MONASH University

Advances in Artificial Intelligence for Data Visualization: Developing Computer Vision Models to Automate Reading of Data Plots, with Application to Predictive Model Diagnostics

Weihao Li

B.Comm. (Hons), Monash University

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2022

Department of Econometrics and Business Statistics

Contents

Copyright notice

(Choose one of the following notices.)

(Notice 1)

The second notice certifies the appropriate use of any third-party material in the thesis. Students choosing to deposit their thesis into the restricted access section of the repository are not required to complete Notice 2.

(Notice 2)

I certify that I have made all reasonable efforts to secure copyright permissions for third-party content included in this thesis and have not knowingly added copyright content to my work without the owner's permission.

Abstract

The abstract should outline the main approach and findings of the thesis and must not be more than 500 words.

Declaration

(Standard thesis)

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

(Thesis including published works declaration)

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes (insert number) original papers published in peer reviewed journals and (insert number) submitted publications. The core theme of the thesis is (insert theme). The ideas, development and writing up of all the papers in the thesis were the principal responsibility of myself, the student, working within the (insert name of academic unit) under the supervision of (insert name of supervisor).

(The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.) Remove this paragraph for theses with sole-authored work

In the case of (insert chapter numbers) my contribution to the work involved the following:

Thesis chapter	Publication title	Status (published, in press, accepted or returned for revision)	Nature and % of student contribution	Co-author name(s), nature and % of co-author's contribution	Co-author(s), Monash student Y/N
2	xx	xx	xx	xx	N
3	xx	xx	xx	xx	N
4	xx	xx	xx	xx	N
5	xx	xx	xx	xx	N

I have / have not renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student name: Weihao Li

Student signature:

Date:

Acknowledgements

I would like to thank my pet goldfish for …

Preface

The material in Chapter 1 has been submitted to the journal *Journal of Impossible Results* for possible publication.

The contribution in Chapter **??** of this thesis was presented in the International Symposium on Nonsense held in Dublin, Ireland, in July 2015.

Chapter 1

Introduction

1.1 AI: Four Approaches

Artificial intelligence (AI) is the field of research concerned with understanding and building machines who can demonstrate intelligence. As discussed in Russell and Norvig (2002), historically, there are disagreements among researchers about the definition of intelligence, which is caused by two critical questions:

1. Should AI act and think humanly or rationally?
2. Without the thought process and reasoning, are behaviours sufficient to demonstrate intelligence?

Based on the answer to the above questions, four major approaches to pursue AI have been established. These approaches can be summarized into a two by two table as shown in Figure 1.1, where the row is "Human" vs. "Rational", and the column is "Behaviour" vs. "Thought". Positioning at the top right cell, the **rational agent approach** aims to build agent that perform mathematically perfect acts such that the best expected outcome can always be achieved. In contrast, the **"laws of thought" approach** focus on understanding the logic behind the rationality. Closely related to **cognitive science**, the **cognitive modelling approach** attempts to express theories of human cognition as computer program to mimic the thought process of human. Lastly, the **Turing test approach** is built

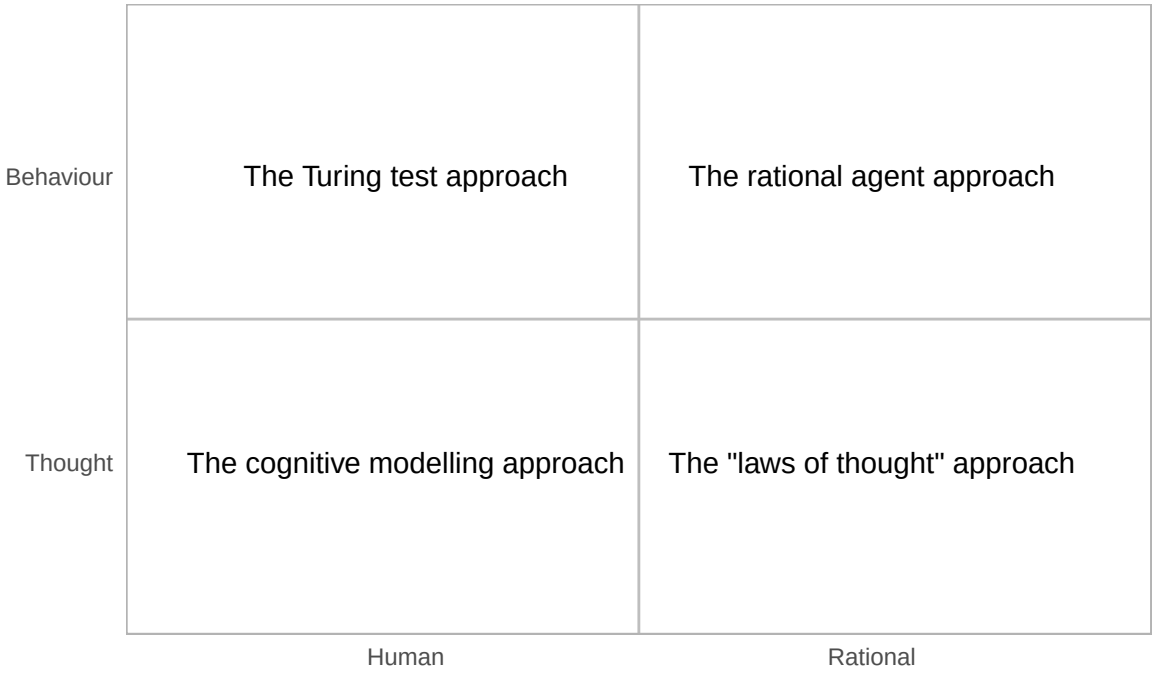


Figure 1.1: *Four possible approaches to pursue AI based on the two dimensions in AI research - human vs. rational and thought vs. behavior (Russell and Norvig, 2002).*

upon the famous **Turing test** proposed by Turing and Haugeland (1950). The test can be roughly described as, whether a human can distinguish another human from a computer with written communications only. To pass the test, several capabilities of computer are required. This includes **natural language processing** for communication with human, **knowledge representation** for encoding knowledge, **automated reasoning** for derivation of conclusions and **machine learning** for improving AI automatically through experience and data. Some researchers argued that written communication is insufficient to demonstrate intelligence, and some degree of physical simulation of a person is still necessary. One such example is the **total Turing test** proposed by Harnad (1991). It adds 3 new requirements to the list, including **computer vision**, **speech recognition** and **robotics**, which are response for interactions with the physical world. Notably, all 7 required capabilities have become major subfields of AI today. And their development has made AI one of the fastest-growing fields in the 21st century (Russell and Norvig, 2002).

With the development of AI, mature AI technologies, such as facial recognition and web recommendation system, have profoundly affected the way modern society operates and

citizen's daily life. This is largely as a consequence of the huge investment in AI industry by the financial market in recent years. Further, the increasingly cheap computing cost and the massive amount of accessible e-commerce data produced in the Internet age provide the possibility for applying data-intensive AI models, which enables AI performance to reach new heights in history (Jordan and Mitchell, 2015). Some AI systems have already been remarkably better than human in certain areas, e.g., game playing. AlphaGo and AlphaZero developed by the Google DeepMind team surpass all human Go players (Silver et al., 2018).

1.2 Predictive Modelling and Visual Diagnostics

Behind the success of AI, a great propotion of AI systems rely on the predictive modelling framework. Donoho (2017) in its summary of data science stated that the concept of this modelling culture could be traced back to an article written by Breiman (2001). In contrast to the generative modelling culture, which aims to develop stochastic models to make inferences about the data generating process, predictive modelling emphasizes the ability of the model to make accurate predictions. Most AI tasks are complex prediction problems where the data mechanism is mysterious, or at least, partly unknowable. Breiman (2001) suggests that generative models are obviously not applicable in these scenarios, while the predictive modelling seeks only an accurate approximated function $f(\boldsymbol{x})$ to describe the relationship between the features $\boldsymbol{x}$ and the responses $\boldsymbol{y}$.

Predictive models are primarily evaluated by predictive accuracy with the use of validation and test data, but in predictive model diagnostics, especially model testing and tuning, data plots play an irreplaceable role. In these diagnostics, though numeric summaries are mostly available and some are even endorsed by finite or asymptotic properties, graphical representation of data is still preferred, or at least needed by researchers, due to its intuitiveness and the possibility to provide unexpected discoveries which may be abstract and unquantifiable.

However, unlike confirmatory data analysis built upon rigorous statistical procedures, e.g., hypothesis testing, visual diagnostics relies on graphical perception - human's ability to interpret and decode the information embedded in the graph (Cleveland and McGill, 1984),

which is to some extent subjective. Further, visual discovery suffers from its unsecured and unconfirmed nature where the degree of the presence of the visual features typically can not be measured quantitatively and objectively, which may lead to over or under-interpretations of the data. One such example is finding a separation between gene groups in a two-dimensional projection from a linear discriminant analysis where there is no difference in the expression levels between the gene groups (Roy Chowdhury et al., 2015).

1.3 Visual Inference

Visual inference was first introduced by Buja et al. (2009) as an inferential framework to extend confirmatory statistics to visual discoveries. This framework redefines the test statistics, tests, null distribution, significance levels and p-value for visual discovery modelled on the confirmatory statistical testing. Figure 4.1 outlines the parallelism between conventional tests and visual discovery.

In visual inference, a visual discovery is defined as a rejection of a null hypothesis, and the same null hypothesis can be rejected by many different visual discoveries (Buja et al.,

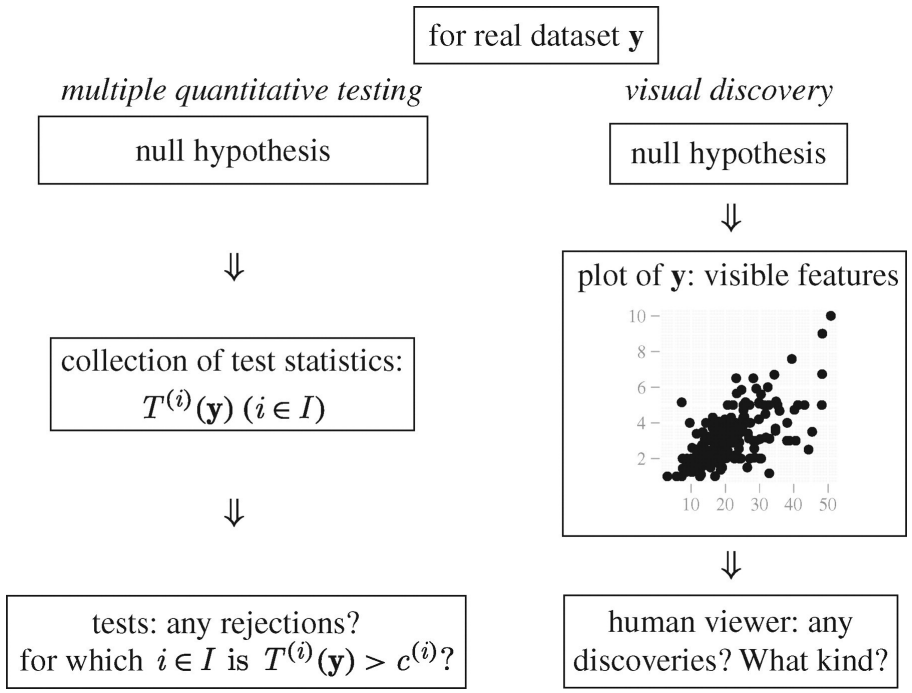


Figure 1.2: *Parallelism between multiple quantitative testing and visual discovery (Buja et al., 2009). Visible features in a plot are viewed as a collection of test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$, and any visual discoveries are treated as evidence against the null hypothesis.*

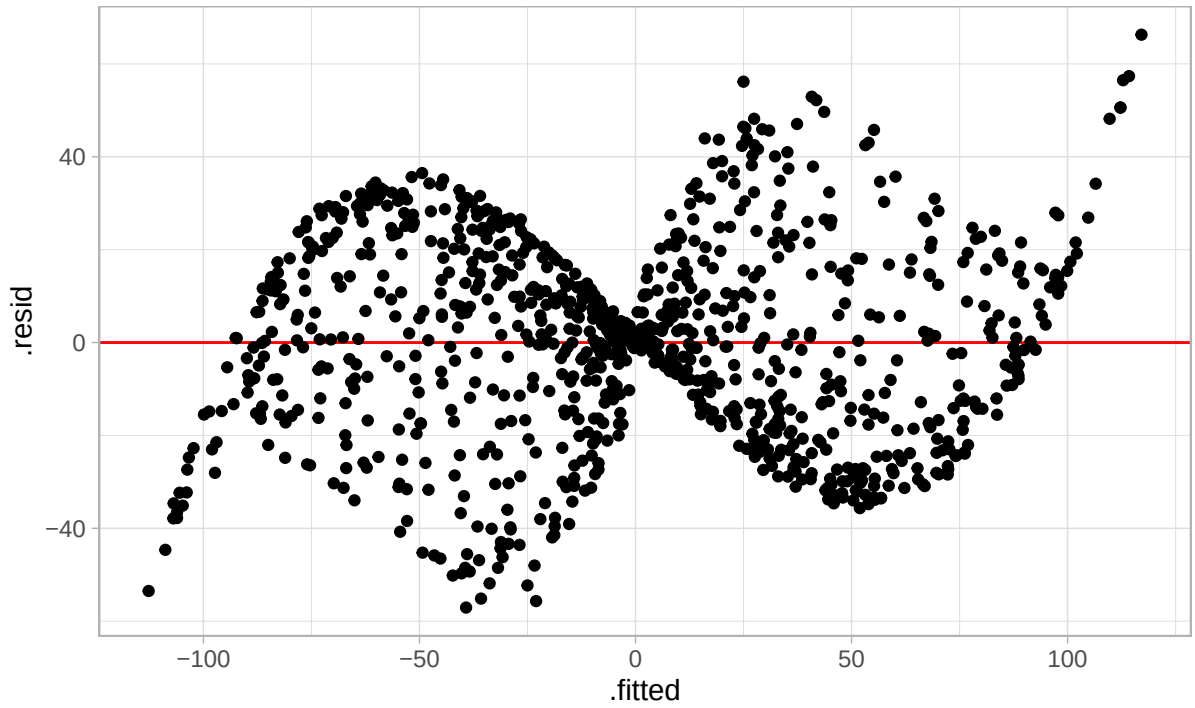


Figure 1.3: *Residuals vs. fitted values plot for a classical linear regression model. The residuals are produced by fitting a two-predictor multiple linear regression model with data generated from a cubic linear model. From the residual plot, "butterfly shape" can be observed which generally would be interpretd as evidence of heteroskedasticity. Further, from the outline of the shape, nonlinear patterns exist. Both visual discoveries are evidence against the null hypothesis, though heteroskedasticity actually does not exist in the data generating process.*

2009). For model diagnostics, the null hypothesis would be the assumed model, while the visual discoveries would be any findings that are inconsistent with the null hypothesis. The same assumed model, such as classical linear regression model, can be rejected by many reasons with residual plot, including nonlinearity and heteroskedasticity as shown in Figure 1.3.

1.4 Pre-specification of Visual Discoverable Features

As discussed in Buja et al. (2009), in the practice of model diagnostics, the range of possible visual discoveries is not pre-specified. In other words, people do not explicitly specify which one or more visual features they are looking for before the read of the diagnostic plot. This is concerning since conventional hypothesis testing always requires the pre-specification of the parameter space Θ of the parameter of interest $\theta \in \Theta$ to form a valid

inferential procedure. To address this issue, a collection of test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$ is defined, where $\mathbf{y}$ is the data and $\boldsymbol{I}$ is a set of all possible visual features. Buja et al. (2009) described each of the test statistics $T^{(i)}(\mathbf{y})$ as a measurement of the degree of presence of a visual feature. Alternatively, Majumder, Hofmann, and Cook (2013) avoids the use of visual features and defined the visual statistics $T(.)$ as a mapping from a dataset to a data plot. Both definitions of visual test statistics are valid, but in the rest of the paper the first definition will be used as it covers some details needed by the following discussion.

The size of the collection $T^{(i)}(\mathbf{y})$ $(i \in I)$ depends on the size of the set I. Thus, if one can define I comprehensively, i.e, pre-specify all the visual discoverable features, the validity issue will be solved. Unfortunately, to our knowledge, there is no such a way to list all visual features. In linear regression diagnostics, possible visual features of a residual plot may be outliers, shapes and clusters. But this is an incomplete list which does not enumerate all the visual features.

Similarly, Wilkinson, Anand, and Grossman (2005) proposed the work called graph theoretic scagnostics, which adopted the idea of "scagnostics" - scatter plot diagnostics from (can't find the 1984 citation). It includes 9 computable scagnostics measures defined on planar proximity graphs: "Outlying", "Convex", "Skinny", "Stringy", "Straight", "Monotonic", "Skewed", "Clumpy" and "Striated" which attempts to describe outliers, shape, density, trend and coherence of the data. This approach is inspiring but it still does not give the complete list of visual discoverable features. In fact, it is possible that such a list will never be complete as suggested in Buja et al. (2009).

Thinking out of the box, Buja et al. (2009) argued that there is actually no need for pre-specification of visual discoverable features. In model diagnostics, when the null hypothesis is rejected, the reasons for rejecting the hypothesis will also be known. This is because observers can not only point out the fact that visual discoveries have been found, but also describe the particular visual features they observed. Those features will correspond to the subset of the collection of visual test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$ which resulted in rejection. This argument helps justifies the validity of visual inference.

1.5 Lineup Protocol

With the validity of visual inference being justified, another aspect of hypothesis testing that needs to be addressed is the control of false positive rate or Type I error. Any visual statistic $T^{(i)}(\mathbf{y})$ needs to pair with a critical value $c^{(i)}$ to form a hypothesis test. When a visual feature i is discovered by the observer from a plot, the corresponding visual statistic $T^{(i)}(\mathbf{y})$ may not be known as there is no general agreement on the measurement of the degree of presence of a visual feature. It is only the event that $T^{(i)}(\mathbf{y}) > c^{(i)}$ is confirmed. Similarly, if any visual discovery is found by the observer, we say, there exists $i \in I : \ T^{(i)}(\mathbf{y}) > c^{(i)}$ (Buja et al., 2009).

Using the above definition, the family-wise Type I error can be controlled if one can provide the collection of critical values $c^{(i)}$ $(i \in I)$ such that $P(\text{there exists } i \in I : T^{(i)}(\mathbf{y}) > c^{(i)}|\mathbf{y}) \leq \alpha$, where α is the significance level. However, since the quantity of $T^{(i)}(\mathbf{y})$ may not be known, such collection of critical values can not be provided.

Buja et al. (2009) proposed the lineup protocol as a visual test to calibrate the Type I error issue without the specification of $c^{(i)}$ $(i \in I)$. It is inspired by the "police lineup" or "identity parade" which is the act of asking the eyewitness to identify criminal suspect from a group of irrelevant people. The protocol consists of m randomly placed data plots, where 1 plot is the actual data plot, and $m - 1$ null plots are produced by plotting data simulate from the null distribution which is consistent with the null hypothesis. Then, an observer who have not seen the actual data plot will be asked to point out the most different plot from the lineup.

Under the null hypothesis, it is expected that the actual data plot would have no distinguishable difference with the null plots, and the probability of the observer correctly picks the actual data plot is $1/m$ due to randomness. If we reject the null hypothesis as the observer correctly picks the actual data plot, then the Type I error of this test is $1/m$.

This provides us with an mechanism to control the Type I error, because m - the number of plots in a lineup can be chosen. A larger value of m will result in a smaller Type I error, but the limit to the value of m depends on the number of plots a human willing to view (Buja

et al., 2009). Typically, m will be set to 20 which is equivalent to set $\alpha = 0.05$, a general choice of significance level for conventional testing among statisticians.

Further, if we involve K independent observers in a visual test, and let X be a random variable denoting the number of observers correctly picking the actual data plot. Then, under the null hypothesis $X \sim \text{Binom}_{K,1/m}$, and therefore, the p-value of a lineup of size m evaluated by K observer is given as

$$P(X \geq x) = \sum_{i=x}^{K} \binom{K}{i} \left(\frac{1}{m}\right)^{i} \left(\frac{m-1}{m}\right)^{k-i}, \tag{1.1}$$

where x is the realization of number of observers correctly picking the actual data plot (Majumder, Hofmann, and Cook, 2013).

The multiple individuals approach avoids the limit of m, while provides visual tests with p-value much smaller than 0.05. In fact, the lower bound of p-value decreases exponentially as K increases. With just 4 individuals and 20 data plots in a lineup, the p-value could be as small as 0.0001.

Compared to the conventional test, whose power only depends on the parameter of interest θ, several studies (see Hofmann et al., 2012; Majumder, Hofmann, and Cook, 2013, 2014; Roy Chowdhury et al., 2015; Loy, Follett, and Hofmann, 2016) have shown the power of the visual test is subject-specific. Thus, to be able to account for individual's ability, an individual is required to evaluate multiple lineups (Majumder, Hofmann, and Cook, 2013).

Assumes that individuals have the same ability and a lineup has been evaluated by multiple individuals, under the alternative hypothesis, the estimated power for a lineup can be expressed as $\hat{p} = x/K$, the estimated probability of identifying the actual data plot from the lineup. If the individual skill needs to be taken into account, and L lienups have been evaluated by K individuals, Majumder, Hofmann, and Cook (2013) suggests that mixed effects logistic regression model can be fit as:

$$g(p_{li}) = W_{li}\delta + Z_{li}\tau_{li},$$

where $g(.)$ is the logit link function $g(p) = log(p) - log(1-p); 0 \leq p \leq 1$. $W_{li}, 1 \leq i \leq K$, $1 \leq l \leq L$, is the covariate matrix including lineup-specific elements and demographic information of individuals, and δ is a vector of parameters. Z is the random effects matrix, and τ is a vector of variables follow $N(\mathbf{0}, \sigma_\tau \boldsymbol{I}_{KL \times KL})$.

Then, the estimated power for lineup l and individual i can be calculated as $\hat{p}_{li} = g^{-1}(W_{li}\hat{\delta} + Z_{li}\hat{\tau}_{li})$ (Majumder, Hofmann, and Cook, 2013).

null data generate mechanism

1.6 Applications of Lineup Protocol

Lineup protocols are

List applications, fields, significance here.

1.7 Limitations of visual tests

1. infeasible in a large scale
2. unfriendly to vision-impaired people
3. high finical cost and human cost
4. time consuming

similar to Handicraft in pre-industrial society

1.8 Automatic visual inference -> Computer vision

relieve people's workload by automating repeating tasks, and provide standard result in a control environment

the use of technology and machinery to enable mass evaluation of visual tests

1.9 Discussion of potential methods

1.9.1 different approaches of AI

- Not aim for understanding the thought process

- Not aim for mimicking the human vision mechanism
- May be able to define distance metrics to measure difference between data plots for making mathematically prefect decisions
- May be able to use computer vision model to approximate how people evaluate lineups

computer vision model:

- Use human data to train model with human selection as target -> mimic the human behaviour
- Use simulated lineup to train model with actual data plot as target -> assume the actual data plot is the most different one
- Use simulated data plot to train model with null or not null as target -> limited to the alternatives given to the model, and Type I error can not be controlled
- Use simulated data plot to train model, then let the model evaluate each plot of a lineup -> leads to multiple selections in a lineup, and the model does not compare different plots in a lineup
  - Use a very large m to ensure the null hypothesis can be rejected by a classifier slightly better than a random selector. The problem becomes how large the m should be
  - Use multiple models as multiple individuals to evaluate lineup -> Which models should be used? How many models should be used?
- Let the model select multiple plots from a simulated lineup while comparing different plots in a lineup -> how to build such a system?
  - The input is m plots, what is the output? The probability of being the most different one?
  - Then how many plots should be selected? Top 5? Or cumulative probability greater than a threshold?
  - How to select the threshold?

Chapter 2

Human Subject Exeriments (toc)

What needs to be discussed?

1. online recruitment platform (Prolific)

- pre-screen
- approve and reject
- redirect
- payment

2. study web site

- PythonAnywhere
- Flask, Python
- jsPsych
- survey
  - age
  - education
  - preferred pronoun
  - previous experience
- training

- 20 lineups
  - setup depends on the study

3. Cubic model

- data generating process
- null model
- potential shapes
- effect size
- conventional test

4. Heteroskedasticity model

- data generating process
- null model
- potential shapes
- effect size
- conventional test

4. Number of participants needed by the study
5. Study 1

- logistic regression
  - easy
  - medium
  - hard
- number of participants
- time
- countries

4. Study 2

- logistic regression
  - easy
  - medium
  - hard
- number of participants
- time
- countries

Chapter 3

Human Subject Exeriments

To collect data of human performance on reading residual plot of linear regression model with nonlinearity and heteroskedasticity defects, two human experiments were conducted. Participants of both experiments were recruited using an online platform called Prolific (Prolific, 2022).

Prolific provides an international participant pool with the option to apply flexible pre-screening filters. In this study, we recruited 62 participants who was fluent in English with at least 10 previous submissions and 98% approval rate in other Prolific studies for quality control. Further, balance sample across gender was imposed to prevent gender bias.

In Prolific, researchers can either approve or reject submissions based on the quality of the responses. If a submission is approved by the researcher, the participant will be paid a certain amount of money per hour of time spent on the experiment. To assess the quality of the responses, two attention checks were given to each participant during the experiment, where at least one of them was required to pass for the approval of submission.

Throughout the experiments, participants were requested to complete a short survey and evaluate 20 lineups on a website in an hour. Each lineup consists of one actual residual plot and 19 null residual plots produced by plotting null residuals simulated from the residual rotation distributions. Among the 20 lineups, there are two extremely easy lineups used as attention checks which everyone should get correct. The short survey was intended to

Table 3.1: *Explanations about reasons for choosing data plots from a lineup*

Reasons	Explanations
outlier(s)	In a data plot, an outlier is a point that differs significantly from other points.
cluster(s)	In a data plot, a cluster is a group of points positioned closely together. And usually, there will be gaps between different clusters.
shape	It could be any common shapes we would see in real life, like triangle shape, U-shape, butterfly shape, etc.
other	Patricipants needs to give their reasons in the text input.

collect information about participant that might affect their ability in reading data plot including age, highest level of education, preferred pronoun and previous experience in similar study. For the evaluation of the lineup, participant first needed to select one or multiple most different data plot(s) from the lineup by clicking the corresponding image. Then, the reason for choosing the plots needed to be provided by selecting one of the given options - "outlier(s)", "cluster(s)", "shape" and "other". Explanations about these options were provided in the training page and the mouseover text. Table 3.1 gives the detailed explanations about these reasons. If the option "other" was selected, text input for the specified reason would be collected. Lastly, the degree of difference between their chosen data plot(s) and other data plots needed to be selected among 5 levels - "not at all", "slightly", "moderately", "very" and "extremely". Notably, if participants could not tell the difference between the data plots, there was an option to skip the evaluation of the lineup. However, to prevent participants abusing this option, warnings were given at the beginning of the study that skipping too many lineups might lead to rejection of submission since it demonstrated clear low-effort throughout the experiment.

The study website was powered by Flask (Grinberg, 2018), a web framework written in Python 3 (Van Rossum and Drake, 2009), hosted on PythonAnywhere (PythonAnywhere, 2022), a web hosting service provider. Due to the limit of the storage on PythonAnywhere, lineup images needed for the website were hosted separately on Github Pages (Gtihub, 2022b), a static site hosting service provided by Github (Gtihub, 2022a), stored in private Github repositories. The uniform resource locator (URL) to lineup images were set to be unique random strings such that images can not be accessed by general approaches without knowing the correct URL. This avoided participants seeing the lineup beforehand. The front-end of the website was built using a JavaScript (Flanagan, 2006) library jsPsych (De Leeuw, 2015) which is specialized in creating online behavioral experiments. One of the reasons we chose to use this library was it had a modularized but highly customizable

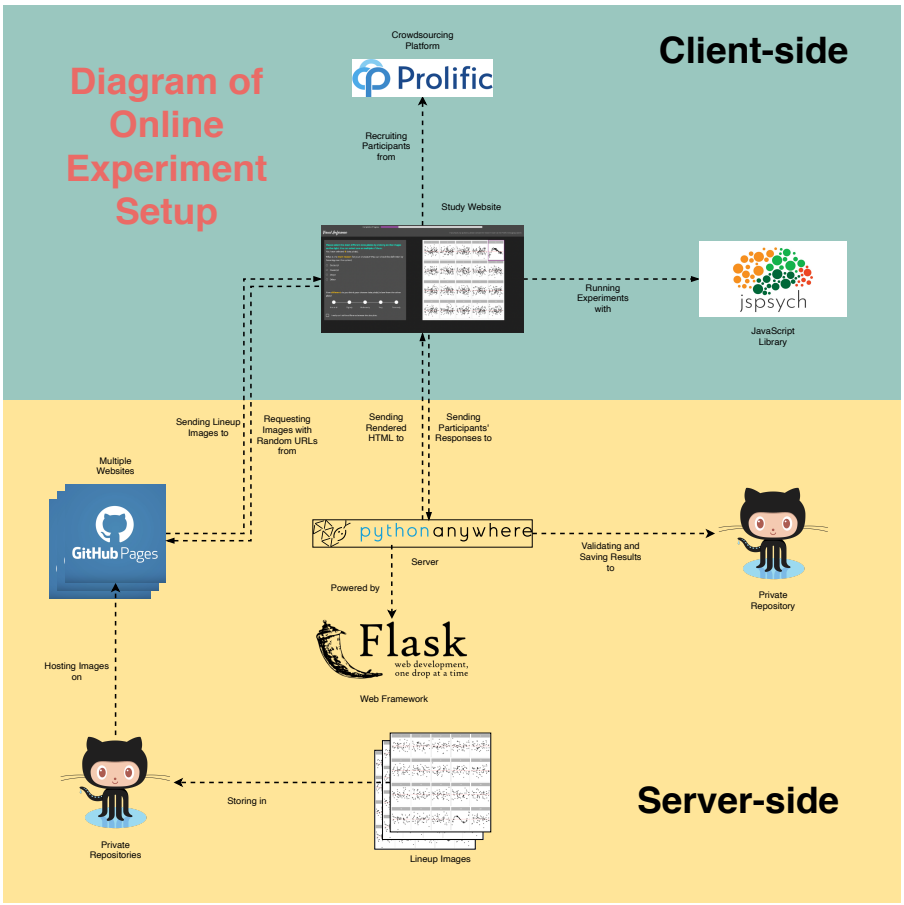


Figure 3.1: *Diagram of online experiment setup. The server-side of the study webiste used Flask as backend hosted on PythonAnywhere. And the client-side used jsPsych to run experiment.*

template which could record participant's response time automatically. This is essential for us to confirm the quality of the data by checking exceptionally fast or slow responses. Figure 3.1 summarizes the online experiment setup.

The first two pages of the website are the explanatory statement and the consent form. Participants needed to read the documents and agree with the terms to advance to the short survey. With the completion of the survey, the next page is the training page, which contained instructions on how to evaluate the lineup, give the reason and choose the confidence level. Followed by the training page, there are 20 lineups each on a single page. Figure 3.2 illustrates the layout of the website. At the end of the experiment, the participants would be redirected back to Prolific and waited for researchers' responses.

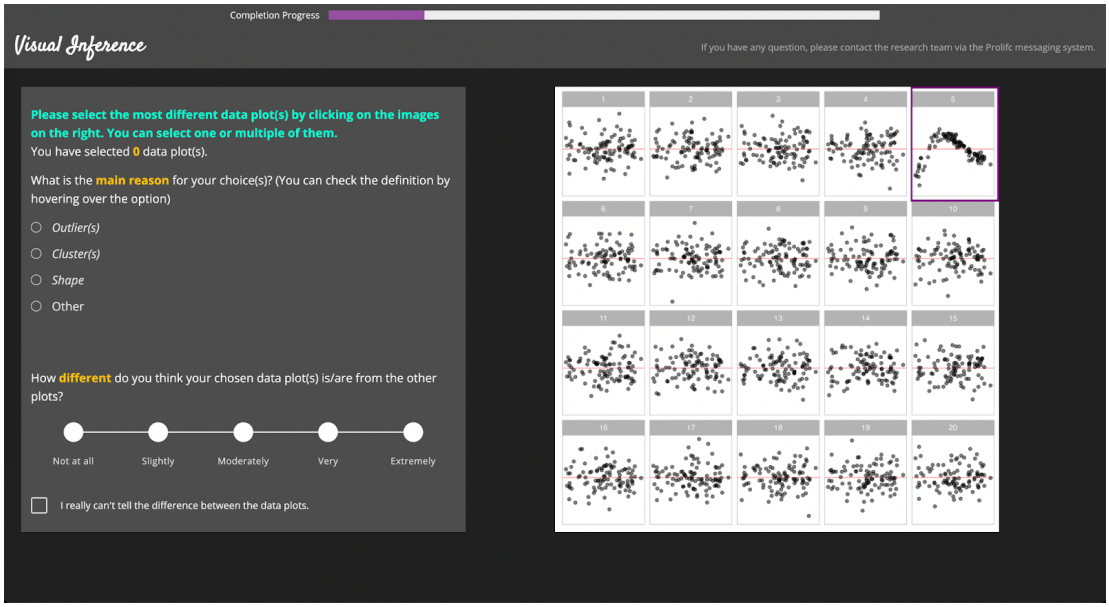


Figure 3.2: *Layout of the study website. Participants needed to choose the most different plots on the right and select their reasons and confidence levels on the left.*

Next we will discuss the simulation setup for this study. The experiment data was simulated by the use of the programming language R (R Core Team, 2021). For the ease of reproducibility, functions to build models, simulate data from models, produce lineup with data, allocate stimulis for subjects and evaluate subject responses from this study are bundled in the package `visage` with a unique object-oriented programming system built upon the environment feature of R. In the description of the simulation, corresponding functionalities of the package will be introduced.

Two models were used in both experiments, which were linear models with some degree of violations of classical assumptions. Residual vs. fitted values plot were used to present the model defects.

3.1 Cubic Model

The first model was a cubic linear model with two regressors, which can be expressed by:

$$\boldsymbol{Y} = 1 + (2-c)\boldsymbol{X} + c\boldsymbol{Z} + a[(2-c)\boldsymbol{X}]^2 + a(c\boldsymbol{Z})^2 + b[(2-c)\boldsymbol{X}]^3 + b(c\boldsymbol{Z})^3 + \boldsymbol{\varepsilon}, \qquad (3.1)$$

where $c \in (0,2)$, $a \in (-3,3)$, $b \in (-3,3)$, $\boldsymbol{\varepsilon} \overset{iid}{\sim} N(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{Z}$ are $n \times 1$ matrices.

This defines a cubic relationship between $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{Z}$. Meanwhile, to create nonlinearity defect, the null model followed the assumptions of the classical normal linear regression model (CNLRM), fitted by OLS is:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X} + \beta_2 \boldsymbol{Z} + \boldsymbol{u}, \tag{3.2}$$

where $\boldsymbol{u} \sim N(0, \sigma_u^2 \boldsymbol{I})$.

Clearly, there will be omitted-variable bias since the null model leaves out the quadric and cubic terms.

Lemma 3.1 (Distribution of residuals produced by the cubic model)**.** *Given the data generating process in Equation (3.1), and null model in Equation (3.2). Let* $\boldsymbol{X}_a = [\boldsymbol{1}, \boldsymbol{X}, \boldsymbol{Z}]$ *denotes the set of regressors in matrix form. Then, the residuals obtained from the null model are*

$$\boldsymbol{e} \sim N(\boldsymbol{R}_a \boldsymbol{X}_b \boldsymbol{\beta}_b, \sigma^2 \boldsymbol{R}_a),$$

where $\boldsymbol{R}_a = \boldsymbol{I} - \boldsymbol{X}_a(\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}\boldsymbol{X}_a'$, $\boldsymbol{X}_b = [\boldsymbol{X}^2, \boldsymbol{Z}^2, \boldsymbol{X}^3, \boldsymbol{Z}^3]$ *and* $\boldsymbol{\beta}_b = (a(2-c)^2, ac^2, b(2-c)^3, bc^3)'$.

Proof. Using the Frisch–Waugh–Lovell theorem, the residuals obtained by the null model are

$$\boldsymbol{e} = \boldsymbol{R}_a \boldsymbol{Y} = \boldsymbol{R}_a(\boldsymbol{X}_a \boldsymbol{\beta}_a + \boldsymbol{X}_b \boldsymbol{\beta}_b + \boldsymbol{\varepsilon}),$$

where $\boldsymbol{R}_a = \boldsymbol{I} - \boldsymbol{X}_a(\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}\boldsymbol{X}_a'$, $\boldsymbol{\beta}_a = (1, 2-c, c)'$, $\boldsymbol{X}_b = [\boldsymbol{X}^2, \boldsymbol{Z}^2, \boldsymbol{X}^3, \boldsymbol{Z}^3]$ and $\boldsymbol{\beta}_b = (a(2-c)^2, ac^2, b(2-c)^3, bc^3)'$.

Because $\boldsymbol{R}_a \boldsymbol{X}_a = \boldsymbol{0}$, we have $\boldsymbol{e} = \boldsymbol{R}_a(\boldsymbol{X}_b \boldsymbol{\beta}_b + \boldsymbol{\varepsilon})$. Since $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 \boldsymbol{I})$, it follows that $\boldsymbol{e} \sim N(\boldsymbol{R}_a \boldsymbol{X}_b \boldsymbol{\beta}_b, \sigma^2 \boldsymbol{R}_a)$. □

Lemma 3.1 shows that the expectation of the residuals is clearly a function of $\boldsymbol{X}$ and $\boldsymbol{Z}$. Hence, we would expect to see some patterns in the residual plot. Let X_i and Z_i, $i = 1, ..., n$, be independent random variables follow uniform distribution $U(-1, 1)$. Given the expectation of the residuals, we could plot the expected values of residuals against the

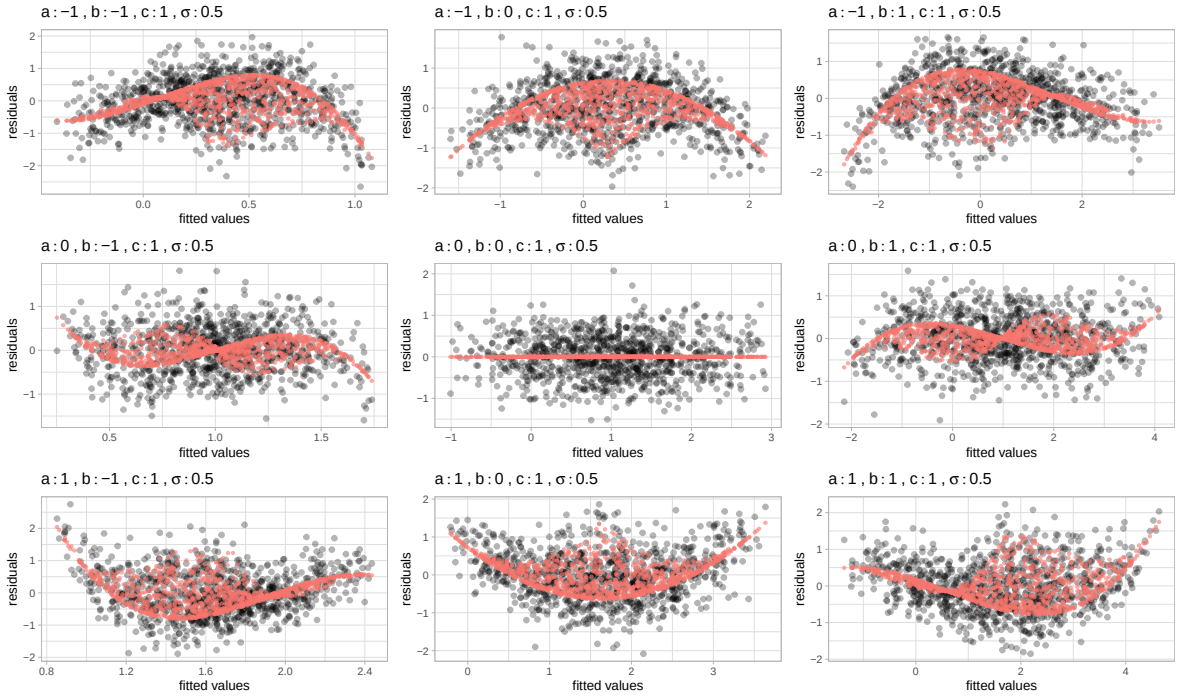


Figure 3.3: *cubic 1*

observed values. Figure 3.3, 3.4 and 3.5 illustrate the shape of residuals and their expected values under different parameter settings.

From the Figure 3.3, it can be observed that with fixed σ and c, a and b are controlling the 2D projection of a hypersurface, and seemingly performing some rotations along different axes. Figure 3.4 shows that with fixed a, b and σ, c is controlling the contribution of $\boldsymbol{X}$ and $\boldsymbol{Z}$ to $\boldsymbol{Y}$. If we move c toward 0 or 2, one variable will donminate another, which will mitigate some joint effects and resemble a typical cubic function. In Figure 3.5, a, b and c is held constant, and σ is controlling the noises around the expected values. As σ decreases, the underlying shape shows up.

The residuals used in these three figures are simulated from the cubic models built using the `cubic_model()` function from the package `visage`. `cubic_model()` is a cubic model class constructor, which takes arguments `a`, `b`, `c`, `sigma`, `x` and `z`, where the first four are numeric values defined above, and `x` and `z` are random variable instances created by the random variable abstract base class constructor `rand_var()`.

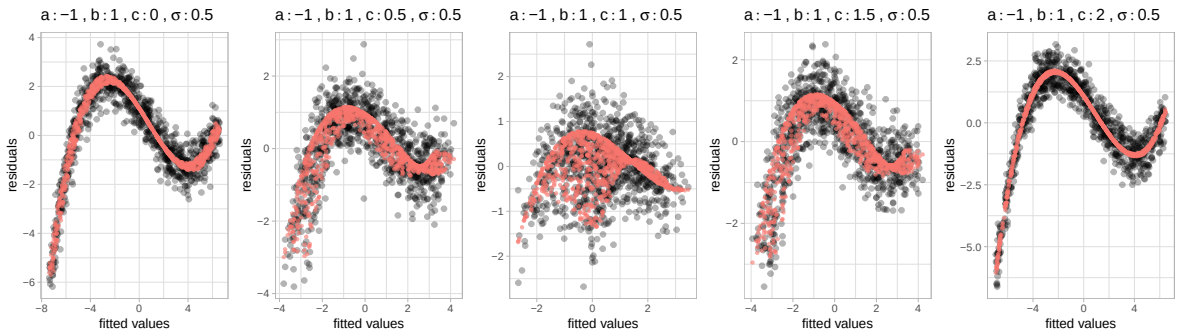


Figure 3.4: *cubic 2*

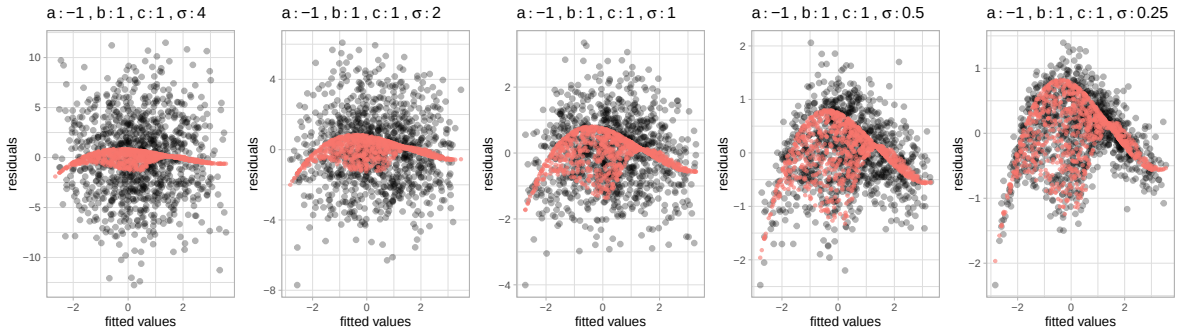


Figure 3.5: *cubic 3*

Here, we set $\boldsymbol{X}$ and $\boldsymbol{Z}$ to be random uniform variables ranged from -1 to 1. This can be done by using the random uniform variable class constructor `rand_uniform()` inherited from the random variable abstract base class. It only takes two arguments which are the lower bound and the upper bound of the support.

```
library(visage)
mod <- cubic_model(a = -3, b = -3, c = 1, sigma = 0.5,
                   x = rand_uniform(-1, 1), z = rand_uniform(-1, 1))
```

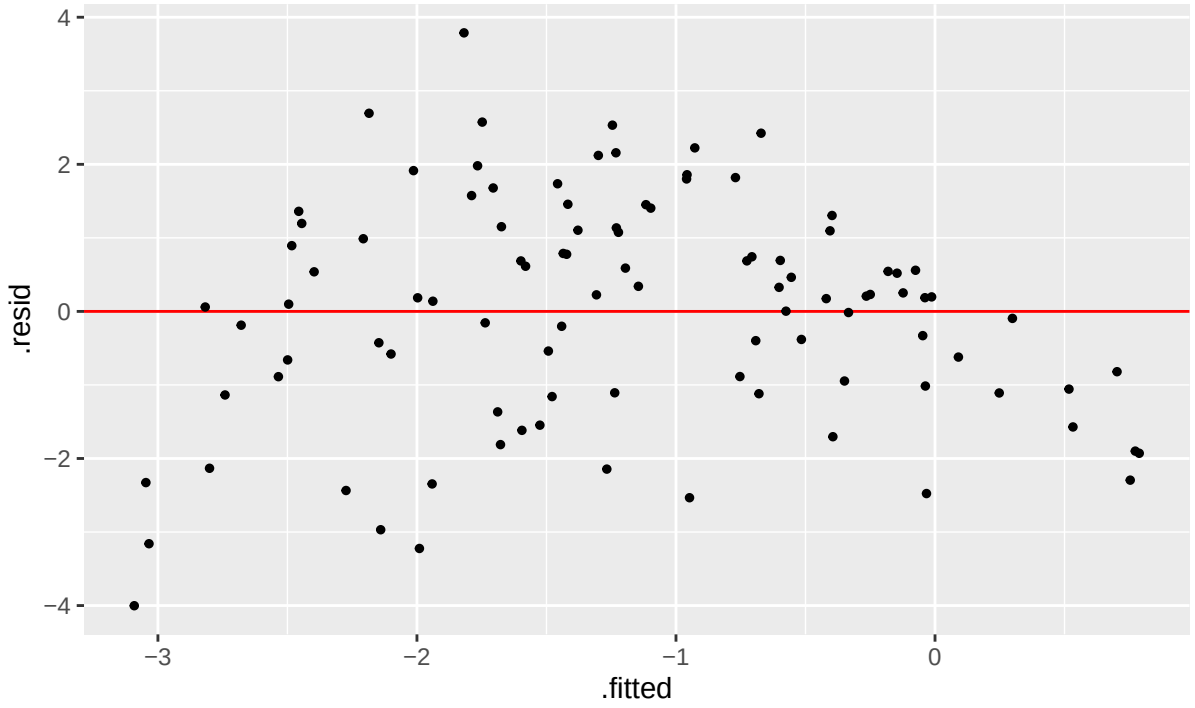
An instance of cubic model class contains methods of simulating data and making residual plot. Method `mod$gen()` returns a data frame containing realizations of X, Z, Y and ε simulated from the model. The number of realizations depends on the integer argument `n`. In addition, if argument `fit_model = TRUE`, a null model will be fitted using the simulated data and residuals and fitted values will be included in the returned data frame.

```
mod$gen(n = 5, fit_model = TRUE)
```

```
##             y          x          z          e     .resid     .fitted
## 1  0.44298045  0.1862316 -0.5114228  0.2749620  0.3561304  0.08685009
## 2 -0.46546366 -0.8542129 -0.3789133  0.3543143  0.2827120 -0.74817570
## 3  0.07636314  0.0383946 -0.3832281 -0.3024668 -0.2698449  0.34620801
## 4 -1.04762148 -0.4177674 -0.5876853 -0.3101100 -0.1202573 -0.92736416
## 5 -0.90198583 -0.5439192 -0.4630171 -0.1448947 -0.2487402 -0.65324563
```

Method `mod$plot()` produce a ggplot (Wickham, 2011) object. It takes a data frame containing columns `.resid` and `.fitted` as input, along with a character argument `type` indicating the type of the data plot, and other aesthetic arguments such as `size` and `alpha` to control the appearance of the plot.

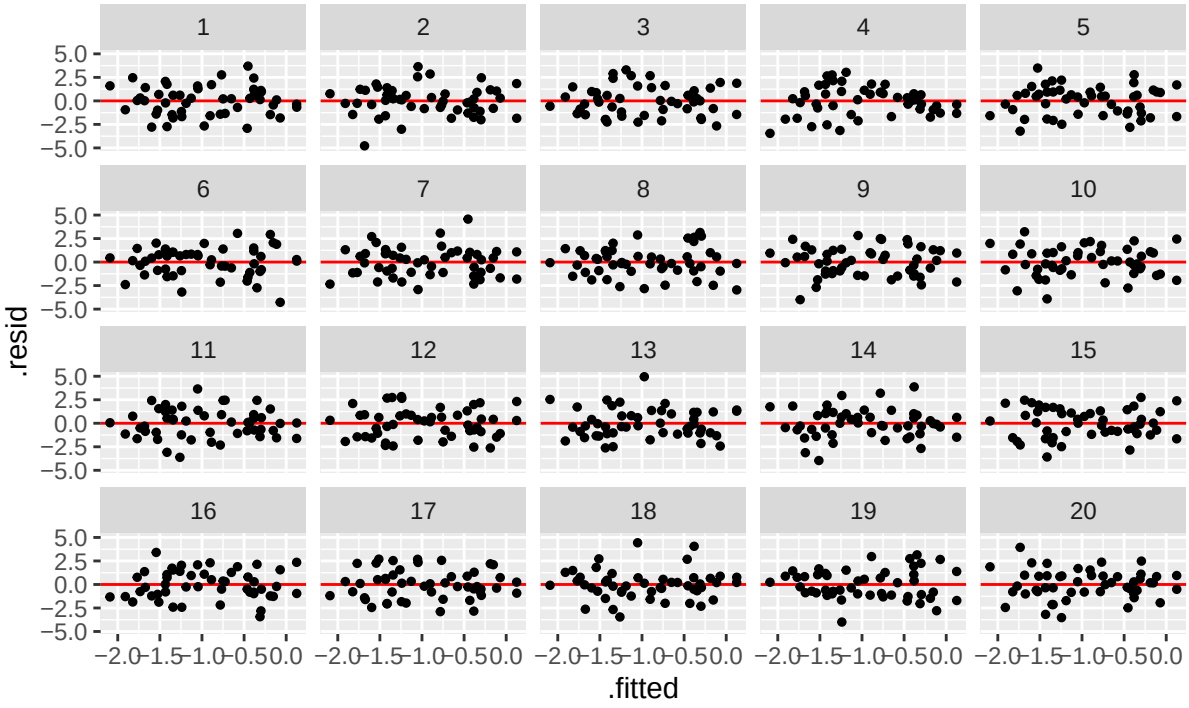
```
mod$plot(mod$gen(n = 100, fit_model = TRUE), type = "resid", size = 1)
```



Lineup can be produced by using the methods `mod$gen_lineup()` and `mod$plot_lineup()`. Method `mod$gen_lineup` takes the number of realizations `n`

and the number of plots in a lineup `k` as inputs. And the method `mod$plot_lineup()` has the same user interface as `mod$plot()`.

```
mod$plot_lineup(mod$gen_lineup(n = 50, k = 20), type = "resid", size = 1)
```



The cubic model class also provides method to compute the expected values of residuals. Method `mod$E()` takes a data frame with columns `x` and `z` as input, and returns a vector of expected values of residuals.

```
mod$E(mod$gen(n = 5))
```

```
## [1] -0.9369269  0.1945678  0.2816923 -0.3254415  0.7861083
```

Since we know that under the null hypothesis, the residual $\boldsymbol{e} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{R}_a)$. Thus, the difference between the expected values $\boldsymbol{R}_a \boldsymbol{X}_b \boldsymbol{\beta}_b$ and $\mathbf{0}$ represents the direct impact of the parameters a and b on the residuals. It is expected that the larger the magnitude of the expected value relative to the variance and covariance, the easier the human to spot the patterns in the residual plot.

To obtain a measure of the impact of a and b on the residuals adjusted for variance and covariance, we need to address several properties of the residuals. First, the variance

of the residuals $\sigma^2 \boldsymbol{R}_a$ is not an identity matrix. This can be fixed by standardizing the residuals by their variance-covariance matrix. Second, the difference between $\boldsymbol{R}_a \boldsymbol{X}_b \boldsymbol{\beta}_b$ and $\mathbf{0}$ could be negative, which is not ideal for comparison. Thus, the magnitude needs to be squared. Third, the measure needs to be a scalar. We could apply a weighted average operator $\boldsymbol{W}$ on the transformed expected residuals to obtain a single numeric value. For simplicity, we set $\boldsymbol{W} = n^{-1}\mathbf{1}$. Considering the high time complexity of computing the square root of $\boldsymbol{R}_a$, off-diagonal elements of $\boldsymbol{R}_a$ are set to be zeros. This gives the effect size:

$$ES = n^{-1}||\sigma^{-1}\boldsymbol{R}_a^{-\frac{1}{2}}\boldsymbol{R}_a\boldsymbol{X}_b\boldsymbol{\beta}_b||^2 = n^{-1}\sigma^{-2}||\boldsymbol{R}_a^{\frac{1}{2}}\boldsymbol{X}_b\boldsymbol{\beta}_b||^2 \approx n^{-1}\sigma^{-2}||diag(\boldsymbol{R}_a)^{\frac{1}{2}}\boldsymbol{X}_b\boldsymbol{\beta}_b||^2,$$

where $diag(\boldsymbol{R}_a)$ is the diagonal matrix constructed from the diagonal elements of $\boldsymbol{R}_a$.

The interpretation of this effect size is the impact of parameter a and b on the squared deviation of the standardized expected residual per observation. It is not directly related to the shape, or the pattern human observed from the residual plot, but it is a reasonable approximation of the degree of the visual deviation from the null residuals under our cubic model setting. Figure 3.6 shows four residual plots with different effect sizes. As effect sizes increases, the strength of the signal become stronger.

Under the cubic model setting, there is an exact conventional test for testing the non-linearity defect, which is F-test. For F-test, the null hypothesis is $H_0 : a = b = 0$, and the alternative hypothesis is H_1 : at least one of them $\neq 0$. During the simulation of the lineup data, the F-statistic and the p-value will be recorded for comparison between the power of conventional test and visual test.

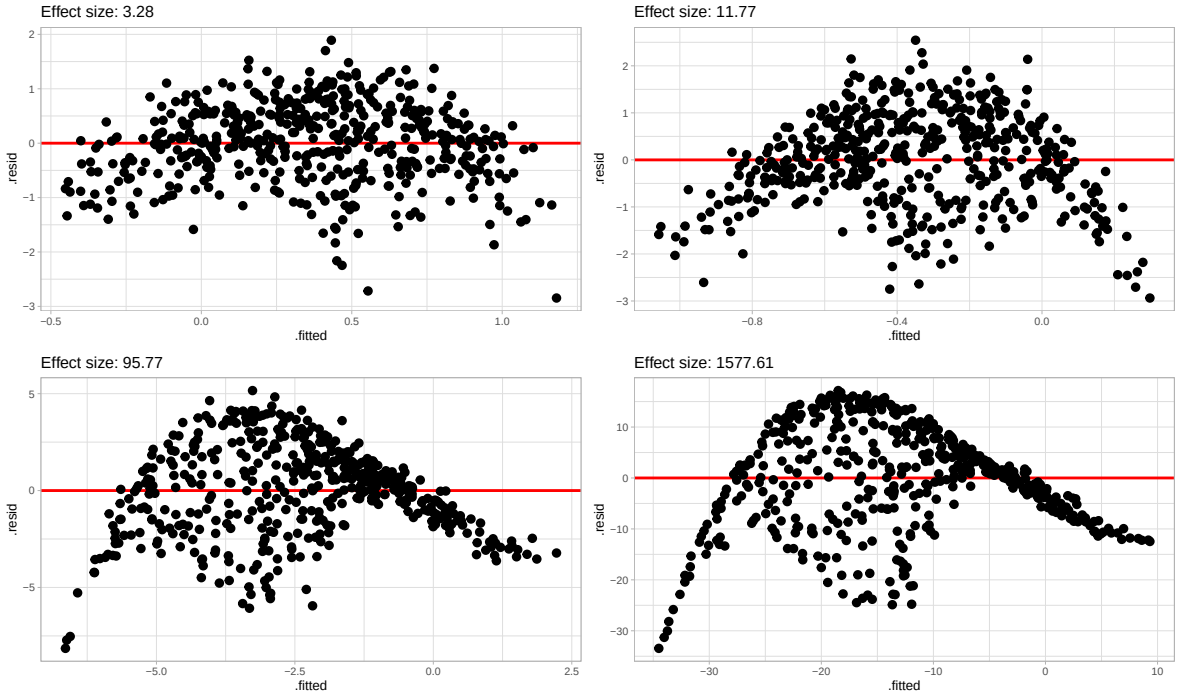


Figure 3.6: *Cubic model residual plots under different effect sizes. The larger the effect size, the stronger the signal.*

3.2 Heteroskedasticity Model

Another model used in the experiments was a heteroskedasticity model with one regressor, which can be expressed by:

$$Y_i = 1 + X_i + \varepsilon_i,\ i = 1, ..., n, \tag{3.3}$$

where $a \in \{-1, 0, 1\}$, $b \in (0, 32)$ and $\varepsilon_i \overset{iid}{\sim} N(0, 1 + b(2 - |a|)(X_i - a)^2)$.

To create heteroskedasticity defect, OLS was used to fit the null model:

$$\boldsymbol{Y} = \beta_0 + \beta_1 \boldsymbol{X} + \boldsymbol{u}, \tag{3.4}$$

where $\boldsymbol{u} \sim N(\boldsymbol{0}, \sigma_u^2 \boldsymbol{I})$.

In this case, estimators of β_0 and β_1 are unbiased, but the error term has non-constant variance.

Lemma 3.2 (Distribution of residuals produced by the heteroskedasticity model)**.** *Given the data generating process in Equation (3.3) and null model in Equation (3.4). Let* $\boldsymbol{X}_a = [\mathbf{1}, \boldsymbol{X}]$ *denotes the set of regressors in matrix form. The residuals obtained from the null model are*

$$\boldsymbol{e} \sim N(\mathbf{0}, \boldsymbol{R}_a \boldsymbol{V}),$$

where $\boldsymbol{R}_a = \boldsymbol{I} - \boldsymbol{X}_a(\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}\boldsymbol{X}_a'$ *and* $\boldsymbol{V}$ *is a diagonal matrix with* $V_{ii} = 1 + b(2 - |a|)(X_i - a)^2$, $i = 1, ..., n$.

Proof. Using the Frisch–Waugh–Lovell theorem, the residuals obtained by the null model are $\boldsymbol{e} = \boldsymbol{R}_a\boldsymbol{Y} = \boldsymbol{R}_a(\boldsymbol{X}_a\boldsymbol{\beta}_a + \boldsymbol{\varepsilon})$, where $\boldsymbol{R}_a = \boldsymbol{I} - \boldsymbol{X}_a(\boldsymbol{X}_a'\boldsymbol{X}_a)^{-1}\boldsymbol{X}_a'$ and $\boldsymbol{\beta}_a = (1, 1)'$.

Because $\boldsymbol{R}_a\boldsymbol{X}_a = \mathbf{0}$, we have $\boldsymbol{e} = \boldsymbol{R}_a\boldsymbol{\varepsilon}$. Hence, the residuals $\boldsymbol{e}$ follow $N(\mathbf{0}, \boldsymbol{R}_a\boldsymbol{V})$, where $\boldsymbol{V}$ is a diagonal matrix with $V_{ii} = 1 + b(2 - |a|)(X_i - a)^2$, $i = 1, ..., n$. □

Lemma 3.2 shows that variance and covariance of the residuals depend on $\boldsymbol{X}$. We could plot the one standard deviation around the residuals to indicate the region about 68% of the residuals should landed on. Figure 3.7 illustrates different shapes of residuals undder various values of a and b. From the plot, it can be observed that if $a = 0$, the residual plot looks like the shape of butterfly. If $a = \pm 1$, it looks like a triangle, and the sign of a determines the direction of the shape. Parameter b is controlling the strength of the signal. As b increases, the pattern becomes less noisy.

Similar to the cubic model, the heteroskedasticity model could also be built by the heteroskedasticity model class constructor `heter_model()`. This function takes three arguments as inputs, which are `a`, `b` and `x`. `a` and `b` are numeric parameters defined in Equation (3.3). `x` needs to be a random variable object.

```
library(visage)
mod <- heter_model(a = 0, b = 16, x = rand_uniform(-1, 1))
```

Since both the cubic model class and the heteroskedasticity class are inherited from the visual inference model class, which has defined methods of simulating data, making residual plot and producing lineup, heteroskedasticity model object can be used in a

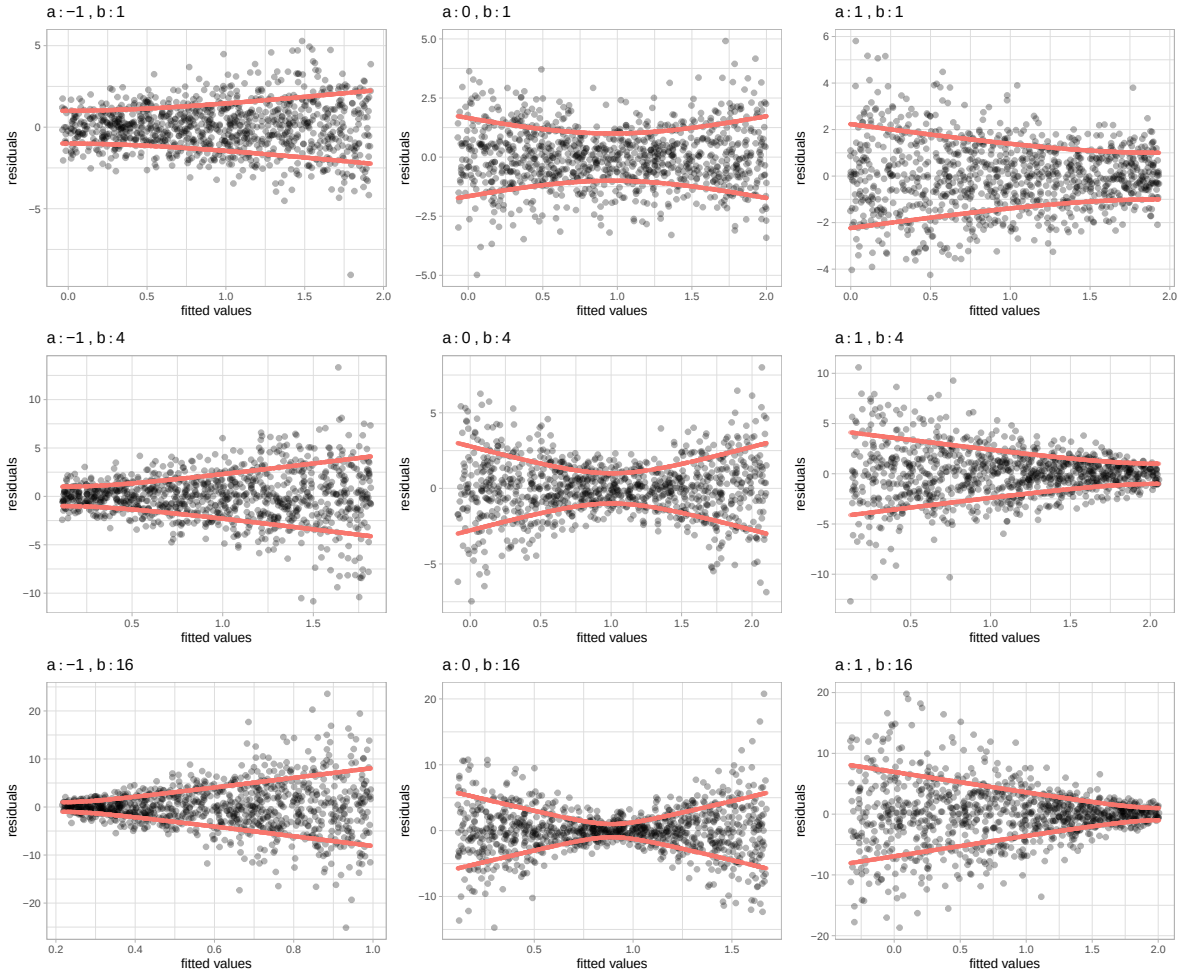


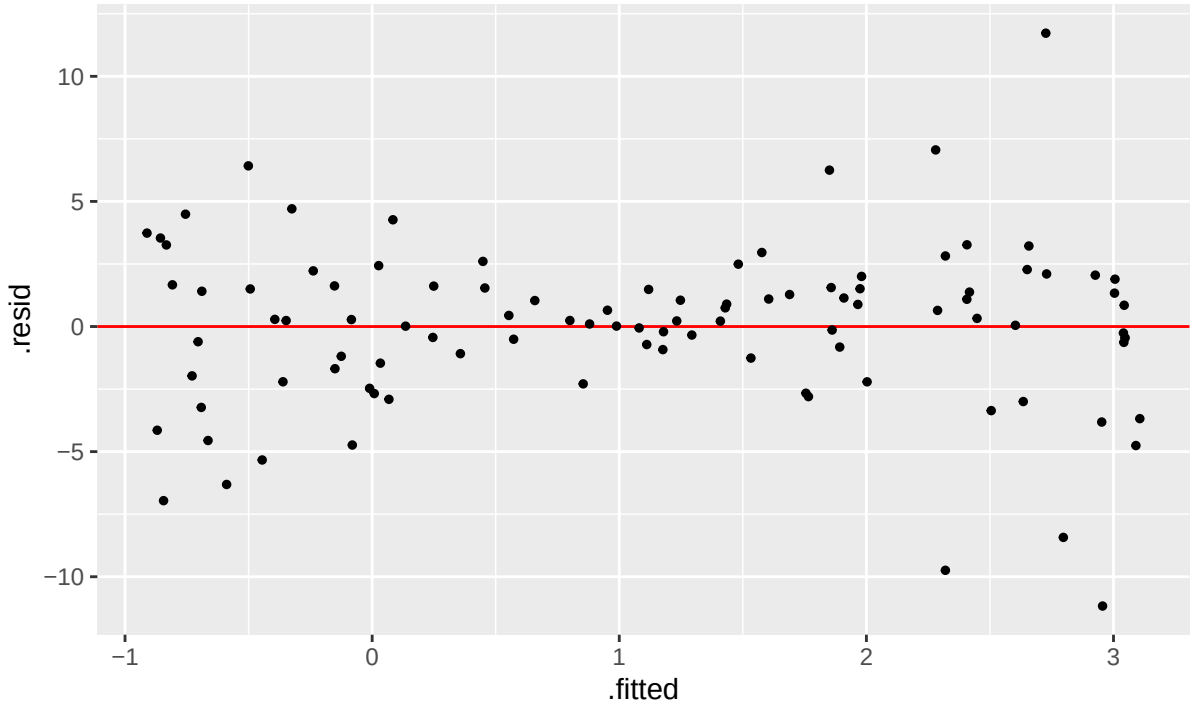
Figure 3.7: *test*

similar way as cubic model object. The following codes give examples of the use of the object.

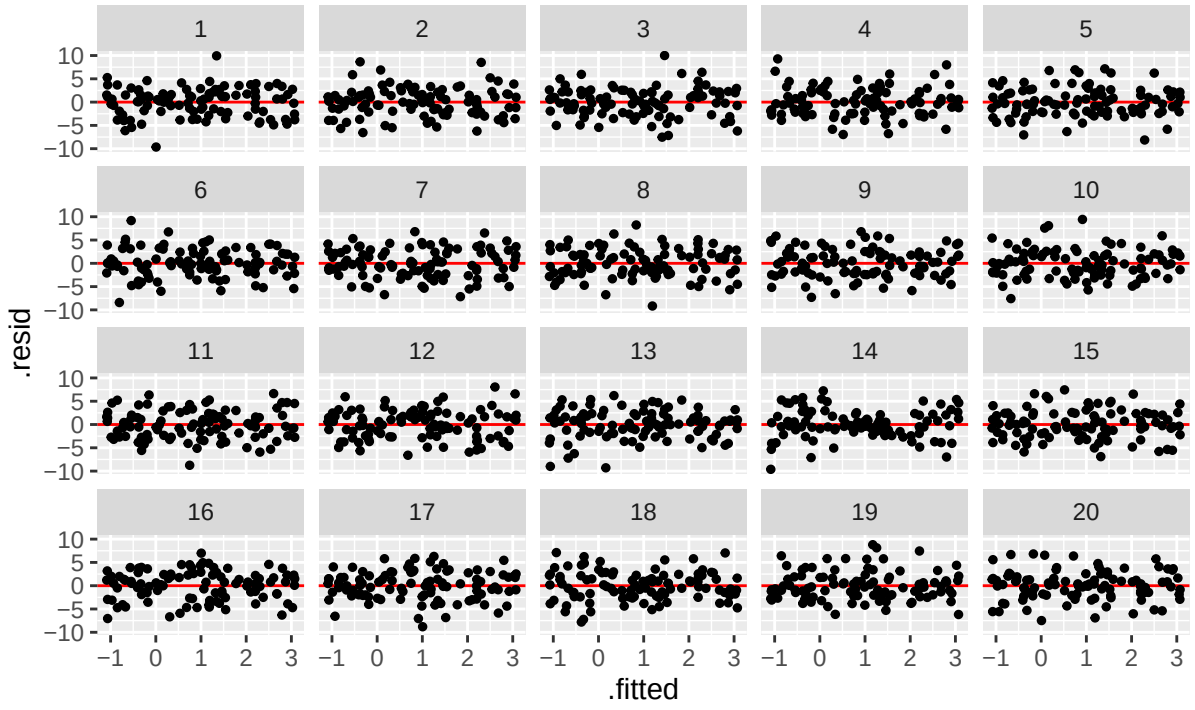
```
mod$gen(5, fit_model = TRUE)
```

```
##          y          x    sigma          e     .resid   .fitted
## 1 7.205499  0.9053470 5.218132 5.3001525  0.7420902 6.463409
## 2 2.560696 -0.8626811 4.981465 2.4233771  1.0441078 1.516588
## 3 1.121106 -0.6545505 3.835357 0.7756568 -0.9778170 2.098923
## 4 5.296467  0.7171334 4.178154 3.5793335 -0.6403337 5.936801
## 5 2.962130 -0.2859733 1.901837 2.2481030 -0.1680473 3.130177
```

```
mod$plot(mod$gen(100, fit_model = TRUE), size = 1)
```




```
mod$plot_lineup(mod$gen_lineup(100), size = 1)
```



According to Lemma 3.2, when $b = 0$, the matrix V collapses to an identity matrix. Assume the shape of butterfly and triangle have identical visual impacts, the only factor

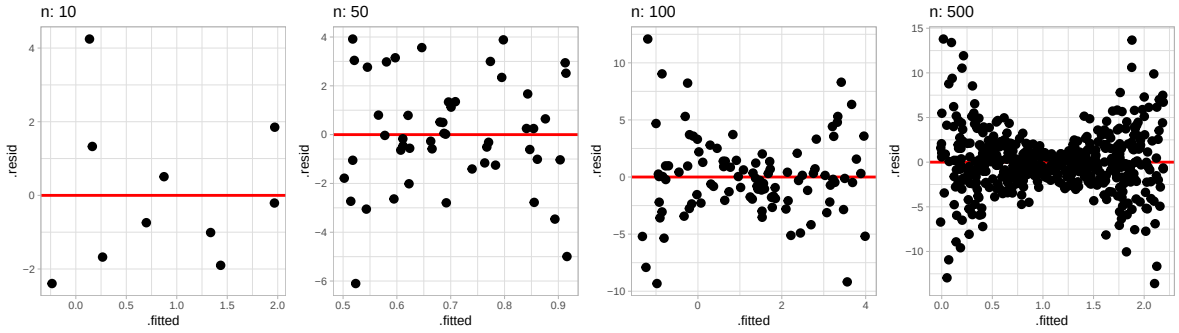


Figure 3.8: *Residuals generated from the same heteroskedasticity model but with different sample size. As sample size increases, the shape of butterfly becomes more obvious.*

affects human to recognize the pattern is the strength of the signal, where parameter a has no role to play. In addition, sample size has a huge impact on the chance of human recognizing the pattern. As shown in Figure 3.8, as sample size increases, the pattern becomes easier to observe. However, this effect is less noticeable with large sample size as the outline of the shape has been drawn and residuals have less probability to land outside of it. Thus, the effect size of this model can be expressed by $ES = b\sqrt{n}$. The square root operator is used for addressing the large sample issue. Figure 3.9 shows the effectiveness of the effect size.

For the heteroskedasticity model, the conventional test we used was Breusch–Pagan test (Breusch and Pagan, 1979), which tested whether the variance of the error terms of the regression is dependent on the regressors with the auxiliary regression equation

$$\boldsymbol{e}^2 = \gamma_0 + \gamma_1 \boldsymbol{X} + \gamma_2 \boldsymbol{X}^2 + \boldsymbol{v}.$$

Majumder, Hofmann, and Cook (2013) suggested that visual test is not expected to perform equally well as conventional test especially when there exists a exact conventional test. However, in contrast to the F-test used in the cubic model, Breusch–Pagan test is an approximate test. Thus, the power of visual test may exceed the power of Breusch–Pagan test. Throughout the study, Breusch–Pagan test statistic and p-value were recorded for comparison.

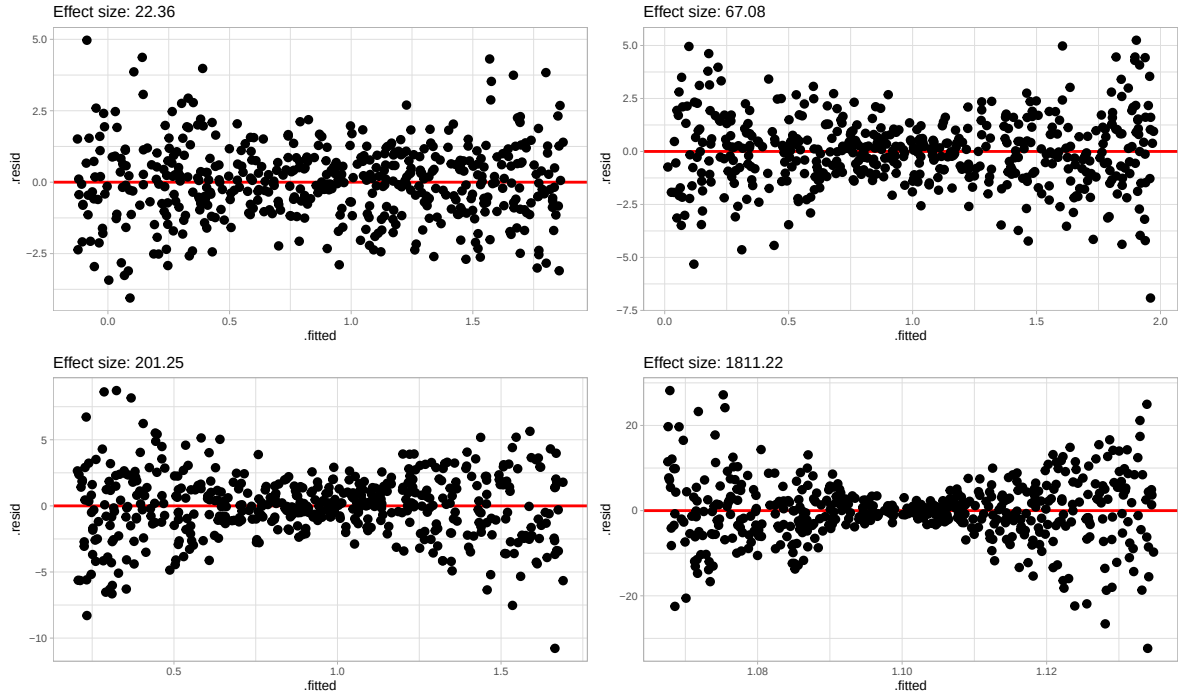


Figure 3.9: *Residuals of heteroskedasticity model with different effect size. As effect size increases, the shape of butterfly becomes less noisy.*

3.3 Distribution of regressors

The model definitions given in the previous two sections does not include the specification of the regressors. In this sections, distribution of $\boldsymbol{X}$ and $\boldsymbol{Z}$ will be discussed.

The cubic model involved the use of both $\boldsymbol{X}$ and $\boldsymbol{Z}$. In the simulation, X_i, $i = 1, ..., n$, had equal chance to follow one of the following distributions: $N(0, 0.09)$, $U(-1, 1)$, $Lognormal(0, 0.36)/3 - 1$ and $-Lognormal(0, 0.36)/3 + 1$. Uniform and normal distribution were symmetric and common. Adjusted lognormal distribution and adjusted negative lognormal distribution provided right-skewed and left-skewed density respectively. These distributions were chosen such that most the realizations will fall between -1 and 1.

The distribution of Z_i, $i = 1, ..., n$, had 50% chance to be a uniform distribution ranged from -1 to 1, and 50% chance to be a discrete uniform distribution with z_n outcomes simulated from a uniform distribution ranged from -1 to 1. z_n itself was a discrete uniform distribution with outcomes $\{3, 4, 5, 6, 7, 8, 9, 10\}$, which defined the number of possible

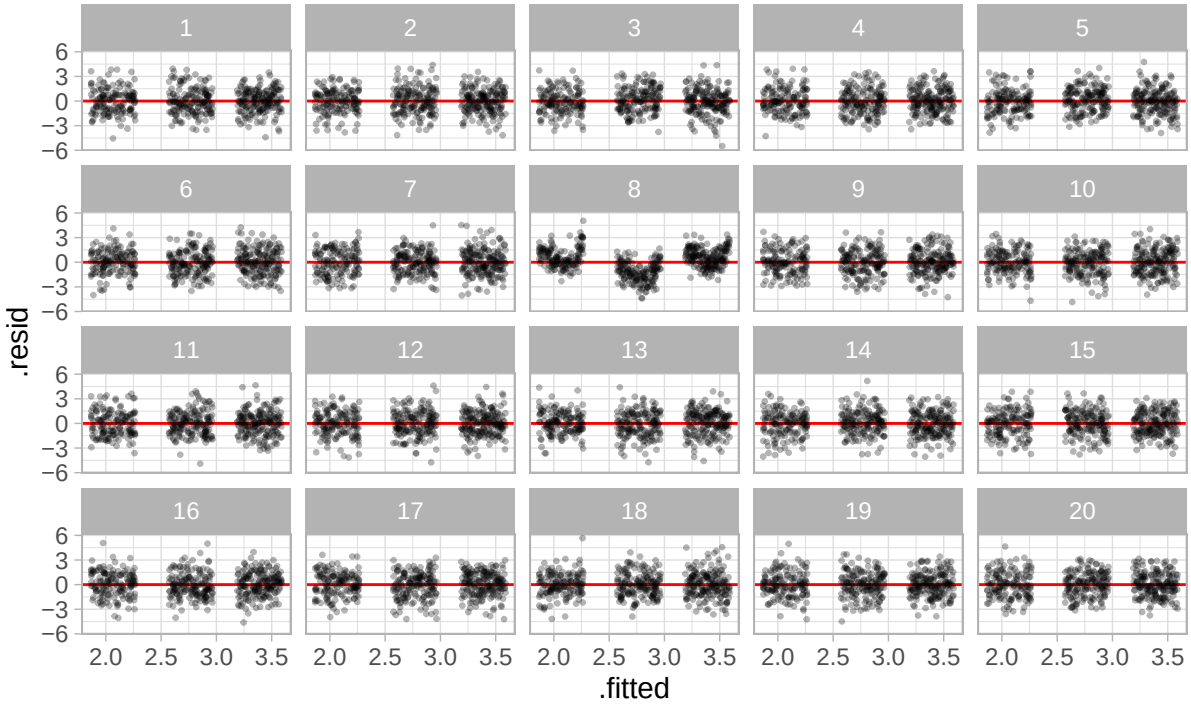


Figure 3.10: *Discreteness in residuals created by using a discrete uniform random variable as one of the regressors in a cubic model. For each residual plot of the lineup, there are three clusters because the number of possible values the regressor can take is three.*

values Z_i could take. As shown in Figure 3.10, this setup would create discreteness in residual plot, which could enrich the pool of visual patterns.

$\boldsymbol{X}$ used in the heteroskedasticity model was a combination of $\boldsymbol{X}$ and $\boldsymbol{Z}$ used in the cubic model. It could be one of the five distributions mentioned above - normal distribution, uniform distribution, adjusted lognormal distribution, adjusted negative lognormal distribution and discrete uniform distribution.

3.4 Experiment I and II

In this study, multiple selection was allowed for the evaluation of a lineup. Given our desired significance level was $\alpha = 0.05$ and the number of plots in a lineup was 20, a single visual test procedure needed to involve multiple evaluations. Otherwise, the p-value of the test would be greater than 0.05 in spite of the actual data plot being selected. The number of plots selected by a user depended on the user and the difficulty of the lineup. According to the pilot study evaluated by 10 faculty members of Department

of Econometrics and Business Analytics, Monash University, the number of selections would be generally smaller than 4, which suggested a visual test consisted of 5 evaluations were sufficient to yield p-value smaller than 0.05 with at least three detections. Thus, each lineup was replicated for five times and evaluated by different participants. However, our study website was designed such that any set of lineups consisting of 20 lineups generated beforehand would be shown to only one person only once. And whether the participant would finish the experiment was unpredictable. Therefore, some lineups had insufficient number of evaluations, but some had more evaluations than expected. This slightly affected the estimate of the power of the visual test, which will be discussed in Section 3.5.

We called the lineup being detected if the actual data plot was one of the selections of an evaluation of a lineup. The sample size calculation was based on the detection rate of the lineup, which was closely related to the difficulty level of the lineup. With the data collected from the pilot study, two logistic regressions given in Table 3.2 and Figure 3.11 were developed to describe the relationship between the natural logarithm of effect size and the detection rate for the cubic model and the heteroskedasticity model.

Table 3.2: *GLM result 1*

	Dependent variable:	
	detect	
	Cubic	Heteroskedasticity
	(1)	(2)
log(effect_size)	0.611***	0.737***
	(0.098)	(0.137)
Constant	0.457***	−2.402***
	(0.165)	(0.645)
Observations	246	289
Log Likelihood	−122.305	−147.971
Akaike Inf. Crit.	248.610	299.943
Note:	*p<0.1; **p<0.05; ***p<0.01	

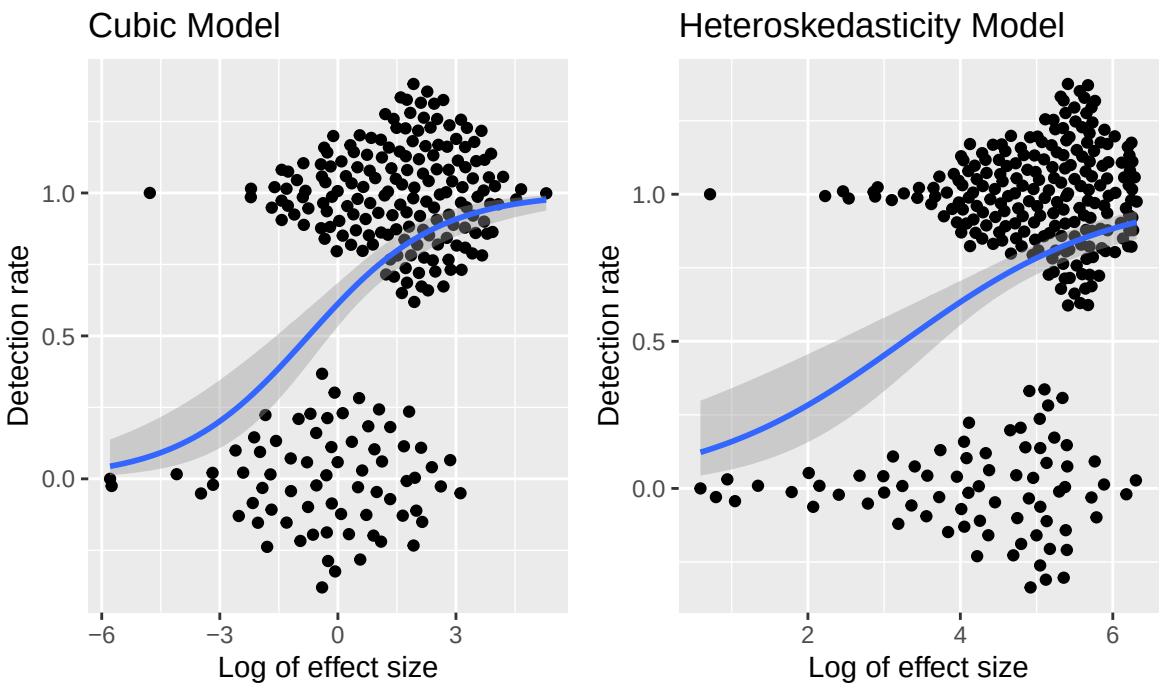


Figure 3.11: *pilot*

Lineups were classified into three categories - easy, medium and hard. Easy lineups were those of which predicted detection rates were higher than 80%, while medium lineups were between 40% to 80% and hard lineups were less than 40%.

For the first experiment, we would like to have at least 20 detection with chance greater than 99.99% in each category. Thus, we prepared 68 easy lineups, 120 medium lineups and 388 hard lineups with five replications. Every participant would get at least 8 easy or medium lineups. Two out of 10 extremely lineups with predicted detection rate over 90% were randomly given to every participant as attention checks. We initially planned to recruit $(68 + 120 + 388) \times 5/(20 - 2) = 160$ participants but eventually decided to only recruit 20 participants because it was the first time we launched an experiment on Prolific where factors like payment method and quality of participants were unclear at the moment. As a consequence, most of the lineups had only two evaluations and not all the lineups were used. However, data collected from the first experiment was sufficient for building a more accurate logistic regression to predict detection rate for the second experiment.

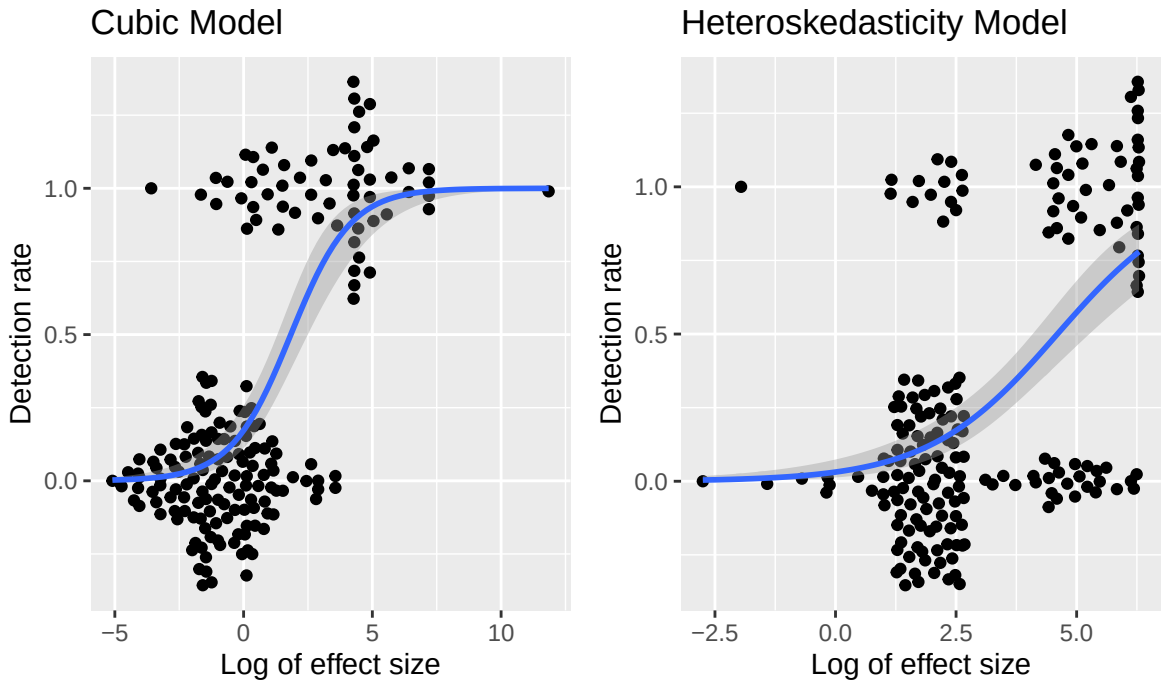


Figure 3.12: *test*

Table 3.3: *GLM result 2*

	Dependent variable:	
	detect	
	Cubic	Heteroskedasticity
	(1)	(2)
I(log(effect_size))	0.818***	0.710***
	(0.113)	(0.107)
Constant	−1.649***	−3.425***
	(0.254)	(0.454)
Observations	201	199
Log Likelihood	−68.678	−89.985
Akaike Inf. Crit.	141.357	183.971
Note:	*p<0.1; **p<0.05; ***p<0.01	

According to the new logistic regression given in Table 3.3 and Figure 3.12, it was found that the first experiment might be too difficult for participants as the detection rate was lower than what we estimated from the pilot study, especially for the heteroskedasticity model. Thus, for the second experiments, we relaxed the sample size requirements and increased the proportion of easy lineups. In addition, due to budget constraint, the maximum number of participants was 50. With the new logistic regression fitted with data collected from the first experiment, 28 easy lineups, 36 medium lineups and 100 hard lineups were simulated for the second experiment. 50 participants were recruited but 6 were withdrew.

3.5 P-value and Power Estimation of Visual Test Allowed for Multiple Selections

The p-value calculation for multiple selection is an extension of Equation (1.1). The assumptions about the independence still needed. Let K be the number of independent evaluations of a visual test, s_i, $i = 1, ..., K$ be the number of selections of the evaluations, and X be the random variable denoting the number of detections. Then, the p-value of the visual test is given as:

$$P(X \geq x) = \sum_{j=x}^{K} Pr(j|s_1, ..., s_K). \tag{3.5}$$

The distribution of X given $s_1, ...s_K$ can not be derived trivially, as it is a sampling without replacement problem. In practice, this distribution can be approximated by computer simulation.

Function `sim_dist()` from package `visage` is designed to approximate the distribution of number of detections of a lineup via Monte Carlo method. It takes the number of evaluations `n_eval`, the number of selections `n_sel`, and the number of plots in a lineup `n_plot` as inputs, then outputs a discrete distribution.

```
sim_dist(n_eval = 3, n_sel = c(2, 2, 3), n_plot = 20)
```

```
##       0       1       2       3
## 0.70182 0.25202 0.04346 0.00270
```

Function `calc_p_value()` from the same package calculates the p-value using the distribution returned by `sim_dist()`. It takes an additional argument `n_detect`, which is the number of detections.

```
calc_p_value(n_detect = 2, n_eval = 3, n_sel = c(2, 2, 3), n_plot = 20)
```

```
## [1] 0.04688
```

The above example shows that a visual test with two detections and three independent evaluations in which three observers select two, two, and three plots out of 20 plots respectively yields a p-value smaller than 0.05.

Assume there is a visual test V_K, where K denoting the number of evaluations. The corresponding p-value of V_K can be computed by using Equation (3.5). Meanwhile, if one evaluation is randomly deleted from V_K, the remaining evaluations can still be used to form another valid visual test V_{K-1}. In fact, considering all the possibilities, K different outcomes for V_{K-1} can be obtained. Since all outcomes occur with equal probability, the proportion of outcomes which reject the null hypothesis can be used as an estimate of the power of the visual test V_{K-1}. Similarly, if we would like to estimate the power of the visual test V_{K-j} given the evaluations of V_K, for $j < K$, we could find all the possible combinations of K elements, taken $k - j$ at a time to obtain $\binom{K}{K-j}$ different outcomes for V_{K-j}. Then, the estimated power is given as $R/\binom{K}{K-j}$, where R is the number of outcomes which reject the null hypothesis.

Function `calc_p_value_comb()` from the package `visage` can be used to compute p-values of V_{K-j}. The first argument is `detected`, which needs to be a vector of Boolean values denoting whether the observer detects the actual data plot. Desired number of evaluations and number of selections need to be provided via argument `n_eval` and `n_sel`.

For example, given a visual test V_3 where the first observer selects one plot and gets correct, the second observer selects one plot but misses, and the third observer selects two plots and gets correct. The p-value of all possible outcomes of V_2 can be computed using the following code:

```
calc_p_value_comb(detected = c(TRUE, FALSE, TRUE),
                  n_eval = 2,
                  n_sel = c(1, 1, 2))
```

```
## [1] 0.09352 0.00624 0.14454
## attr(,"combinations")
##      [,1] [,2] [,3]
## [1,]    1    1    2
## [2,]    2    3    3
```

`calc_p_value_comb()` returns a vector of p-value along with an attribute which is a matrix representing elements being used by the specific outcome. Elements in the first column indicates that the first observer and the second observer are involved in the visual test, and the corresponding p-value is 0.09352.

3.6 Results

We collected 400 lineup evaluations made by 20 participants in experiment I and 880 lineup evaluations made by 44 participants. In total, 442 unique lineups were evaluated by 64 subjects. In experiment I, one of the participants skipped all 20 lineups. Hence, the submission was rejected and removed from the data set. In experiment II, there was a participant failed one of the two attention checks, but there was no further evidence of low-effort throughout the experiment. So, the submission was kept.

3.6.1 Power Estimation

It was expected that with larger effect size, the power of the visual test would be greater for both cubic model and heteroskedasticity model. Using the power calculation method discussed in Section 3.5, power of each lineup from one to five evaluations was estimated. The estimated power was further used in fitting a quasibinomial generalized linear model

(estimate of visual power)

(power against different parameters)

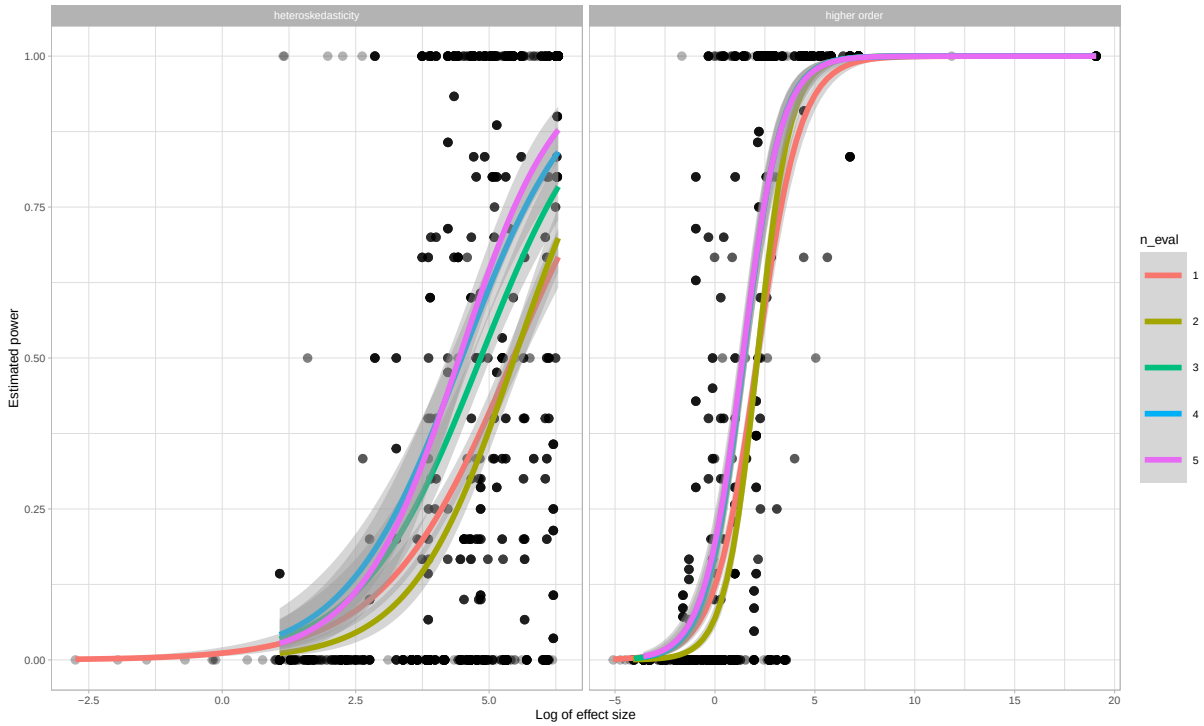


Figure 3.13: *test*

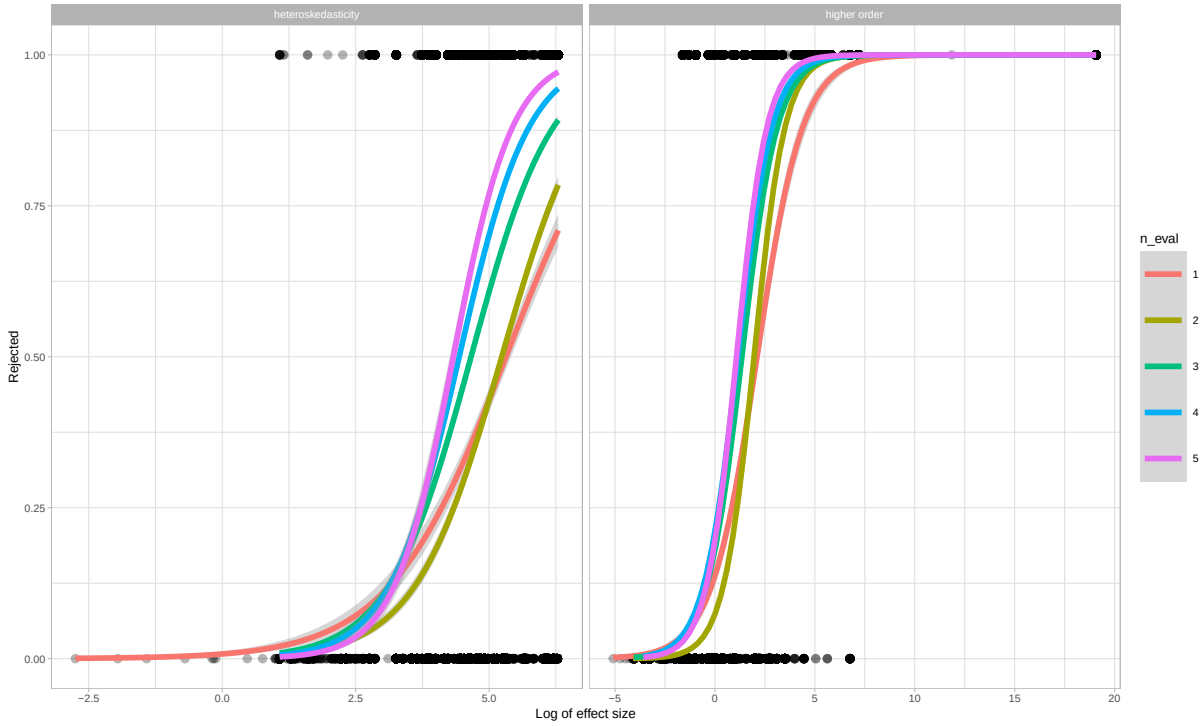


Figure 3.14: *test*

(power against conventional p-value)

(conventional power)

(visual p-value vs. conventional p-value)

(response time)

(confidence level)

(relative freq of picks)

Chapter 4

Automatic Visual Statistical Inference, with Application to Linear Regression Diagnostics

4.1 Abstract

4.2 Introduction

4.2.1 Model Diagnostics

[ET: suggestion: A model can be fitted to data with no guarantee of a meaningful interpretation. Model diagnostics play an important role in assessing the appropriateness of the model. The assessment can involve examining the goodness of fit, checking if there are potential violations of model assumptions and validating models with external information.]

Model diagnostics is the part of data analysis, preceded by the fit of a model, whose primary objectives are to examine the goodness of fit and reveal potential violations of model assumptions.

[ET: model fit is more than just goodness of fit and checking model assumptions. It's too reductive to say this because you could have a model with good fit (according to some metric) and no model violation but if the way that the data were collected was flawed then scientific interpretation of the model is rendered useless. Diagnostics involve also the domain knowledge checks as well.]

In these diagnostics, though numeric summaries are mostly available and some are even endorsed by finite or asymptotic properties, graphical representation of data is still preferred, or at least needed, due to its intuitiveness and the possibility to provide unexpected discoveries which may be abstract and unquantifiable.

[ET: Generally, your sentences are too long. Break it up into smaller sentences. The generally idea is to make sentences that *continue an idea from the preceding sentence* so it leads from one to the next naturally. E.g. Today is cloudy. Cloudy days are rare. Perhaps today will be different to other days!]

However, unlike confirmatory data analysis built upon rigorous statistical procedures, e.g., hypothesis testing, visual diagnostics relies on graphical perception - human's ability to interpret and decode the information embedded in the graph (Cleveland and McGill, 1984), which is to some extent subjective. Further, visual discovery suffers from its unsecured and unconfirmed nature where the degree of the presence of the visual features typically can not be measured quantitatively and objectively, which may lead to over or under-interpretations of the data. One such example is finding a separation between gene groups in a two-dimensional projection from a linear discriminant analysis where there is no difference in the expression levels between the gene groups (Roy Chowdhury et al., 2015).

[ET: Confirmatory data analysis is well characterised by a rigorous procedure to discern particular hypothesis. The most widespread form of confirmatory data analysis are the use of p-values in testing hypothesis in a frequentist framework. More specifically, hypothesis testing in a frequentist framework involve stating a well-defined hypotheses; summarising the evidence as a test statistic; and calculating the probability of observing a statistic as extreme as the observed test statistic under the null hypothesis. This rigour, however, is often not present when inferences are drawn from a plot.

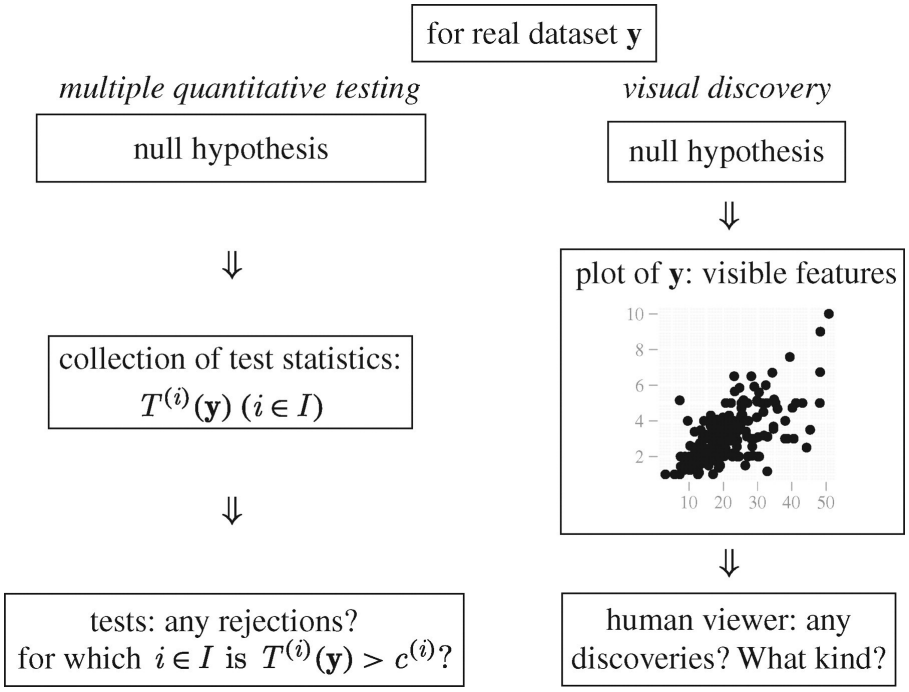


Figure 4.1: *Parallelism between multiple quantitative testing and visual discovery (Buja et al., 2009). Visible features in a plot are viewed as a collection of test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$, and any visual discoveries are treated as evidence against the null hypothesis.*

Something about model diagnostics with a plot being the common thing to do..]

4.2.2 Visual Inference

Visual inference was first introduced by Buja et al. (2009) as an inferential framework to extend confirmatory statistics to visual discoveries. This framework redefines the test statistics, tests, null distribution, significance levels and p-value for visual discovery modelled on the confirmatory statistical testing. Figure **??** outlines the parallelism between conventional tests and visual discovery.

[ET: note use Rmd syntax for figures references! So it's easy to use it for HTML later if you want to do something like Earo and have PDF + HTML thesis.]

Parallelism between multiple quantitative testing and visual discovery (Buja et al., 2009). Visible features in a plot are viewed as a collection of test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$, and any visual discoveries are treated as evidence against the null hypothesis. Parallelism between multiple quantitative testing and visual discovery (Buja et al., 2009).

In visual inference, a visual discovery is defined as a rejection of a null hypothesis, and the same null hypothesis can be rejected by many different visual discoveries (Buja et al., 2009). For model diagnostics, the null hypothesis would be the assumed model, while the visual discoveries would be any findings that are inconsistent with the hypothesis. The same assumed model, such as classical linear regression model, can be rejected by both nonlinearity and heteroskedasticity with the residual plot as shown in Figure 1.3.

4.2.3 Pre-specification of Visual Discoverable Features

As discussed in Buja et al. (2009), in the practice of model diagnostics, the range of possible visual discoveries is not pre-specified. In other words, people do not explicitly specify which one or more visual features they are looking for before the read of the diagnostic plot. This is concerning since conventional hypothesis testing always requires the pre-specification of the parameter space Θ of the parameter of interest $\theta \in \Theta$ to form a valid inferential procedure. To address this issue, a collection of test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$ is defined, where $\mathbf{y}$ is the data and $\boldsymbol{I}$ is a set of all possible visual features. Buja et al. (2009) described each of the test statistics $T^{(i)}(\mathbf{y})$ as a measurement of the degree of presence of a visual feature. Alternatively, Majumder, Hofmann, and Cook (2013) avoids the use of visual features and defined the visual statistics $T(.)$ as a mapping from a dataset to a data plot. Both definitions of visual test statistics are valid, but in the rest of the paper, the first definition will be used as it covered some details needed by this work.

The size of the collection $T^{(i)}(\mathbf{y})$ $(i \in I)$ depends on the size of the set I. Thus, if one can define I comprehensively, i.e, pre-specify all the visual discoverable features, the validity issue will be solved. Unfortunately, to our knowledge, there is no such a way to list all visual features. In linear regression diagnostics, possible visual features of a residual plot may be outliers, shapes and clusters. But this is an incomplete list which does not enumerate all the visual features.

Similarly, Wilkinson, Anand, and Grossman (2005) proposed the work called graph theoretic scagnostics, which adopted the idea of "scagnostics" - scatter plot diagnostics from (can't find the 1984 citation). It includes 9 computable scagnostics measures defined on planar proximity graphs: "Outlying", "Convex", "Skinny", "Stringy", "Straight",

"Monotonic", "Skewed", "Clumpy" and "Striated" which attempts to describe outliers, shape, density, trend and coherence of the data. This approach is inspiring but it still does not give the complete list of visual discoverable features. In fact, it is possible that such a list will never be complete as suggested in Buja et al. (2009).

Thinking out of the box, Buja et al. (2009) argued that there is actually no need for pre-specification of visual discoverable features. In model diagnostics, when the null hypothesis is rejected, the reasons for rejecting the hypothesis will also be known. This is because observers can not only point out the fact that visual discoveries have been found, but also describe the particular visual features they observed. Those features will correspond to the subset of the collection of visual test statistics $T^{(i)}(\mathbf{y})$ $(i \in I)$ which resulted in rejection. This argument helps justifies the validity of visual inference.

4.2.4 Lineup Protocol

With the validity of visual inference being justified, another aspect of hypothesis testing that needs to be addressed is the control of false positive rate or Type I error. Any visual statistic $T^{(i)}(\mathbf{y})$ needs to pair with a critical value $c^{(i)}$ to form a hypothesis test. When a visual feature i is discovered by the observer from a plot, the corresponding visual statistic $T^{(i)}(\mathbf{y})$ may not be known as there is no general agreement on the measurement of the degree of presence of a visual feature. It is only the event that $T^{(i)}(\mathbf{y}) > c^{(i)}$ is confirmed. Similarly, if any visual discovery is found by the observer, we say, there exists $i \in I : \ T^{(i)}(\mathbf{y}) > c^{(i)}$ (Buja et al., 2009).

Using the above definition, the family-wise Type I error can be controlled if one can provide the collection of critical values $c^{(i)}$ $(i \in I)$ such that $P(\text{there exists } i \in I : T^{(i)}(\mathbf{y}) > c^{(i)}|\mathbf{y}) \leq \alpha$, where α is the significance level. However, since the quantity of $T^{(i)}(\mathbf{y})$ may not be known, such collection of critical values can not be provided.

Buja et al. (2009) proposed the lineup protocol as a visual test to calibrate the Type I error issue without the specification of $c^{(i)}$ $(i \in I)$. It is inspired by the "police lineup" or "identity parade" which is the act of asking the eyewitness to identify criminal suspect from a group of irrelevant people. The protocol consists of m randomly placed data plots, where 1 plot is the actual data plot, and $m - 1$ null plots are produced by plotting data

simulate from the null distribution which is consistent with the null hypothesis. Then, an observer who have not seen the actual data plot will be asked to point out the most different plot from the lineup.

Under the null hypothesis, it is expected that the actual data plot would have no distinguishable difference with the null plots, and the probability of the observer correctly picks the actual data plot is $1/m$ due to randomness. If we reject the null hypothesis as the observer correctly picks the actual data plot, then the Type I error of this test is $1/m$.

This provides us with an mechanism to control the Type I error, because m - the number of plots in a lineup can be chosen. Further, if we involve K independent observers in a visual test, and let X be a random variable denoting the number of observers correctly picking the actual data plot. Then, under the null hypothesis $X \sim \text{Binom}_{K,1/m}$, and therefore, the p-value of a lineup of size m evaluated by K observer is given as

$$P(X \geq x) = \sum_{i=x}^{K} \binom{K}{i} \left(\frac{1}{m}\right)^{i} \left(\frac{m-1}{m}\right)^{k-i},$$

where x is the realization of number of observers correctly picking the actual data plot (Majumder, Hofmann, and Cook, 2013).

4.2.5 Visual Inference Applied to Linear Regression

How people used visual inference in linear regression?

4.2.6 Limitation of the Visual Inference

What are the limitations?

4.2.7 Computer Vision Model

What is computer vision model?

4.2.8 Contribution

What has been done by this paper?

4.2.9 Structure of This Paper

What is the structure of the paper?

Model diagnostics is the part of data analysis whose primary objectives are to examine the goodness of the model fit and reveal potential violations of the assumptions. Graphical approaches

For regression diagnostics, it may includes the needs of

Linear regression is an modelling approach to describe the relationship between an response variable and one or more explanatory variable. It has been widely used for both generative modeling and predictive modelling.

Regression diagnostics is needed

1. to check whether the assumptions has been violated
2. to check whether the line fit the data

Model diagnostics for linear regression is well developed

Appendix A

Additional stuff

You might put some computer output here, or maybe additional tables.

Note that line 5 must appear before your first appendix. But other appendices can just start like any other chapter.

Bibliography

Breiman, L (2001). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science* **16**(3), 199–231.

Breusch, TS and AR Pagan (1979). A simple test for heteroscedasticity and random coefficient variation. *Econometrica: Journal of the econometric society*. Publisher: JSTOR, 1287–1294.

Buja, A, D Cook, H Hofmann, M Lawrence, EK Lee, DF Swayne, and H Wickham (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**(1906), 4361–4383.

Cleveland, WS and R McGill (1984). Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association* **79**(387), 531–554.

De Leeuw, JR (2015). jsPsych: A JavaScript library for creating behavioral experiments in a Web browser. *Behavior research methods* **47**(1). Publisher: Springer, 1–12.

Donoho, D (2017). 50 Years of Data Science. *Journal of Computational and Graphical Statistics* **26**(4), 745–766.

Flanagan, D (2006). *JavaScript: the definitive guide*. " O'Reilly Media, Inc.".

Grinberg, M (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

Gtihub (2022a). *Github*. https://www.github.com/.

Gtihub (2022b). *Github Pages*. https://pages.github.com/.

Harnad, S (1991). Other bodies, other minds: A machine incarnation of an old philosophical problem. *Minds and Machines* **1**(1). Publisher: Springer, 43–54.

Hofmann, H, L Follett, M Majumder, and D Cook (2012). Graphical Tests for Power Comparison of Competing Designs. *IEEE Transactions on Visualization and Computer Graphics* **18**(12). Conference Name: IEEE Transactions on Visualization and Computer Graphics, 2441–2448.

Jordan, MI and TM Mitchell (2015). Machine learning: Trends, perspectives, and prospects. en. *Science* **349**(6245), 255–260.

Loy, A, L Follett, and H Hofmann (2016). Variations of Q–Q Plots: The Power of Our Eyes! *The American Statistician* **70**(2). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00031305.2015.1077728, 202–214. (Visited on 01/23/2022).

Majumder, M, H Hofmann, and D Cook (2013). Validation of Visual Statistical Inference, Applied to Linear Models. *Journal of the American Statistical Association* **108**(503). Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2013.808157, 942–956. (Visited on 01/21/2022).

Majumder, M, H Hofmann, and D Cook (2014). Human Factors Influencing Visual Statistical Inference. *arXiv:1408.1974 [stat]*. arXiv: 1408.1974.

Prolific (2022). *Prolific*. https://www.prolific.co.

PythonAnywhere (2022). *PythonAnywhere*. https://www.pythonanywhere.com.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Roy Chowdhury, N, D Cook, H Hofmann, M Majumder, EK Lee, and AL Toth (2015). Using visual statistical inference to better understand random class separations in high dimension, low sample size data. en. *Computational Statistics* **30**(2), 293–316. (Visited on 01/23/2022).

Russell, S and P Norvig (2002). Artificial intelligence: a modern approach.

Silver, D, T Hubert, J Schrittwieser, I Antonoglou, M Lai, A Guez, M Lanctot, L Sifre, D Kumaran, T Graepel, T Lillicrap, K Simonyan, and D Hassabis (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. en. *Science* **362**(6419), 1140–1144. (Visited on 02/09/2022).

Turing, AM and J Haugeland (1950). Computing machinery and intelligence. *The Turing Test: Verbal Behavior as the Hallmark of Intelligence*, 29–56.

Van Rossum, G and FL Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace.

Wickham, H (2011). ggplot2. en. *WIREs Computational Statistics* **3**(2). _eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.147, 180–185. (Visited on 01/20/2022).

Wilkinson, L, A Anand, and R Grossman (2005). Graph-theoretic scagnostics. In: *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.* ISSN: 1522-404X, pp.157–164.