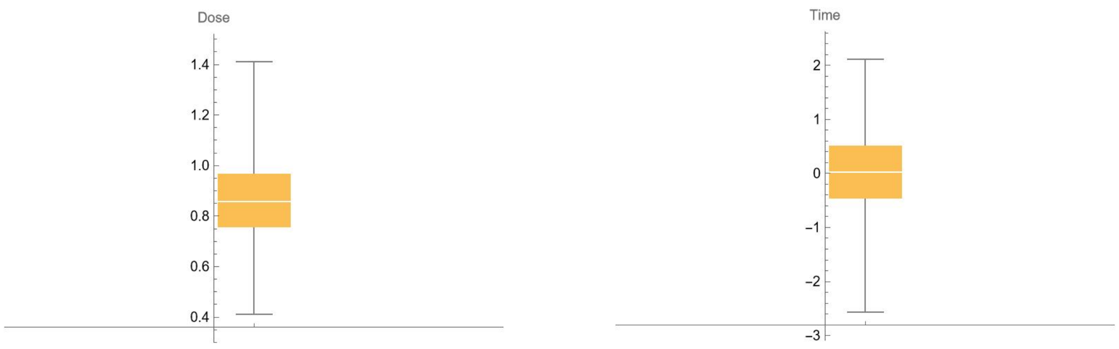


Figure 5.20: $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 1$, $t_1 = 1$, $t_2 = 48$

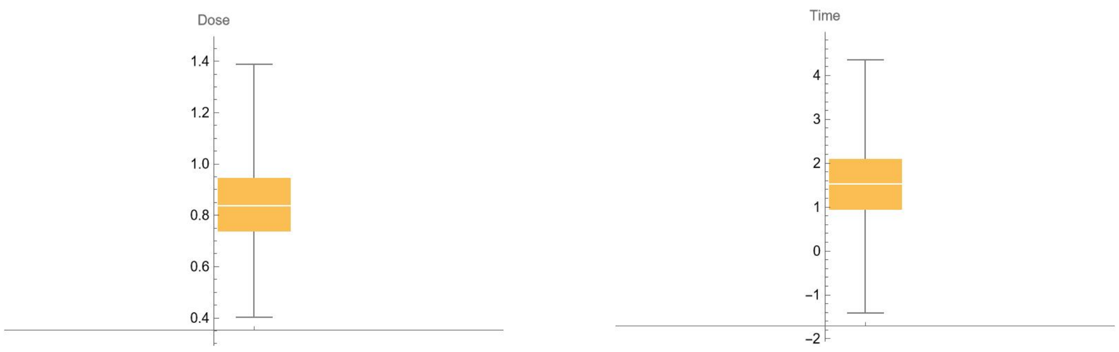


Figure 5.21: $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 1$, $t_1 = 4$, $t_2 = 48$

We present the results for design dose $D = 2$ with Box Whisker Plots in figure5.22, as well as the relative statistics in table 5.12.

Table 5.12: Bias, Variance and Absolute Bias of $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 2$

	$\hat{D}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	-0.189638	0.0585153	0.255243

	$\hat{t}$		
	Bias	Variance	Absolute Bias
$t_1 = 1$, $t_2 = 4$	-0.410802	0.708851	0.762771

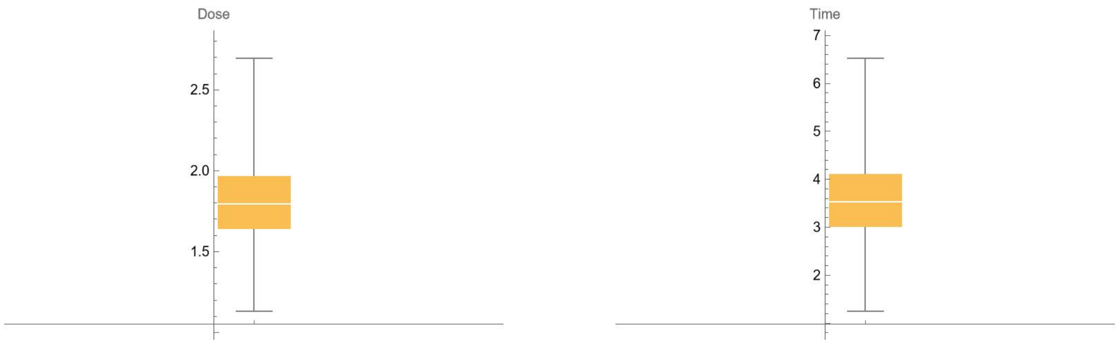


Figure 5.22: $\hat{D}$ and $\hat{t}$ from Bootstrap Simulated Data generated using real data from $D = 2$, $t_1 = 4$, $t_2 = 48$

The results we have in this section overall have inherited the Bias from the data as is to be expected. Our model performs quite well as the Variance is quite low across the board.

Chapter 6

A Complete Guide with examples

In this chapter, we provide a complete guide to performing dose and time estimations under our framework.

6.1 Summary of Estimation

First, we take two existing calibration curves from the laboratory,

$$Y_{t_1} = a_1 D + b_1$$

at t_1 and

$$Y_{t_2} = a_2 D + b_2$$

at t_2, where $a_1, a_2, b_1, b_2, t_1, t_2$ are known, and solve

$$Y_{t_1} = at_1 + b + D(Ae^{ut_1} + Be^{vt_1})$$

and

$$Y_{t_2} = a(t_1 + \Delta) + b + D(Ae^{u(t_1+\Delta)} + Be^{v(t_1+\Delta)})$$

where $u = -0.3495$, $v = -0.01843$, $\Delta = t_2 - t_1$, for a, b, A, B by equating the parameters.

Then, if we know the time after exposure, we take one sample from the patient and to the average foci count Y_t of the sample at t, and solve

$$Y_t = at + b + D(Ae^{ut} + Be^{vt})$$

for dose estimation D to get

$$D = \frac{Y - at - b}{Ae^{ut} + Be^{vt}}.$$

This concludes the estimation.

If we do not know the time after exposure, we take two samples from the patient at two different times that are Δ apart to acquire two average foci counts Y_{t_1}, Y_{t_2} and solve

$$Y_{t_1} = at_1 + b + D(Ae^{ut_1} + Be^{vt_1})$$

and

$$Y_{t_1+\Delta} = a(t_1 + \Delta) + b + D(Ae^{u(t_1+\Delta)} + Be^{v(t_1+\Delta)})$$

for dose estimation D and time estimation t_1, using the `Nsolve` function of Mathematica. The important thing here is to choose a Δ that best suit our needs.

6.2 Choice of Δ

We mentioned in section 4.4 that a bigger Δ leads to a better estimation, but we can get away with a smaller Δ when the dose is high, so we should use any knowledge of the potential time and dose of exposure alongside the activation maps when choosing Δ.

Suppose we count $n = 500$ cells in each sample. For example, suppose a patient comes in and claims to have been exposed to a likely low or no dose of ionizing radiation sometime between the past $1h$ and $3h$. In this case we are forced to use the activation map for $D = 0.5$ (figure 6.1) as this one is the most restrictive of Δ.

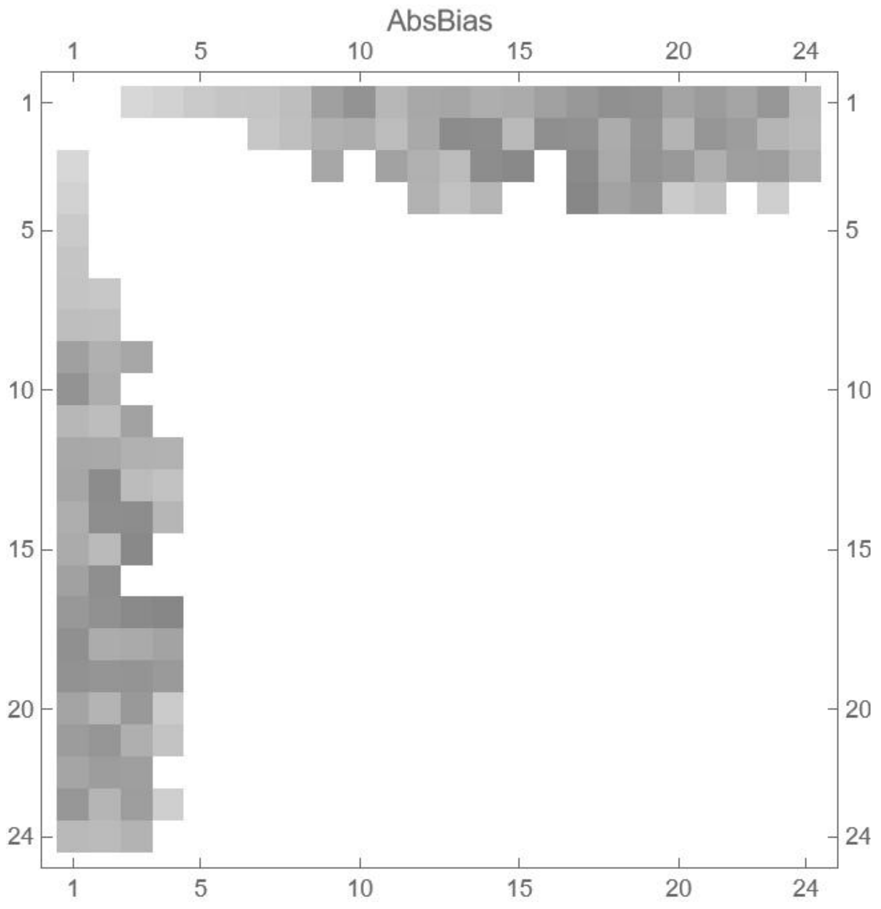


Figure 6.1: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 0.5$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

We can see that when t_1 could be anywhere between 1 and 3, we would likely have reliable estimations if $9 \leq t_2 \leq 24$, so we should take Δ to be at least 8.

Now suppose a patient comes in and claims to have likely been exposed to a likely high dose of ionizing radiation sometime between the past $1h$ and $4h$. In this case we might wish to use the activation map for $D = 2$ (figure 6.2) since the patient might need a more urgent medical treatment so we would like to shorten the time it takes to produce an estimation.

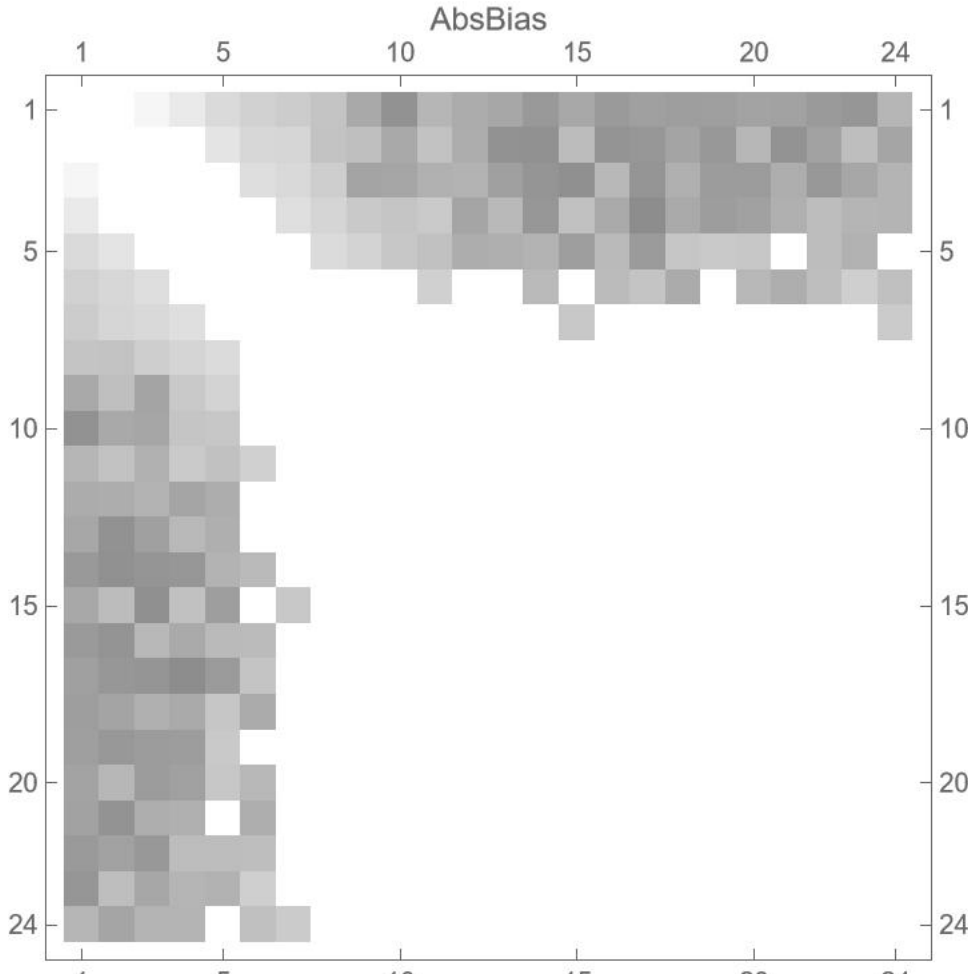


Figure 6.2: Activation Map with cutoff at 1 of $\hat{D}$ from overdispersed (Dispersion parameter 50) Poisson distributed simulation data generated with $D = 2$, $t_1 = 1, 2, ..., 24$, $t_2 = 1, 2, ..., 24$

We can see that when t_1 could be anywhere between 1 and 4, we would likely have reliable estimations if $7 \leq t_2 \leq 24$, so we should take $\Delta = 6$ if we want a quick estimation or perhaps a bigger Δ for a more accurate estimation.

It is worth noting that the activation maps were created using data generated with linear calibration curves in section 4.4, and the 'saturation' was not taken into account. It is not in our power to quantify the 'saturation' due to lack of literature on the subject, and we have to assume that it is lab dependent. In practice, one might be encouraged to keep this in mind, and refer to 5.3 when analyzing data.

Chapter 7

Conclusion

In this project, we aimed to build a model that would allow us to determine ionizing radiation dose and exposure time using a sequence of biomarker measurements. We tested this model using both simulated and real data. We conclude from the results that the model performs well with reasonable discrepancies that can be explained. A study with similar ideas to this project is provided in [20]. This study used MCMC techniques instead of the decay mechanism to form a generalized relationship between dose, time of exposure and Yield.

7.1 Key Contributions

We greatly reduced the time dependency of previous methods of ionizing radiation dose estimation that uses γ-H2AX foci as the biomarker, and we have also sped up the statistical estimation process in many cases.

7.2 Limitations and Possible Future Directions

Recall the Basic Model:

$$Y_t = at + b + D(Ae^{ut} + Be^{vt}).$$

We decided to take u and v from past research and keep them as they are in the modelling process. We discussed in section 3.4 that we cannot determine u and v alongside A and B even if we had more than two calibration curves due to technical difficulties. Future research with ample resources can attempt to generate all four of these parameters and test if this would increase the precision and accuracy of the model.

The overall performance of the models on real data is quite good considering the data we were given have only $n = 200$ foci count at each time point. Future research might adopt perhaps at least $n = 500$ to test the model further. Also, the data we were given have only one sample with each design dose which might reduce the overdispersion of the data and cause an underestimation of Variance in our Bootstrap testings. Future research are encouraged to test the model with more samples.

We have no data between time points $4h$ and $24h$ yet it is our aim to produce the estimation within at least $24h$, so further research might collect foci counts at, for example, time points $8h$, $12h$, $16h$, $20h$ etc. to test the model further. Also, we tested with design dose $D = 4$ in our simulations but we do not have real data with this design dose. Further research might perform this test should they be interested.

One might notice that we did not attempt to verify the activation maps with real data. Since we only have one sample for each design dose, it makes no sense to compare the test results with the activation maps as the lack of inter sample variety means we can only look at the bias, which by itself contribute little to any meaningful inference on the validity of the activation maps.

An intuitive way to describe our model is that it is: a generalization of linear calibration curves using the decay mechanism. We discussed in section 5.3 the pros and cons of linear and quadratic calibration curves. A possible future research might look at a way to generalize quadratic calibration curves to create a model similar to what we did. In order to do this, we would need, in addition to the decay mechanism that describes foci decay, the 'saturation' mechanism that describes how 'saturation' causes the undercount of foci. Research regarding the 'saturation' mechanism is currently lacking.

Appendix A

code

A.1 The Basic Model

```
In[1]:= ClearAll["Global`*"];

In[2]:= Reduce[{a Exp[-0.3495] + b Exp[-0.01843] == 12.559,
  a Exp[-0.3495*24] + b Exp[-0.01843*24] == 1.937}, {a, b}]

Out[2]= a == 13.6222 && b == 3.00975

In[3]:= Reduce[{x + y == 0.131, x + 24 y == 0.179}, {x, y}]

Out[3]= x == 0.128913 && y == 0.00208696

In[9]:= dose = 2;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;
```

```
In[13]:= NSolve[{Yone ==
    0.129 + 0.002087 t +
     d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
   Ytwo == 0.129 + 0.002087 (t + delta) +
     d (13.6222 Exp[-0.3495 (t + delta)] +
        3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals]

During evaluation of In[13]:= NSolve::ratnz: NSolve was unable to solve the system wi

Out[13]= {{d -> 1.99996, t -> 0.999927}}
```

A.2 Testing with Simulated Data

A.2.1 Performance under Overdispersion

```
In[253]:= ClearAll["Global`*"];
SeedRandom[1];

In[255]:= samplen = 100;

In[256]:= biasd1 = {};
biast1 = {};
varianced1 = {};
variancet1 = {};
absbiasd1 = {};
absbiast1 = {};

biasd4 = {};
biast4 = {};
varianced4 = {};
```

```
variancet4 = {};
absbiasd4 = {};
absbiast4 = {};

In[268]:= n = 200;

In[269]:= dose = 1;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist2001 = {};
tlist2001 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
  dlist2001 = Join[dlist2001,  output[[All, 1, 2]]];
  tlist2001 = Join[tlist2001,  output[[All, 2, 2]]]]

dlist2001
tlist2001

biasd = Mean[dlist2001] - dose
biast = Mean[tlist2001] - 1
varianced = Variance[dlist2001]
variancet = Variance[tlist2001]
absbiasd = Mean[Abs[dlist2001 - dose]]
absbiast = Mean[Abs[tlist2001 - 1]]

biasd1 = Append[biasd1, biasd];
biast1 = Append[biast1, biast];
varianced1 = Append[varianced1, varianced];
variancet1 = Append[variancet1, variancet];
absbiasd1 = Append[absbiasd1, absbiasd];
absbiast1 = Append[absbiast1, absbiast];

d2001 = BoxWhiskerChart[dlist2001]
t2001 = BoxWhiskerChart[tlist2001]




During evaluation of In[269]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[269]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[269]:= NSolve::ratnz: NSolve was unable to solve the system v
During evaluation of In[269]:= General::stop: Further output of NSolve::ratnz will be

Out[280]= {1.04711, 0.556323, 0.868238, 1.03268, 0.876049, 0.651934, \
0.756939, 1.4116, 1.0801, 0.693474, 1.16963, 1.18706, 1.52475, \
0.781892, 0.468612, 0.921825, 0.510977, 1.41023, 0.727627, 0.962589, \
0.7262, 0.655679, 1.25328, 1.79953, 0.966227, 0.560911, 1.08894, \
0.776735, 1.01211, 1.33907, 0.876817, 1.00575, 1.47479, 0.254216, \
0.314681, 1.08965, 0.522159, 1.5357, 1.4325, 0.984209, 1.10131, \
0.740808, 0.683355, 0.508314, 0.558155, 0.76386, 0.484884, 0.990556, \
2.04612, 1.0085, 1.07154, 0.684426, 0.138215, 1.42379, 1.35817, \
1.76481, 0.857222, 0.710394, 1.15177, 1.3166, 1.17138, 0.584441, \
0.442658, 0.611031, 1.07175, 0.6436, 0.828579, 1.51974, 0.937306, \
1.05617, 1.29096, 0.77072, 1.0826, 0.612983, 1.06598, 1.0581, \
0.701402, 1.39106, 1.42976, 1.15959, 1.18163, 0.662382, 1.51079, \
0.984215, 1.33031, 0.761605, 1.04553, 1.13463, 0.451122, 1.38247, \
0.362971, 1.30139, 0.540984, 1.34844, 1.19867, 0.797028, 1.24571, \
1.35338, 1.37762, 0.759193}

Out[281]= {1.75976, −0.505255, −0.125362, 0.292788, 0.0500834, \
0.405383, 0.432385, 1.47261, 1.16657, −0.971973, 1.73362, 1.325, \
1.98805, 0.760921, −2.01543, 0.676121, −1.36647, 2.87247, −0.0577065, \
1.41192, −0.557116, 0.0654441, 2.19618, 3.0856, 0.279486, −0.54892, \
1.48779, 0.201458, 1.12954, 2.14071, 0.586884, 1.93735, 2.3853, \
−3.75811, −2.11448, 0.730525, −1.22292, 2.38404, 1.7774, 0.576914, \
1.98699, 0.370504, 0.0473636, −1.38845, −1.3346, −0.914626, −1.53455, \
1.08153, 4.09926, 1.3609, 1.12548, −0.489604, −5.29667, 1.54125, \
4.16103, 3.8565, 0.310859, 0.182332, 0.753827, 3.21307, 0.46785, \
−0.697919, −2.22053, −0.314878, 1.00169, −0.986243, 1.32057, 3.26137, \
0.376557, 1.40798, 1.77346, −0.0514074, 0.76396, −0.367407, 1.92665, \
1.23512, 0.07188, 2.00528, 2.48899, 2.13463, 1.92985, −0.228849, \
3.13001, 2.21021, 1.99541, 0.589749, 1.9571, 1.04715, −2.07132, \
1.97757, −2.16531, 1.88335, 0.00254126, 2.09205, 1.50231, 0.198246, \
1.16118, 2.62365, 2.52736, −0.526176}
```

Out[282]= −0.0216852

Out[283]= −0.253654

Out[284]= 0.133602

Out[285]= 2.61637

Out[286]= 0.29858

Out[287]= 1.25911

```
In[296]:= dose = 4;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 − p);
ntwo = Ytwo*p/(1 − p);

dlist2004 = {};
tlist2004 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist2004 = Join[dlist2004,  output[[All, 1, 2]]];
 tlist2004 = Join[tlist2004,  output[[All, 2, 2]]]]

dlist2004
tlist2004

biasd = Mean[dlist2004] - dose
biast = Mean[tlist2004] - 1
varianced = Variance[dlist2004]
variancet = Variance[tlist2004]
absbiasd = Mean[Abs[dlist2004 - dose]]
absbiast = Mean[Abs[tlist2004 - 1]]

biasd4 = Append[biasd4, biasd];
biast4 = Append[biast4, biast];
varianced4 = Append[varianced4, varianced];
variancet4 = Append[variancet4, variancet];
absbiasd4 = Append[absbiasd4, absbiasd];
absbiast4 = Append[absbiast4, absbiast];
```

```
d2004 = BoxWhiskerChart[dlist2004]
t2004 = BoxWhiskerChart[tlist2004]



During evaluation of In[296]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[296]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[296]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[296]:= General::stop: Further output of NSolve::ratnz will be

Out[307]= {3.62785, 4.40618, 3.97223, 4.15598, 4.31531, 3.42327, \
3.00409, 2.63255, 3.03651, 5.02426, 3.48993, 4.24525, 4.51405, \
4.73781, 3.49936, 4.38643, 4.66737, 4.02638, 2.22318, 3.7118, \
4.60224, 3.64686, 5.53777, 4.2837, 3.021, 4.27464, 4.1961, 2.92229, \
3.91811, 4.09851, 5.42069, 3.30366, 3.40563, 3.89066, 4.28062, \
4.40111, 4.83166, 4.03048, 3.31431, 3.29814, 3.68383, 4.45361, \
2.64463, 3.48135, 3.62122, 3.08033, 3.55893, 4.52944, 4.35405, \
2.81942, 2.70778, 4.13728, 3.91173, 3.529, 5.13033, 3.35955, 6.82997, \
5.32206, 2.94264, 4.17215, 4.16319, 5.09939, 2.88174, 4.00988, \
4.01101, 3.40834, 4.66194, 5.31602, 3.48159, 3.1572, 4.66255, \
4.57477, 5.0295, 4.04345, 4.54365, 5.2075, 3.55677, 3.63754, 3.45567, \
3.88642, 3.80678, 4.79282, 5.23928, 3.66883, 3.94654, 3.22458, \
4.52219, 4.12907, 5.217, 4.69629, 4.01077, 2.87257, 3.05288, 3.99998, \
4.69434, 3.48835, 6.06561, 4.58247, 3.8184, 3.70765}

Out[308]= {0.781199, 1.71987, 1.09249, 0.796108, 1.17842, 0.200248, \
-0.019096, -0.363225, 0.016598, 2.09096, 1.07196, 1.30768, 1.85174, \
1.95011, 0.586017, 1.00369, 2.29088, 1.05613, -1.00448, 0.400674, \
1.68708, 0.495851, 3.12419, 1.23662, 0.0558065, 1.48124, 1.11145, \
0.383723, 1.20128, 0.918471, 2.44399, 0.734835, 0.722855, 0.683916, \
1.4257, 0.962575, 2.14699, 1.28284, 0.410784, 0.438183, 0.181638, \
1.62918, −0.823256, 0.979715, 0.0463536, −0.0140634, 0.36623, \
2.09768, 1.14522, 0.079249, −0.36752, 1.32161, 1.00777, 0.108092, \
1.79507, 0.440062, 3.50782, 2.38533, −0.0772326, 0.805605, 1.35433, \
2.52457, −0.267023, 1.11189, 1.23081, 0.126167, 1.86312, 2.21704, \
0.509977, 0.300841, 1.41884, 1.70232, 2.00612, 1.17967, 1.42817, \
1.88836, 0.372041, 0.77256, 0.347112, 1.19304, 0.767013, 1.4208, \
2.35148, 0.517914, 1.41201, 0.446481, 1.79255, 0.747376, 2.09483, \
1.56666, 1.08925, −0.435667, 0.526261, 0.63615, 1.99408, 0.613475, \
2.47067, 1.45609, 0.606466, 0.195141}

Out[309]= 0.0237173

Out[310]= 0.031199

Out[311]= 0.642673

Out[312]= 0.714097

Out[313]= 0.629045

Out[314]= 0.677035

In[323]:= n = 500;

In[324]:= dose = 1;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;
```

```
dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist5001 = {};
tlist5001 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist5001 = Join[dlist5001, output[[All, 1, 2]]];
 tlist5001 = Join[tlist5001, output[[All, 2, 2]]]]

dlist5001
tlist5001

biasd = Mean[dlist5001] - dose
biast = Mean[tlist5001] - 1
varianced = Variance[dlist5001]
```

```
variancet = Variance[tlist5001]
absbiasd = Mean[Abs[dlist5001 - dose]]
absbiast = Mean[Abs[tlist5001 - 1]]

biasd1 = Append[biasd1, biasd];
biast1 = Append[biast1, biast];
varianced1 = Append[varianced1, varianced];
variancet1 = Append[variancet1, variancet];
absbiasd1 = Append[absbiasd1, absbiasd];
absbiast1 = Append[absbiast1, absbiast];

d5001 = BoxWhiskerChart[dlist5001]
t5001 = BoxWhiskerChart[tlist5001]



During evaluation of In[324]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[324]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[324]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[324]:= General::stop: Further output of NSolve::ratnz will be

Out[335]= {1.33471, 1.19541, 1.26366, 0.716781, 1.10011, 1.09243, \
0.765407, 0.919162, 0.793673, 1.17819, 1.33199, 0.783221, 1.14673, \
1.80649, 1.1508, 1.26702, 1.25837, 1.02105, 0.778372, 1.09256, \
1.1698, 1.12074, 1.0173, 1.01164, 1.06021, 0.839005, 1.68199, \
1.22878, 0.558347, 0.868503, 0.753923, 0.813431, 0.99781, 1.06805, \
0.681372, 0.875033, 0.763298, 0.800088, 1.3728, 0.96095, 1.16607, \
0.835427, 1.21435, 1.23542, 0.775062, 0.749005, 1.12306, 0.963423, \
1.09987, 1.45382, 1.15693, 0.647185, 1.08874, 0.894838, 0.795864, \
1.06628, 0.638481, 0.804178, 1.03561, 0.758684, 0.845533, 0.983841, \
1.29187, 0.903132, 1.21075, 1.08107, 1.16114, 1.09698, 0.861413, \
0.909583, 1.10004, 1.02158, 1.24914, 0.825657, 0.636505, 1.17824, \
0.579387, 1.25582, 1.21852, 1.64001, 0.626578, 1.27069, 1.01868, \
0.913947, 0.608753, 1.06427, 1.207, 1.1399, 1.13392, 1.37318, \
0.945225, 1.09021, 1.26275, 0.83379, 0.988277, 1.2092, 1.43631, \
1.12717, 1.4068, 1.05824}
```

Out[336]=
```
{2.09083, 1.8865, 1.33557, −0.470636, 1.20929, 1.02938, \
0.153613, 0.858249, 0.79481, 2.51896, 2.28627, 0.41252, 1.30867, \
3.81479, 1.45545, 2.59434, 2.25831, 0.575423, 0.283205, 1.03636, \
1.86653, 0.896769, 1.21581, 0.904259, 1.12789, 0.18328, 2.69965, \
1.97125, −0.861852, 0.447064, −0.396367, 0.383367, 1.28603, 1.86957, \
−0.575969, 0.211851, −0.445977, 0.234642, 2.22504, 0.594317, 2.04058, \
0.352812, 1.83985, 1.60795, 0.264716, −0.463256, 1.52644, 0.737783, \
1.30223, 2.01106, 2.46686, −0.146174, 0.893804, 0.115052, 0.00882603, \
1.94257, −0.647849, 0.810337, 1.53437, 0.639496, 0.010303, 1.14085, \
2.10703, 0.95025, 1.38676, 1.62015, 1.20289, 1.26116, 0.318321, \
0.775575, 1.31113, 1.10975, 1.75413, 0.105373, −0.377852, 1.47274, \
−1.04489, 2.2861, 2.27333, 2.83031, −0.728733, 2.38166, 1.80986, \
0.727923, −0.899548, 0.746078, 1.88826, 1.94485, 0.987092, 1.60385, \
1.30749, 1.28619, 1.9852, 0.448783, 1.04031, 1.79623, 2.0662, \
1.42013, 2.73676, 1.24254}
```

Out[337]= 0.0390659

Out[338]= 0.103911

Out[339]= 0.0593413

```
Out[340]= 0.936341

Out[341]= 0.199223

Out[342]= 0.788575

In[351]:= dose = 4;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist5004 = {};
tlist5004 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
         3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist5004 = Join[dlist5004,  output[[All, 1, 2]]];
 tlist5004 = Join[tlist5004,  output[[All, 2, 2]]]]

dlist5004
tlist5004

biasd = Mean[dlist5004] - dose
biast = Mean[tlist5004] - 1
varianced = Variance[dlist5004]
variancet = Variance[tlist5004]
absbiasd = Mean[Abs[dlist5004 - dose]]
absbiast = Mean[Abs[tlist5004 - 1]]

biasd4 = Append[biasd4, biasd];
biast4 = Append[biast4, biast];
varianced4 = Append[varianced4, varianced];
variancet4 = Append[variancet4, variancet];
absbiasd4 = Append[absbiasd4, absbiasd];
absbiast4 = Append[absbiast4, absbiast];

d5004 = BoxWhiskerChart[dlist5004]
t5004 = BoxWhiskerChart[tlist5004]




During evaluation of In[351]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[351]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[351]:= NSolve::ratnz: NSolve was unable to solve the system v
During evaluation of In[351]:= General::stop: Further output of NSolve::ratnz will be

Out[362]= {5.16789, 4.33705, 4.36373, 4.30491, 4.91827, 4.7299, \
3.6863, 3.67068, 3.81071, 3.24145, 3.20537, 3.86248, 3.53377, \
4.93733, 4.52972, 4.1819, 4.31164, 3.7828, 4.01879, 4.47516, 3.64782, \
3.37752, 3.72685, 4.54486, 3.66223, 3.78147, 4.17827, 4.18624, \
4.43599, 4.61445, 3.87987, 4.28114, 3.83664, 4.16696, 4.09779, \
3.37568, 4.02166, 3.48073, 3.84378, 3.87935, 3.46243, 3.79996, \
4.16484, 3.31472, 3.5973, 4.02054, 4.43798, 4.43832, 4.0402, 3.74565, \
3.84551, 3.85203, 3.46075, 3.91056, 4.21356, 4.83488, 4.29261, \
4.11528, 4.21287, 4.55176, 4.33477, 3.73851, 3.94229, 4.51551, \
2.94697, 4.26208, 4.09001, 3.12794, 3.78912, 4.11151, 3.81737, \
3.7481, 3.1752, 3.23963, 4.02899, 3.71985, 3.94163, 4.22574, 4.08285, \
3.71719, 3.79902, 4.0958, 3.53886, 4.38857, 4.11696, 3.86379, \
4.62277, 3.63425, 4.29623, 4.5763, 4.06608, 3.90716, 3.63664, \
3.40984, 3.61703, 3.95039, 4.405, 4.4204, 4.47323, 2.88518}

Out[363]= {2.05554, 1.22934, 1.21317, 1.57951, 1.58591, 1.4509, \
0.637609, 0.768648, 1.09015, 0.120384, 0.0945957, 1.01934, 0.533291, \
1.81638, 1.48504, 1.04904, 1.06874, 0.885949, 1.03768, 1.59322, \
0.614718, 0.296625, 0.792838, 1.62709, 0.795302, 0.794272, 1.09515, \
1.23704, 1.42274, 1.25942, 0.974297, 1.16571, 0.947099, 0.923744, \
1.32835, 0.638983, 0.787982, 0.567359, 0.631495, 1.00983, 0.411914, \
0.678564, 1.32213, −0.00692639, 0.610332, 1.12982, 1.64876, 1.50127, \
0.761171, 0.694221, 0.912639, 0.83149, 0.622573, 0.956371, 1.38683, \
1.66864, 0.973339, 1.17735, 1.43107, 1.59763, 1.18843, 0.900774, \
1.14958, 1.4154, −0.135794, 1.21248, 1.32165, 0.141261, 0.815233, \
1.06093, 1.02401, 0.831882, 0.184341, 0.123773, 0.762494, 0.64817, \
1.1546, 0.989062, 1.07829, 0.521921, 1.27595, 1.19359, 0.409598, \
1.07383, 1.40219, 0.738111, 1.5492, 0.505901, 1.73242, 1.77566, \
0.993732, 1.05126, 0.601043, 0.387903, 0.373087, 0.825438, 1.46371, \
1.54951, 1.30783, −0.0268467}

Out[364]= −0.0134029

Out[365]= −0.018947

Out[366]= 0.198833

Out[367]= 0.209392

Out[368]= 0.358189

Out[369]= 0.364214

In[378]:= n = 1000;

In[379]:= dose = 1;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 − p);
ntwo = Ytwo*p/(1 − p);

dlist10001 = {};
tlist10001 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
     0.129 + 0.002087 t +
      d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
    Ytwogen ==
     0.129 + 0.002087 (t + delta) +
      d (13.6222 Exp[-0.3495 (t + delta)] +
         3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist10001 = Join[dlist10001,  output[[All, 1, 2]]];
 tlist10001 = Join[tlist10001,  output[[All, 2, 2]]]]

dlist10001
tlist10001

biasd = Mean[dlist10001] - dose
biast = Mean[tlist10001] - 1
varianced = Variance[dlist10001]
variancet = Variance[tlist10001]
absbiasd = Mean[Abs[dlist10001 - dose]]
absbiast = Mean[Abs[tlist10001 - 1]]

biasd1 = Append[biasd1, biasd];
biast1 = Append[biast1, biast];
varianced1 = Append[varianced1, varianced];
variancet1 = Append[variancet1, variancet];
absbiasd1 = Append[absbiasd1, absbiasd];
absbiast1 = Append[absbiast1, absbiast];
```

```
d10001 = BoxWhiskerChart[dlist10001]
t10001 = BoxWhiskerChart[tlist10001]



During evaluation of In[379]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[379]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[379]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[379]:= General::stop: Further output of NSolve::ratnz will be

Out[390]= {0.897551, 1.12452, 0.9035, 0.8071, 0.865724, 1.11572, \
1.19914, 1.12277, 0.854436, 0.708617, 0.941195, 0.945669, 1.00256, \
1.13971, 1.04793, 0.850977, 1.05639, 1.20536, 1.2331, 0.84247, \
0.996708, 1.02389, 0.944787, 1.25794, 0.828553, 1.26622, 1.06193, \
0.972666, 0.588149, 1.1014, 0.777907, 0.845645, 0.86711, 0.842102, \
0.8188, 0.947777, 1.22902, 1.03968, 1.33798, 0.804671, 1.21576, \
1.20064, 0.963785, 0.917081, 1.07733, 0.659771, 0.873381, 1.04009, \
1.07144, 1.05406, 1.25327, 1.10952, 0.799532, 0.973816, 1.00871, \
1.01732, 0.903857, 0.810113, 1.04174, 0.88405, 0.943946, 1.13499, \
1.03021, 0.988955, 0.966333, 0.969047, 1.10914, 1.14293, 1.38118, \
0.955791, 1.19932, 0.959389, 0.596198, 0.877166, 0.988945, 0.936612, \
1.00358, 0.536275, 1.29479, 0.790918, 1.17167, 1.01842, 1.26999, \
1.40703, 1.06931, 1.08982, 1.00563, 1.12665, 0.989031, 0.896192, \
1.45315, 0.974529, 0.948484, 1.13811, 0.830301, 0.869508, 1.17865, \
0.977733, 0.731299, 0.885692}

Out[391]= {0.602173, 1.65097, 0.653398, 0.404502, 0.631569, 1.52418, \
1.42745, 0.818967, 0.76622, −0.281721, 0.821274, 0.995268, 0.600791, \
1.57549, 0.697002, 0.571779, 1.33712, 1.57064, 1.9291, 0.21091, \
0.731926, 0.590582, 1.00425, 2.07532, 0.129583, 1.74539, 1.13713, \
0.855802, −1.10915, 1.24858, −0.0439639, 0.0846804, 0.721526, \
0.220277, 0.54233, 0.72392, 1.98203, 1.08166, 1.98592, −0.0480688, \
1.37846, 1.69082, 1.26938, 0.888518, 1.37226, −0.0190601, 0.10635, \
1.29872, 1.74489, 1.15871, 1.24344, 1.60616, 0.374739, 0.597641, \
1.28595, 0.933202, 0.548448, 0.0834453, 1.21994, 0.688443, 0.831897, \
1.52309, 1.12911, 1.52077, 0.698196, 1.12347, 1.37694, 1.58011, \
2.5639, 0.892702, 1.7507, 0.952401, −0.672837, 0.511, 1.19545, \
0.365624, 0.914176, −0.878149, 2.00629, 0.306288, 1.65608, 1.27768, \
1.88782, 2.42981, 1.49674, 1.53327, 0.827785, 1.54903, 1.28862, \
0.421552, 2.36834, 0.992543, 0.735355, 0.990128, 0.11055, 0.478944, \
1.71457, 1.3852, −0.358119, 0.500264}

Out[392]= 0.00129581

Out[393]= −0.0338541

Out[394]= 0.0307607

Out[395]= 0.483603

Out[396]= 0.136299

Out[397]= 0.551874

In[406]:= dose = 4;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;
```

```
dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist10004 = {};
tlist10004 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist10004 = Join[dlist10004, output[[All, 1, 2]]];
 tlist10004 = Join[tlist10004, output[[All, 2, 2]]]]

dlist10004
tlist10004

biasd = Mean[dlist10004] - dose
biast = Mean[tlist10004] - 1
varianced = Variance[dlist10004]
```

```
variancet = Variance[tlist10004]
absbiasd = Mean[Abs[dlist10004 - dose]]
absbiast = Mean[Abs[tlist10004 - 1]]

biasd4 = Append[biasd4, biasd];
biast4 = Append[biast4, biast];
varianced4 = Append[varianced4, varianced];
variancet4 = Append[variancet4, variancet];
absbiasd4 = Append[absbiasd4, absbiasd];
absbiast4 = Append[absbiast4, absbiast];

d10004 = BoxWhiskerChart[dlist10004]
t10004 = BoxWhiskerChart[tlist10004]



During evaluation of In[406]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[406]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[406]:= NSolve::ratnz: NSolve was unable to solve the system v

During evaluation of In[406]:= General::stop: Further output of NSolve::ratnz will be

Out[417]= {3.1052, 3.77269, 4.10355, 3.89901, 3.79595, 3.63426, \
3.86343, 4.11904, 5.02887, 4.54542, 4.21566, 3.59189, 4.05792, \
3.37801, 3.99697, 3.82532, 4.13382, 3.97718, 3.80104, 3.4294, \
3.97919, 3.90557, 3.76892, 3.871, 4.08989, 3.54615, 4.03575, 4.1304, \
4.35455, 4.1459, 4.40557, 3.73718, 3.87197, 4.286, 3.54427, 3.8329, \
3.73957, 3.84368, 3.96993, 3.69554, 3.77314, 3.06137, 4.14608, \
4.04645, 3.64848, 3.62851, 3.48821, 3.67348, 3.73807, 3.5438, \
3.64148, 3.32357, 3.98573, 4.03953, 3.66252, 4.20893, 4.48585, \
3.97286, 3.44995, 4.10704, 3.85543, 3.68805, 3.2427, 4.03905, \
3.92809, 4.07118, 4.48344, 3.90015, 3.57461, 3.59286, 4.27713, \
3.79664, 4.68365, 3.73996, 3.85097, 3.69422, 3.92672, 4.1645, \
4.29994, 3.93736, 4.14624, 3.37455, 3.28587, 4.26476, 4.28981, \
3.87381, 3.86611, 3.8216, 3.91228, 3.75191, 4.51274, 4.10216, 3.8697, \
4.58667, 3.82567, 3.4812, 3.8236, 4.21989, 3.62337, 4.12742}
```

Out[418]= {0.0823825, 0.888558, 0.903627, 0.939553, 0.925976, \

```
0.799929, 0.662217, 1.30061, 1.88702, 1.52898, 0.972628, 0.743487, \
0.954227, 0.505761, 0.870307, 0.803398, 0.955143, 1.04079, 0.636048, \
0.39047, 1.08879, 1.00121, 0.871628, 1.03007, 1.07568, 0.658062, \
1.13937, 1.05799, 1.34276, 1.15807, 1.35029, 0.653998, 0.979184, \
1.3237, 0.467959, 0.974064, 0.798172, 0.815645, 0.998851, 0.882927, \
0.691116, −0.00917503, 1.32797, 1.10292, 0.736974, 0.760729, \
0.353492, 0.749505, 0.557004, 0.460884, 0.747327, 0.371021, 1.02385, \
1.17583, 0.702055, 1.32788, 1.70262, 1.11495, 0.321695, 0.962432, \
0.97165, 0.625791, 0.316184, 1.01167, 1.10418, 1.00641, 1.57976, \
0.983433, 0.447846, 0.370176, 1.19334, 0.805124, 1.5952, 0.833131, \
0.938409, 0.507799, 1.0291, 1.21087, 1.10321, 0.970172, 1.21599, \
0.49963, 0.190163, 1.16158, 1.37146, 0.833809, 1.09822, 0.701234, \
0.909421, 0.666543, 1.40103, 1.16159, 0.983421, 1.56617, 0.858614, \
0.535293, 0.894476, 1.19019, 0.730416, 1.17361}
```

Out[419]= −0.098404

Out[420]= −0.0861307

Out[421]= 0.115453

Out[422]= 0.121286

```
Out[423]= 0.2775

Out[424]= 0.271629

In[433]:= n = 2000;

In[434]:= dose = 1;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist20001 = {};
tlist20001 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
      d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist20001 = Join[dlist20001, output[[All, 1, 2]]];
 tlist20001 = Join[tlist20001, output[[All, 2, 2]]]]

dlist20001
tlist20001

biasd = Mean[dlist20001] - dose
biast = Mean[tlist20001] - 1
varianced = Variance[dlist20001]
variancet = Variance[tlist20001]
absbiasd = Mean[Abs[dlist20001 - dose]]
absbiast = Mean[Abs[tlist20001 - 1]]

biasd1 = Append[biasd1, biasd];
biast1 = Append[biast1, biast];
varianced1 = Append[varianced1, varianced];
variancet1 = Append[variancet1, variancet];
absbiasd1 = Append[absbiasd1, absbiasd];
absbiast1 = Append[absbiast1, absbiast];

d20001 = BoxWhiskerChart[dlist20001]
t20001 = BoxWhiskerChart[tlist20001]




During evaluation of In[434]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[434]:= NSolve::ratnz: NSolve was unable to solve the system w
During evaluation of In[434]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[434]:= General::stop: Further output of NSolve::ratnz will be

Out[445]= {1.11739, 1.20038, 1.03048, 1.00286, 1.04792, 1.03929, \
0.970032, 1.11509, 0.901058, 1.0141, 0.906117, 0.904009, 0.950515, \
1.0666, 0.904988, 0.748001, 1.08968, 1.20807, 1.07023, 0.77151, \
1.0804, 1.10471, 1.0107, 1.09569, 1.07712, 1.05252, 0.829283, \
0.843453, 0.779533, 0.978031, 0.878707, 0.859613, 0.858917, 0.944058, \
0.807774, 0.819325, 0.907367, 1.02273, 0.946852, 0.979788, 0.902115, \
0.939489, 0.975642, 0.952933, 1.03315, 0.952875, 1.02889, 0.993208, \
0.951393, 0.848126, 0.791691, 0.693915, 0.857415, 0.978496, 0.90242, \
0.97356, 0.902002, 0.865644, 1.04559, 1.02014, 1.1945, 1.26652, \
0.942631, 1.03233, 1.10792, 0.954778, 1.13266, 0.887441, 1.07532, \
0.796323, 1.08178, 1.18551, 0.832513, 1.02165, 1.00393, 0.720275, \
0.93233, 0.754357, 0.816369, 0.998071, 0.95846, 0.959335, 0.791984, \
1.02806, 1.02887, 0.859572, 0.95615, 0.864959, 1.15562, 1.14435, \
0.893392, 0.838602, 0.866369, 1.15402, 0.984128, 0.852351, 0.956138, \
1.0352, 1.08548, 1.09737}

Out[446]= {1.56877, 1.53303, 0.961655, 1.04705, 0.901314, 1.28383, \
1.33074, 1.29749, 0.660995, 1.12504, 0.482608, 0.764652, 0.688667, \
1.28679, 0.682032, 0.00965548, 1.32579, 1.70771, 1.21867, 0.191227, \
1.43715, 1.35039, 1.13773, 1.42017, 1.13294, 1.10477, 0.0419089, \
0.191735, 0.348838, 0.692014, 0.397376, 0.367061, 0.454888, 0.474122, \
0.291532, 0.297761, 0.749775, 1.21515, 0.775494, 1.11629, 0.424702, \
0.339675, 0.760921, 0.634855, 0.799785, 0.766942, 0.875681, 1.09983, \
0.861263, 0.450852, 0.0901769, -0.124904, 0.540653, 0.89476, \
0.889827, 1.01103, 0.496892, 0.5101, 1.39876, 0.960458, 1.67258, \
1.89678, 0.662409, 1.13169, 1.51226, 0.503153, 1.28052, 0.433217, \
1.14687, 0.220237, 1.42751, 1.71687, 0.225346, 0.916093, 0.84801, \
0.159974, 0.783238, -0.120567, 0.427655, 1.03901, 0.868035, 1.11536, \
0.183049, 1.01245, 1.04226, 0.430969, 1.04993, 0.152478, 1.65069, \
1.67481, 0.566243, 0.0886872, 0.358677, 1.4856, 0.878073, 0.662541, \
0.702316, 1.06306, 1.28321, 1.2412}

Out[447]= -0.0320878

Out[448]= -0.158604

Out[449]= 0.0142655

Out[450]= 0.223841

Out[451]= 0.100185

Out[452]= 0.41044

In[461]:= dose = 4;
Yone = 0.131 + 12.559*dose;
Ytwo = 0.179 + 1.937*dose;
delta = 23;

dispersion = 50;
p = 1/dispersion;
none = Yone*p/(1 - p);
ntwo = Ytwo*p/(1 - p);

dlist20004 = {};
tlist20004 = {};

For[i = 1, i <= samplen, i++,
 Yonegen =
  Mean[RandomVariate[NegativeBinomialDistribution[none, p], n]];
 Ytwogen =
  Mean[RandomVariate[NegativeBinomialDistribution[ntwo, p], n]];
 output =
  NSolve[{Yonegen ==
      0.129 + 0.002087 t +
       d (13.6222 Exp[-0.3495 t] + 3.00975 Exp[-0.01843 t]),
     Ytwogen ==
      0.129 + 0.002087 (t + delta) +
       d (13.6222 Exp[-0.3495 (t + delta)] +
          3.00975 Exp[-0.01843 (t + delta)])}, {d, t}, Reals];
 dlist20004 = Join[dlist20004,  output[[All, 1, 2]]];
 tlist20004 = Join[tlist20004,  output[[All, 2, 2]]]]

dlist20004
tlist20004

biasd = Mean[dlist20004] - dose
biast = Mean[tlist20004] - 1
varianced = Variance[dlist20004]
variancet = Variance[tlist20004]
absbiasd = Mean[Abs[dlist20004 - dose]]
absbiast = Mean[Abs[tlist20004 - 1]]

biasd4 = Append[biasd4, biasd];
biast4 = Append[biast4, biast];
varianced4 = Append[varianced4, varianced];
variancet4 = Append[variancet4, variancet];
absbiasd4 = Append[absbiasd4, absbiasd];
absbiast4 = Append[absbiast4, absbiast];
```

```
d20004 = BoxWhiskerChart[dlist20004]
t20004 = BoxWhiskerChart[tlist20004]



During evaluation of In[461]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[461]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[461]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[461]:= General::stop: Further output of NSolve::ratnz will be

Out[472]= {4.05578, 3.74161, 4.15498, 3.60189, 3.90769, 4.15181, \
3.90253, 3.94339, 3.49804, 4.15347, 4.23108, 3.95923, 3.89055, \
3.80618, 3.90185, 3.82739, 3.84414, 4.25799, 3.98287, 4.26967, \
3.81063, 4.20638, 4.54635, 4.09386, 3.87082, 3.47955, 4.44806, \
3.9732, 4.02927, 3.77587, 4.31163, 3.78655, 3.46999, 3.76842, \
4.30426, 4.49661, 3.95017, 3.82039, 3.97983, 3.65897, 3.91818, \
3.86888, 3.85252, 3.9159, 4.02945, 3.52636, 3.73841, 4.08089, \
3.50984, 4.29284, 3.51762, 3.75337, 4.00437, 4.11512, 3.93029, \
3.5318, 4.06813, 3.74029, 4.18423, 3.63707, 4.62775, 4.25605, \
3.68394, 4.20525, 4.16391, 3.67302, 4.10287, 3.5123, 3.76502, \
4.11838, 3.9742, 4.28744, 3.87156, 4.13294, 3.67479, 4.07025, \
4.11603, 3.68965, 3.88608, 3.94296, 3.90589, 3.97678, 3.828, 4.19097, \
3.73906, 4.18237, 4.27036, 4.26921, 4.23642, 4.03519, 4.04645, \
3.99529, 3.95, 4.15544, 4.09884, 3.75043, 3.77038, 3.84705, 4.22742, \
3.76249}

Out[473]= {0.925989, 0.717264, 1.18166, 0.701299, 0.895992, 1.11078, \
0.811256, 0.981919, 0.503363, 1.07215, 1.3387, 0.934503, 0.993452, \
0.578812, 1.05293, 0.825247, 0.861312, 1.24916, 0.860786, 1.21015, \
0.849923, 1.10078, 1.35875, 1.10384, 0.988259, 0.466966, 1.41459, \
0.962933, 1.09449, 0.791499, 1.31301, 0.739079, 0.406522, 0.908492, \
1.28594, 1.45371, 0.934007, 0.828087, 0.979204, 0.605414, 1.11724, \
1.02931, 0.941928, 1.03305, 1.05576, 0.541131, 0.836695, 1.0593, \
0.542882, 1.13718, 0.548208, 0.592534, 1.01409, 1.11469, 0.891495, \
0.499652, 1.04442, 0.786428, 1.23778, 0.637496, 1.62505, 1.22466, \
0.65917, 1.09297, 1.16993, 0.784007, 1.09664, 0.634825, 0.84185, \
1.17048, 0.858536, 1.21202, 0.703566, 1.19482, 0.637943, 1.00434, \
0.992161, 0.79025, 1.04213, 0.997075, 0.846711, 0.990201, 0.837355, \
1.36738, 0.746781, 1.06493, 1.09555, 1.25645, 1.21902, 1.02907, \
0.942347, 1.05152, 0.96669, 1.03426, 1.24693, 0.84676, 0.73197, \
0.823128, 1.1118, 0.740286}

Out[474]= -0.0392905

Out[475]= -0.0426494

Out[476]= 0.0620083

Out[477]= 0.0578625

Out[478]= 0.204885

Out[479]= 0.192518

In[488]:= BoxWhiskerChart[{dlist2001, dlist5001, dlist10001,
   dlist20001}, Frame -> False, Axes -> True, AxesLabel -> "Dose",
  ChartStyle -> {Red, Green, Blue, Purple},
  ChartLegends -> {"n=200", "n=500", "n=1000", "n=2000"}]
```

```
BoxWhiskerChart[{tlist2001, tlist5001, tlist10001, tlist20001},
 Frame -> False, Axes -> True, AxesLabel -> "Time",
 ChartStyle -> {Red, Green, Blue, Purple},
 ChartLegends -> {"n=200", "n=500", "n=1000", "n=2000"}]
BoxWhiskerChart[{dlist2004, dlist5004, dlist10004, dlist20004},
 Frame -> False, Axes -> True, AxesLabel -> "Dose",
 ChartStyle -> {Red, Green, Blue, Purple},
 ChartLegends -> {"n=200", "n=500", "n=1000", "n=2000"}]
BoxWhiskerChart[{tlist2004, tlist5004, tlist10004, tlist20004},
 Frame -> False, Axes -> True, AxesLabel -> "Time",
 ChartStyle -> {Red, Green, Blue, Purple},
 ChartLegends -> {"n=200", "n=500", "n=1000", "n=2000"}]

nlist = {200, 500, 1000, 2000};

ListLinePlot[Transpose@{nlist, biasd1}, AxesLabel -> {"n", "Dose"},
 PlotLabel -> "Bias"]
ListLinePlot[Transpose@{nlist, biast1}, AxesLabel -> {"n", "Time"},
 PlotLabel -> "Bias"]
ListLinePlot[Transpose@{nlist, Abs[biasd1]},
 AxesLabel -> {"n", "Dose"}, PlotLabel -> "|Bias|"]
ListLinePlot[Transpose@{nlist, Abs[biast1]},
 AxesLabel -> {"n", "Time"}, PlotLabel -> "|Bias|"]
ListLinePlot[Transpose@{nlist, varianced1}, AxesLabel -> {"n"},
 PlotLabel -> "Variance"]
ListLinePlot[Transpose@{nlist, variancet1}, AxesLabel -> {"n"},
 PlotLabel -> "Variance"]
ListLinePlot[Transpose@{nlist, absbiasd1}, AxesLabel -> {"n", "Dose"},
  PlotLabel -> "AbsBias"]
ListLinePlot[Transpose@{nlist, absbiast1}, AxesLabel -> {"n", "Time"},
  PlotLabel -> "AbsBias"]
```

```
ListLinePlot[Transpose@{nlist, biasd4}, AxesLabel -> {"n", "Dose"},
 PlotLabel -> "Bias"]
ListLinePlot[Transpose@{nlist, biast4}, AxesLabel -> {"n", "Time"},
 PlotLabel -> "Bias"]
ListLinePlot[Transpose@{nlist, Abs[biasd4]},
 AxesLabel -> {"n", "Dose"}, PlotLabel -> "|Bias|"]
ListLinePlot[Transpose@{nlist, Abs[biast4]},
 AxesLabel -> {"n", "Time"}, PlotLabel -> "|Bias|"]
ListLinePlot[Transpose@{nlist, varianced4}, AxesLabel -> {"n"},
 PlotLabel -> "Variance"]
ListLinePlot[Transpose@{nlist, variancet4}, AxesLabel -> {"n"},
 PlotLabel -> "Variance"]
ListLinePlot[Transpose@{nlist, absbiasd4}, AxesLabel -> {"n", "Dose"},
  PlotLabel -> "AbsBias"]
ListLinePlot[Transpose@{nlist, absbiast4}, AxesLabel -> {"n", "Time"},
  PlotLabel -> "AbsBias"]
```

A.3 Testing with Real Data

A.3.1 Generating Calibration Curves and the Model + Linear Vs. Quadratic

```
n <- 200
n_vector <- rep(n,4)
dose <- c(0, 0.5, 1, 2)
ndose <- n*dose

y2 <- c(0.165, 2.045, 3.79, 6.04)
y24 <- c(0.265, 0.545, 1.12, 2.025)
```

```
ny2 <- n*y2
ny24 <- n*y24

ndf2 <- data.frame(ndose,ny2,n_vector)
ndf24 <- data.frame(ndose,ny24,n_vector)

ncc2 <- glm(ny2~ -1 + n_vector + ndose, family = poisson(link="identity"), data=ndf2)
summary(ncc2)

ncc24 <- glm(ny24~ -1 + n_vector + ndose, family = poisson(link="identity"), data=ndf
summary(ncc24)

ndose2 <- n*dose^2

quadcc2 <- glm(ny2~ -1 + n_vector + ndose + ndose2, family = poisson(link="identity")
summary(quadcc2)

quadcc24 <- glm(ny24~ -1 + n_vector + ndose + ndose2, family = poisson(link="identity
summary(quadcc24)

curve(ncc2$coefficients[1] + ncc2$coefficients[2]*x, from=0, to=2, ylim=c(0,7))
par(new=TRUE)
curve(quadcc2$coefficients[1] + quadcc2$coefficients[2]*x + quadcc2$coefficients[3]*x
points(dose,y2)

curve(ncc24$coefficients[1] + ncc24$coefficients[2]*x, from=0, to=2, ylim=c(0,2.5))
par(new=TRUE)
curve(quadcc24$coefficients[1] + quadcc24$coefficients[2]*x + quadcc24$coefficients[3
points(dose,y24)

curve(ncc2$coefficients[1] + ncc2$coefficients[2]*x, from=0, to=2, ylim=c(0,7), lwd=
par(new=TRUE)
curve(quadcc2$coefficients[1] + quadcc2$coefficients[2]*x + quadcc2$coefficients[3]*x
points(dose,y2, lwd=2)
par(new=TRUE)
curve(ncc24$coefficients[1] + ncc24$coefficients[2]*x, from=0, to=2, ylim=c(0,7), col
par(new=TRUE)
curve(quadcc24$coefficients[1] + quadcc24$coefficients[2]*x + quadcc24$coefficients[3
points(dose,y24, col=2, lwd=2)
legend(x="topleft", legend=c("linear,-2h", "quadratic,-2h", "linear,-24h", "quadratic
```

A.3.2 Testing with Bootstrap Resampling

$D = 0.5$

```
In[131]:= ClearAll["Global`*"];

dhalftoneraw = Import["Bootstrap.xlsx", {"Data", 2, Range[29, 48]}];
dhalftone = Cases[Flatten[dhalftoneraw], _Real]

In[134]:=
dhalftfourraw =
  Import["Bootstrap.xlsx", {"Data", 3, Range[29, 48]}];
dhalftfour = Cases[Flatten[dhalftfourraw], _Real]

In[136]:= dhalfonetwo = {};
thalfonetwo = {};

In[138]:= For[l = 1, l <= 1000, l++,
 SeedRandom[l];

 Yone = Mean[RandomChoice[dhalftone, 200]];
 Ytwo = Mean[RandomChoice[dhalftfour, 200]];

 output =
  NSolve[{Yone ==
     0.191755 + 0.00182273 t +
      d (3.87598 Exp[-0.3495 t] + 1.33846 Exp[-0.01843 t]),
    Ytwo == 0.191755 + 0.00182273 (t + 3) +
      d (3.87598 Exp[-0.3495 (t + 3)] +
         1.33846 Exp[-0.01843 (t + 3)])}, {d, t}, Reals];
 AppendTo[dhalfonetwo, output[[All, 1, 2]][[1]]];
 AppendTo[thalfonetwo, output[[All, 2, 2]][[1]]];
 ]

During evaluation of In[138]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[138]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[138]:= NSolve::ratnz: NSolve was unable to solve the system w

During evaluation of In[138]:= General::stop: Further output of NSolve::ratnz will be

In[157]:= Export["dhalfonetwo.mx", {dhalfonetwo}];
Export["thalfonetwo.mx", {thalfonetwo}];

In[377]:= dhalfonetwo = Import["dhalfonetwo.mx"][[1]];
thalfonetwo = Import["thalfonetwo.mx"][[1]];

In[247]:= biasdhalfonetwo = Mean[dhalfonetwo] - 0.5
biasthalfonetwo = Mean[thalfonetwo] - 1
variancedhalfonetwo = Variance[dhalfonetwo]
variancethalfonetwo = Variance[thalfonetwo]
```

```
absbiasdhalfonetwo = Mean[Abs[dhalfonetwo - 0.5]]
absbiasthalfonetwo = Mean[Abs[thalfonetwo - 1]]

BoxWhiskerChart[dhalfonetwo, Frame -> False, Axes -> True,
 AxesLabel -> "Dose"]
BoxWhiskerChart[thalfonetwo, Frame -> False, Axes -> True,
 AxesLabel -> "Time"]

Out[247]= 1.94282


Out[248]= 5.09965


Out[249]= 1552.39


Out[250]= 214.62


Out[251]= 1.94282


Out[252]= 5.09995


In[415]:= Sort[dhalfonetwo]
Sort[thalfonetwo]
dfix = Delete[Delete[Sort[dhalfonetwo], 999], 999];
tfix = Delete[Delete[Sort[thalfonetwo], 999], 999];

In[407]:= biasdhalfonetwofix = Mean[dfix] - 0.5
biasthalfonetwofix = Mean[tfix] - 1
variancedhalfonetwofix = Variance[dfix]
variancethalfonetwofix = Variance[tfix]
absbiasdhalfonetwofix = Mean[Abs[dfix - 0.5]]
absbiasthalfonetwofix = Mean[Abs[tfix - 1]]
```

```
BoxWhiskerChart[dfix, Frame -> False, Axes -> True,
 AxesLabel -> "Dose"]
BoxWhiskerChart[tfix, Frame -> False, Axes -> True,
 AxesLabel -> "Time"]

Out[407]= 0.62664

Out[408]= 4.47408

Out[409]= 0.0349509

Out[410]= 3.7499

Out[411]= 0.62664

Out[412]= 4.47438
```

Bibliography

[1] Biological dosimetry following radiation exposure.

[2] *Cytogenetic Analysis for Radiation Dose Assessment.* Number 405 in Technical Reports Series. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2001.

[3] *Cytogenetic Dosimetry: Applications in Preparedness for and Response to Radiation Emergencies.* Emergency Preparedness and Response. INTERNATIONAL ATOMIC ENERGY AGENCY, Vienna, 2011.

[4] Elizabeth A Ainsbury and David C Lloyd. Dose estimation software for radiation biodosimetry. *Health Phys.*, 98(2):290–295, February 2010.

[5] C Badie, S Kabacik, Y Balagurunathan, N Bernard, M Brengues, G Faggioni, R Greither, F Lista, A Peinnequin, T Poyot, F Herodin, A Missel, B Terbrueggen, F Zenhausern, K Rothkamm, V Meineke, H Braselmann, C Beinke, and M Abend. Laboratory intercomparison of gene expression assays. *Radiat Res*, 180(2):138–148, July 2013.

[6] Mireia Borràs, Gemma Armengol, Martí De Cabo, Joan-Francesc Barquinero, and Leonardo Barrios. Comparison of methods to quantify histone H2AX phosphorylation and its usefulness for prediction of radiosensitivity. *Int. J. Radiat. Biol.*, 91(12):915–924, October 2015.

[7] Rajesh Kumar Chaurasia, N.N. Bhat, Neeraj Gaur, K.B. Shirsath, U.N. Desai, and B.K. Sapra. Establishment and multiparametric-cytogenetic validation of 60co-gamma-ray induced, phospho-gamma-h2ax calibration curve for rapid biodosimetry and triage management during radiological emergencies. *Mutation Research/Genetic Toxicology and Environmental Mutagenesis*, 866:503354, 2021.

[8] Melvin Lee Kiang Chua, Navita Somaiah, Sara Bourne, Frances Daley, Roger A'hern, Otilia Nuta, Sue Davies, Carsten Herskind, Ann Pearson, Jim Warrington, Sarah Helyer, Roger Owen, John Yarnold, and Kai Rothkamm. Inter-individual and inter-cell type variation in residual DNA damage after in vivo irradiation of human skin. *Radiother. Oncol.*, 99(2):225–230, May 2011.

[9] Joanna Deperas, Marta Szluinska, Marta Deperas-Kaminska, Alan Edwards, David Lloyd, Carita Lindholm, Horst Romm, Laurence Roy, Raymond Moss, Josselin Morand, and Andrzej Wojcik. CABAS: a freely available PC program for fitting calibration curves in chromosome aberration dosimetry. *Radiat. Prot. Dosimetry*, 124(2):115–123, 2007.

[10] A A Edwards. The use of chromosomal aberrations in human lymphocytes for biological dosimetry. *Radiat. Res.*, 148(5):S39, November 1997.

[11] Bradley Efron and Robert J. Tibshirani. *An introduction to the bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability.* Chapman and Hall, New York, 1993.

[12] Jochen Einbeck, Elizabeth A Ainsbury, Rachel Sales, Stephen Barnard, Felix Kaestle, and Manuel Higueras. A statistical framework for radiation dose estimation with uncertainty quantification from the γ-H2AX assay. *PLoS One*, 13(11):e0207464, November 2018.

[13] Lynn Hlatky, Rainer K Sachs, Mariel Vazquez, and Michael N Cornforth. Radiation-induced chromosome aberrations: insights gained from biophysical modeling. *Bioessays*, 24(8):714–723, August 2002.

[14] Simon Horn, Stephen Barnard, and Kai Rothkamm. Gamma-H2AX-based dose estimation for whole and partial body radiation exposure. *PLoS One*, 6(9):e25113, September 2011.

[15] Linda J Kuo and Li-Xi Yang. Gamma-H2AX - a novel biomarker for DNA double-strand breaks. *In Vivo*, 22(3):305–309, May 2008.

[16] Halina Lisowska, Aneta Wegierek-Ciuk, Anna Banasik-Nowak, Janusz Braziewicz, Maria Wojewodzka, Andrzej Wojcik, and Anna Lankoff. The dose-response relationship for dicentric chromosomes and γ-H2AX foci in human peripheral blood lymphocytes: influence of temperature during exposure and intra- and inter-individual variability of donors. *Int. J. Radiat. Biol.*, 89(3):191–199, March 2013.

[17] Juan S López, Mònica Pujol-Canadell, Pedro Puig, Montserrat Ribas, Pablo Carrasco, Gemma Armengol, and Joan F Barquinero. Establishment and validation of surface model for biodosimetry based on γ-H2AX foci detection. *Int. J. Radiat. Biol.*, 98(1):1–10, 2022.

[18] N A Maznyk, R C Wilkins, Z Carr, and D C Lloyd. The capacity, capabilities and needs of the WHO BioDoseNet member laboratories. *Radiat. Prot. Dosimetry*, 151(4):611–620, October 2012.

[19] P. McCullagh and J. A. Nelder. *Generalized Linear Models.* Chapman and Hall / CRC, London, 1989.

[20] Dorota Młynarczyk, Pedro Puig, Carmen Armero, Virgilio Gómez-Rubio, Joan F Barquinero, and Mònica Pujol-Canadell. Radiation dose estimation with time-since-exposure uncertainty using the γ-h2ax biomarker. *Sci. Rep.*, 12(1):19877, November 2022.

[21] Jayne Moquet, Stephen Barnard, and Kai Rothkamm. Gamma-H2AX biodosimetry for use in large scale radiation incidents: comparison of a rapid '96 well lyse/fix' protocol with a routine method. *PeerJ*, 2:e282, March 2014.

[22] Jayne Moquet, Stephen Barnard, Albena Staynova, Carita Lindholm, Octávia Monteiro Gil, Vanda Martins, Ute Rößler, Anne Vral, Charlot Vandevoorde, Maria Wojewódzka, and Kai Rothkamm. The second gamma-H2AX assay inter-comparison exercise carried out in the framework of the european biodosimetry network (RENEB). *Int. J. Radiat. Biol.*, 93(1):58–64, January 2017.

[23] Venkatachalam Perumal, Tamizh Selvan Gnana Sekaran, Venkateswarlu Raavi, Safa Abdul Syed Basheerudeen, Karthik Kanagaraj, Amith Roy Chowdhury, and Solomon Fd Paul. Radiation signature on exposed cells: Relevance in dose estimation. *World J. Radiol.*, 7(9):266–278, September 2015.

[24] P Pfeiffer, W Goedecke, and G Obe. Mechanisms of DNA double-strand break repair and their potential to induce chromosomal aberrations. *Mutagenesis*, 15(4):289–302, July 2000.

[25] E P Rogakou, D R Pilch, A H Orr, V S Ivanova, and W M Bonner. DNA double-stranded breaks induce histone H2AX phosphorylation on serine 139. *J. Biol. Chem.*, 273(10):5858–5868, March 1998.

[26] H Romm, E Ainsbury, S Barnard, L Barrios, J F Barquinero, C Beinke, M Deperas, E Gregoire, A Koivistoinen, C Lindholm, J Moquet, U Oestreicher, R Puig, K Rothkamm, S Sommer, H Thierens, V Vandersickel, A Vral, and A Wojcik. Automatic scoring of dicentric chromosomes as a tool in large scale radiation accidents. *Mutat. Res.*, 756(1-2):174–183, August 2013.

[27] Kai Rothkamm, Stephen Barnard, Elizabeth A Ainsbury, Jenna Al-Hafidh, Joan-Francesc Barquinero, Carita Lindholm, Jayne Moquet, Marjo Perälä, Sandrine Roch-Lefèvre, Harry Scherthan, Hubert Thierens, Anne Vral, and Veerle Vandersickel. Manual versus automated γ-H2AX foci analysis across five european laboratories: can this assay be used for rapid biodosimetry in a large scale radiation accident? *Mutat Res*, 756(1-2):170–173, May 2013.

[28] Kai Rothkamm and Simon Horn. gamma-H2AX as protein biomarker for radiation exposure. *Ann Ist Super Sanita*, 45(3):265–271, 2009.

[29] Muriel Viau, Isabelle Testard, Grace Shim, Luc Morat, Marie D Normil, William M Hempel, and Laure Sabatier. Global quantification of γH2AX as a triage tool for the rapid estimation of received dose in the event of accidental radiation exposure. *Mutat. Res. Genet. Toxicol. Environ. Mutagen.*, 793:123–131, November 2015.

[30] Rujira Wanotayan, Sarinya Wongsanit, Kanokporn Boonsirichai, Kasama Sukapirom, Sakchai Buppaungkul, Putthiporn Charoenphun, Pucharee Songprakhon, Kulachart Jangpatarapongsa, and Pimpon Uttayarat. Quantification of histone H2AX phosphorylation in white blood cells induced by ex vivo gamma irradiation of whole blood by both flow cytometry and foci counting as a dose estimation in rapid triage. *PLoS One*, 17(3):e0265643, March 2022.

[31] Jenny Wu, Peter H Clingen, Victoria J Spanswick, Maria Mellinas-Gomez, Tim Meyer, Igor Puzanov, Duncan Jodrell, Daniel Hochhauser, and John A Hartley. γ-H2AX foci formation as a pharmacodynamic marker of DNA damage produced by DNA cross-linking agents: results from 2 phase I clinical trials of SJG-136 (SG2000). *Clin. Cancer Res.*, 19(3):721–730, February 2013.