MONASH University

MONASH
BUSINESS
SCHOOL

Human vs. Computer

Shuofan Zhang

5/28/2018

Reminder of the first presentation

Our goal

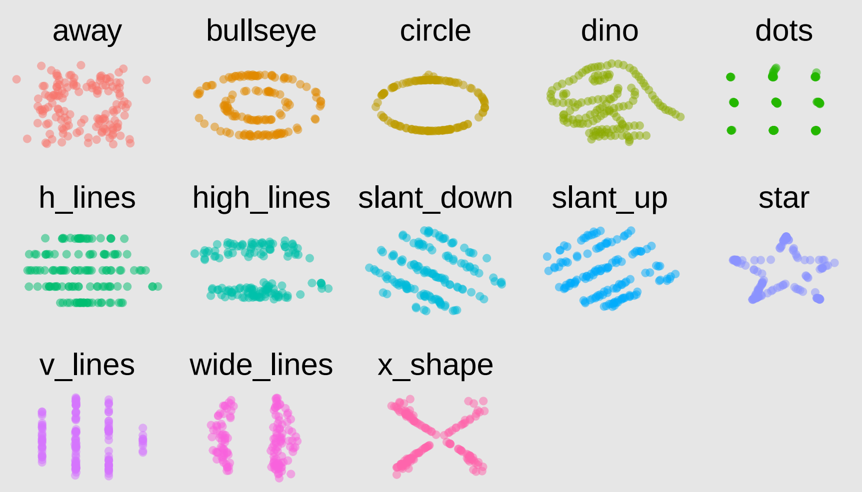
Teach the computer to read residual plots

A major component used to diagnose model fits is a plot of the residuals.

Residual plots are used to assess:

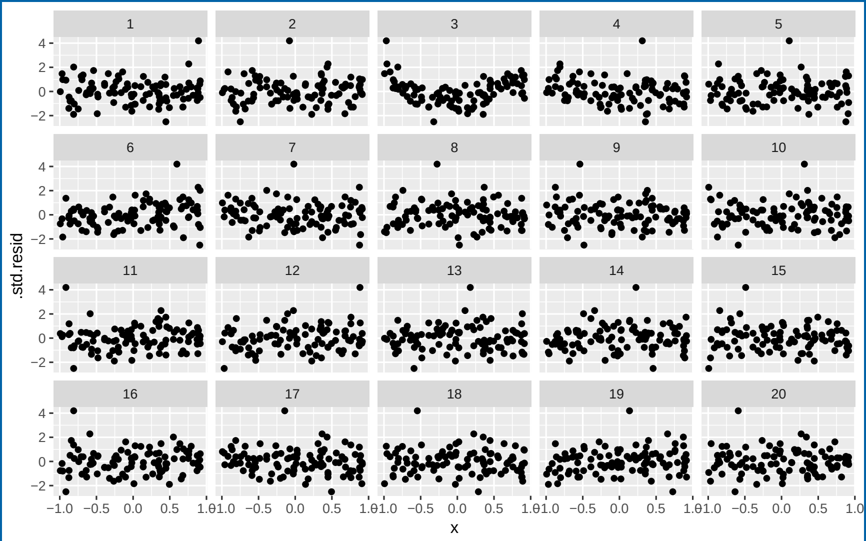
- Gauss-Markov assumption
- Heteroskedasticity
- Clumps of outliers
- . . .

Why plots?



$E(x) = 54.3, E(y) = 47.8, sd(x) = 16.8, sd(y) = 26.9, r = -0.06$

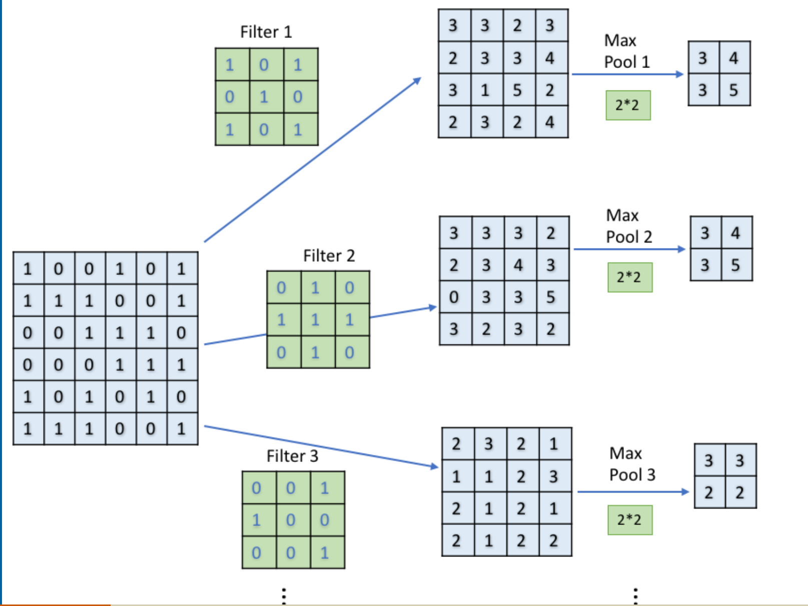
Visual inference



Convolutional neural network (convnets)

- Computer vision has advanced substantially
- If we can train a computer to read residual plots we can have it process a lot more data, than a human can manage.

How convnets works: Diagram of convolution and max pooling



How convnets works

diagram

Aside: Computers can't tell difference between blueberry muffins and chihuahuas

Figure 1: Computers can't tell difference between blueberry muffins and chihuahuas

Our Experiments

- First experiment: Linear vs. Null

H_0: There are no relationships between the two variables. (Null)

H_1: There is linear relationship between the two variables where all Gauss-Markov assumptions are met. (Linear)

- Second experiment: Heteroskedasticity vs. Null

H_0: There is linear relationship between the two variables where all Gauss-Markov assumptions are met. (Null)

H_1: There is linear relationship between the two variables where the variance of the error term is not a constant while all other Gauss-Markov assumptions are met. (Heteroskedasticity)

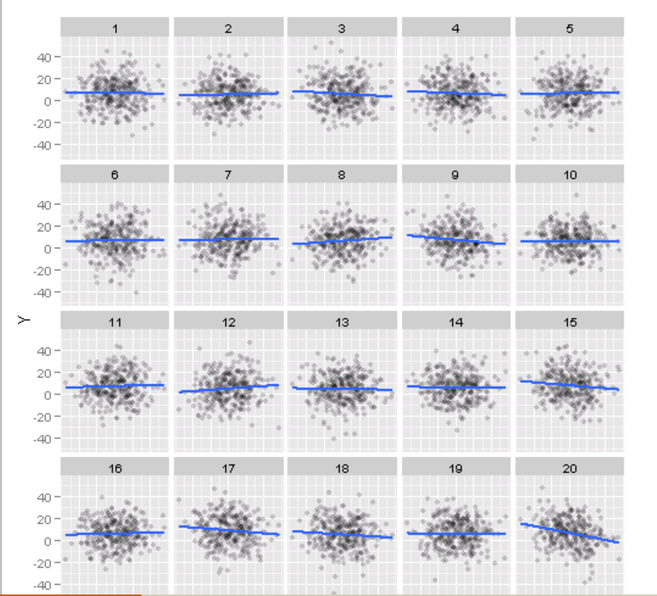
First Experiment: Linear vs. Null

Amazon Mechanical Turk study

- Majumder et al (2013) conducted a large study to compare the performance of the lineup protocol, assessed by human evaluators, in comaprison to the classical test
- Experiment 2 examined $H_o : \beta_k = 0$ vs $H_a : \beta_k \neq 0$ assessing the importance of including variable k in the linear model, conducted with a t-test, and also lineup protocol
- 70 lineups of size 20 plots
- 351 evaluations by human subjects
-
- Trained deep learning model will be used to classify plots from this study. Accuracy will be compared with results by human subjects.

Frist Experiment: Linear vs. Null

Example lineup from Turk experiment 2



First Experiment: Linear vs. Null

Human experiment procedures (diagram)

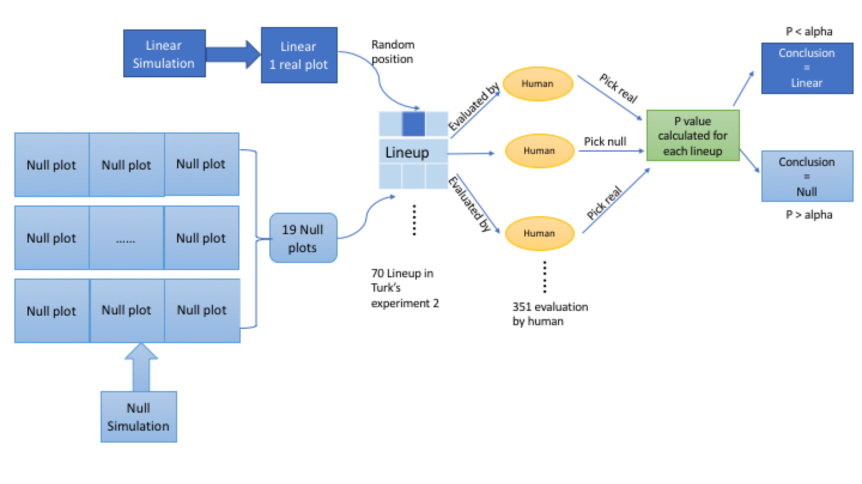


Figure 3: Procedure of computer model experiment

First experiment: Linear vs. Null

Computer model procedures (diagram)

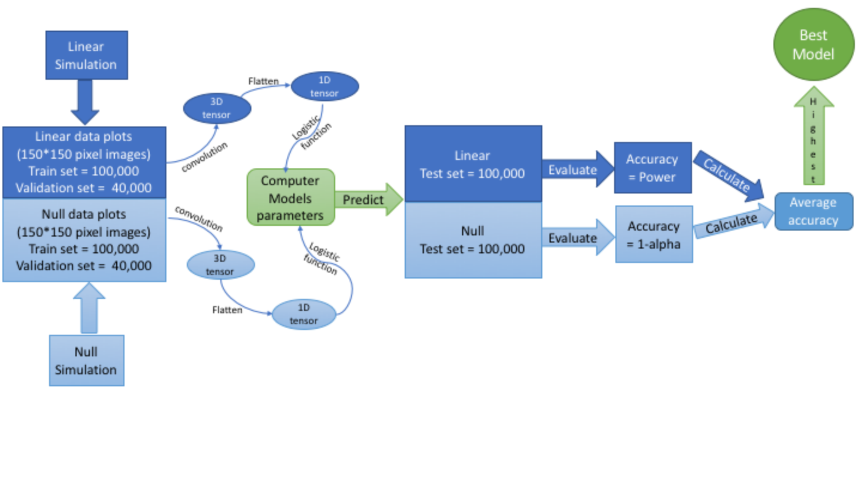


Figure 4: Procedure of computer model experiment

First experiment: Linear vs. Null

Computer model procedures

1. Simulate data (X, Y) from the null and the alternative models
2. Generate scatter plots of X and Y
3. Save scatter plots as 150×150 pixels images
4. Train a deep learning classifier to recognise the patterns from two groups
5. Test the model's performance on new data and compute the accuracy

First Experiment: Linear vs. Null

Data simulation

The linear model:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, \ldots, n$$

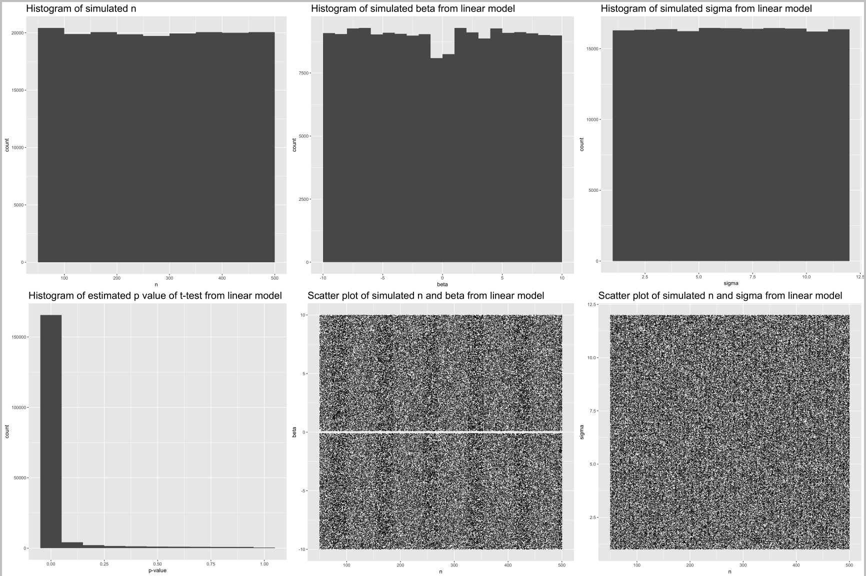
- $X \sim N[0,\ 1]$
- $\beta_0 = 0$
- $\beta_1 \sim U[-10, -0.1] \bigcup [0.1, 10]$
- $\varepsilon \sim N(0, \sigma^2)$ *where* $\sigma \sim U[1, 12]$
- $n = U[50, 500]$

The null model:

- $\beta_1 = 0$

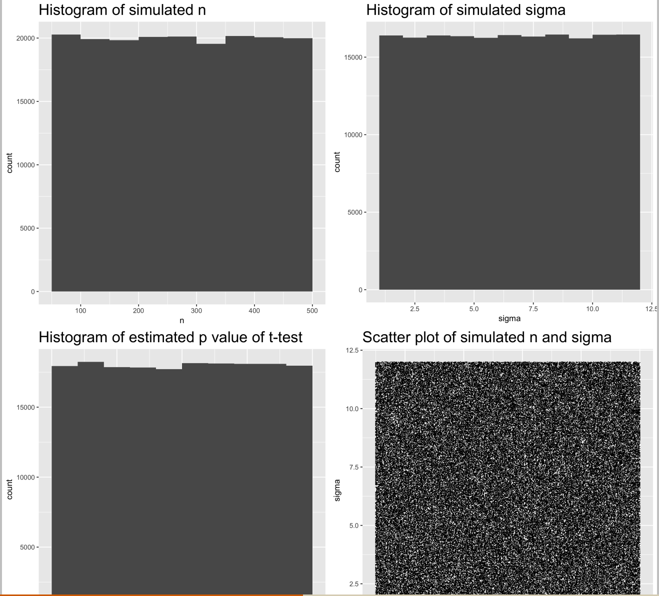
First Experiment: Linear vs. Null

Histogram of simulated parameters from linear model



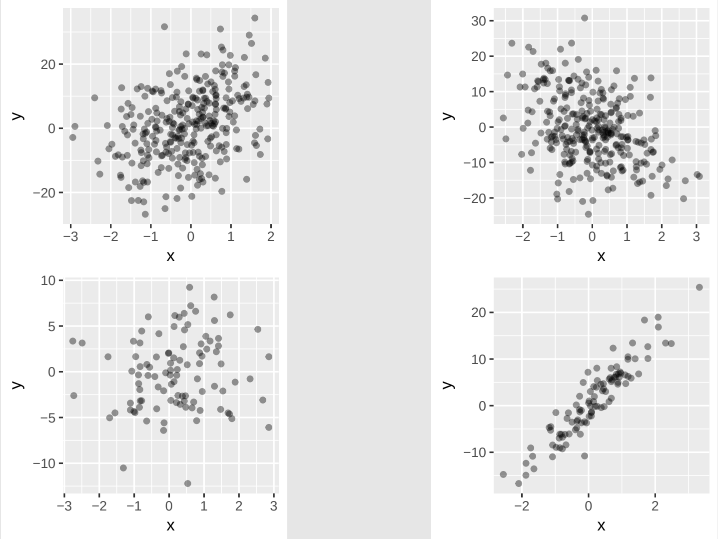
First Experiment: Linear vs. Null

Histogram of simulated parameters from null model



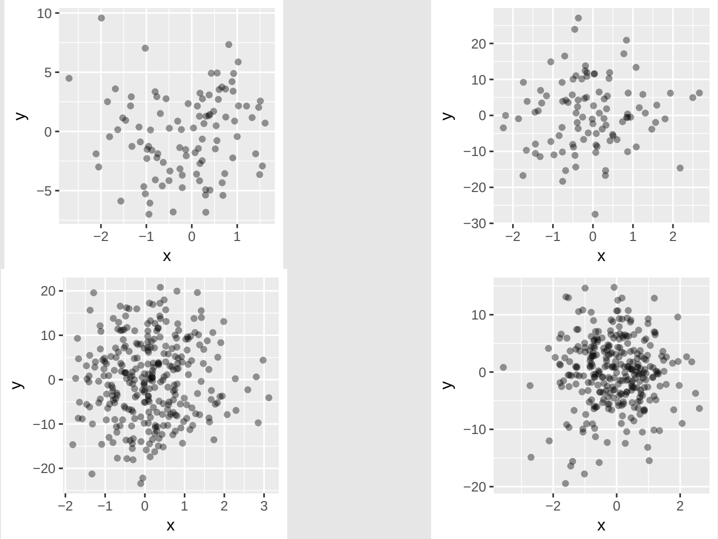
First Experiment: Linear vs. Null

Examples of scatter plots from the linear model



First Experiment: Linear vs. Null

Examples of scatter plots from the null model



First Experiment: Linear vs. Null

Convnets R code

```r
library(keras)
model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3),
                activation = "relu",
                input_shape = c(150, 150, 1)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3),
                activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 128, kernel_size = c(3, 3),
                activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_flatten() %>%
  layer_dense(units = 512, activation = "relu") %>%
  layer_dense(units = 1, activation = "sigmoid")
```

First Experiment: Linear vs. Null

Convnets structure

Model

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 148, 148, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_2 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_2 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_3 (Conv2D)	(None, 34, 34, 128)	73856
max_pooling2d_3 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten_1 (Flatten)	(None, 36992)	0
dense_1 (Dense)	(None, 512)	18940416
dense_2 (Dense)	(None, 1)	513

Total params: 19,033,601
Trainable params: 19,033,601
Non-trainable params: 0

Figure 7: convnets model structure

First Experiment: Linear vs. Null

Training and validation metrics of convnets

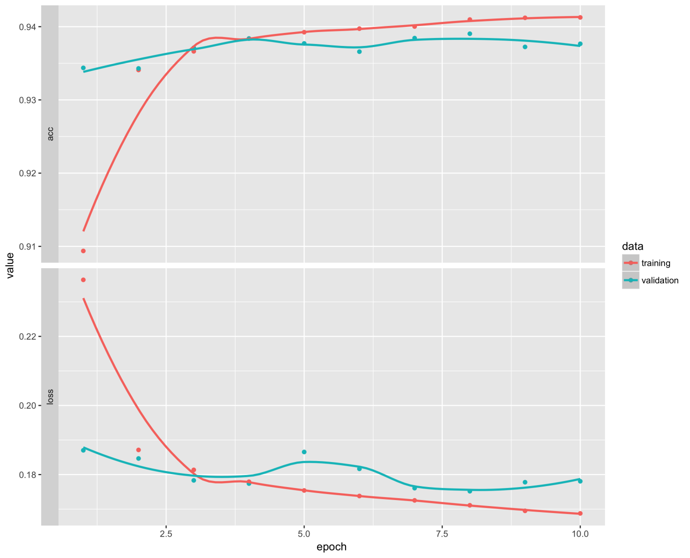


Figure 8: Training and validation metrics of convnets for the first experiment

First Experiment: Linear vs. Null

Convnets model selection

Tests	Linear	Null	Overall
4 epoch	0.892	0.984	0.938
6 epoch	0.889	0.986	0.937
8 epoch	0.896	0.981	0.939
10 epoch	0.904	0.971	0.938
5% t-test	0.921	0.949	0.935

Table 1: Performance of four checkpoints from the *convnets* model, and the 5% significant t-test, computed on the test set. Accuracy is reported for each class, and overall. There is a slight improvement as the number of epochs increases, with 10 epochs being reasonably close to the ideal t-test accuracy.

The 8th model is chosen according to the overall accuracy on the test set.

First Experiment: Linear vs. Null

Computer model performance calculation

The performance of the computer model for the Turk study data is tested in three steps:

- Re-generate the 70 “real plots” using the same data in Turk study (without null plots);
- Create a seperate test directory for the 70 “real plots” exclusively;
- The computer model’s predicted accuracy over the 70 “real plots” are recorded as the model’s performance.

First Experiment: Linear vs. Null

Human evaluation performance calculation

The conclusion of human evaluation is obtained differently from the computer's. Because human evaluated "lineup", not only the "real plots". The performance is tested in five steps:

- Count total number of evaluations made by human for one lineup (N) and the number of correct answers for that lineup (k);
- Obtain N and k for all 70 lineup;
- Calculate p-value associated with each real plot using the formula introduced in section 2 of @MM13;
- Draw conclusion: reject the null when the calculated p-value is smaller than α.
- The accuracy of the conclusions the 70 real plots is presenting for the human performance.

First Experiment: Linear vs. Null

Comparing results

Table 2: Accuracy of testing the 70 data plots evaluated by human computer and the conventional t-test.

Rank	Tests	No. of correct	Accuracy
1	Human 5%	47	0.6714
1	Human 2%	47	0.6714
2	T-test 5%	43	0.6143
3	Computer 2%	39	0.5571
4	T-test 2%	39	0.5571

First Experiment: Linear vs. Null

Aside discussion

It is possible that the best classification strategy found by convnets is in fact the convetional t-test.

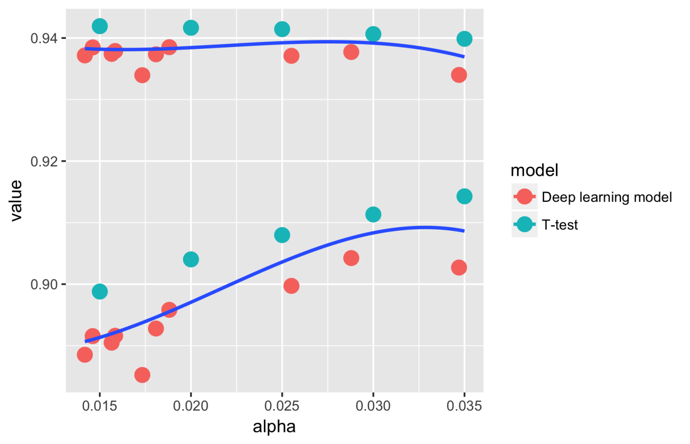


Figure 9: The estimated power and overall accuracy of T-test and convnets for test set

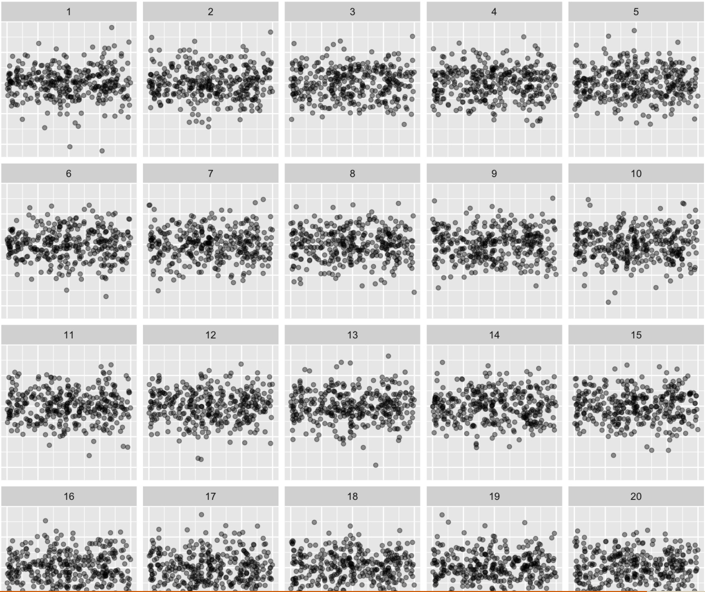
Second Experiment: Heteroskedasticity vs. Null

Human experiment setup

The experiment is to evaluate the human ability of reading heteroskedasticity from residual plots. It is rendered at Monash University, Melbourne Australia. The participants are all students or lecturers in this university. Four survey are randomly sent to 84 people by email, three of the survey consist of ten lineup questions, and the fourth survey has only four lineup questions. Only one lineup question appears in the survey twice, thus, we have 33 $(10 \times 3 + 4 - 1)$ distinct questions in total. A total number of 22 people have participated. Five people evaluated two surveys. One people selected four plots for each lineup, this person's response is removed from the data. In summary, we have 218 effective evaluations from 21 people.

Second Experiment: Heteroskedasticity vs. Null

An example of questions used in the human experiment



Second Experiment: Heteroskedasticity vs. Null

Heteroskedasticity simulation

The Heteroskedasticity model:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \;\; i = 1, \ldots, n$$

- $X \sim U[-1,\ 1]$
- $\beta_0 = 0$
- $\beta_1 \sim U[0.5,\ 1]$
- $\varepsilon \sim N(0,\ (aX + v)^2)$
- $a \sim U(-5,\ -0.05) \bigcup (0.05,\ 5)$
- $v \sim N(0,\ 1)$
- $ax + v - min(ax + v)$ when $min(ax + v) < 0$
- $n \sim U[50,\ 500]$

The null model:

- $\varepsilon \sim N(0,\ c)$
- $c = mean(ax + v)$

Second Experiment: Heteroskedasticity vs. Null

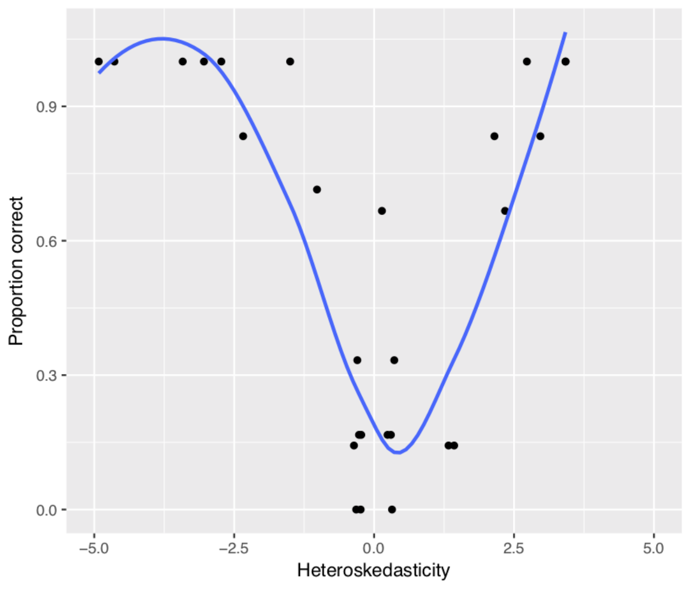
White test

To provide a reference level of how computer and human perform, a special case of White test is used in this experiment. The procedure of the White test [@IE17] is:

- Estimate OLS model for the data, obtain residuals ($\hat{u}$) and the fitted values ($\hat{y}$). Computer the squared OLS residuals ($\hat{u}^2$) and the squared fitted values ($\hat{y}^2$).
- Run an auxiliary regression as $\hat{u}^2 = \eta_0 + \eta_1\hat{y} + \eta_2\hat{y}^2 + error$, obtain the R-squared $R^2_{\hat{u}^2}$
- Calculate the LM statistic which follows χ^2_2 distribution
- Conclude based on p-values given certain α

Second Experiment: Heteroskedasticity vs. Null

Training and validation metrics of convnets



Second Experiment: Heteroskedasticity vs. Null

Convnets model selection

Second Experiment: Heteroskedasticity vs. Null

Human performance in the experiment

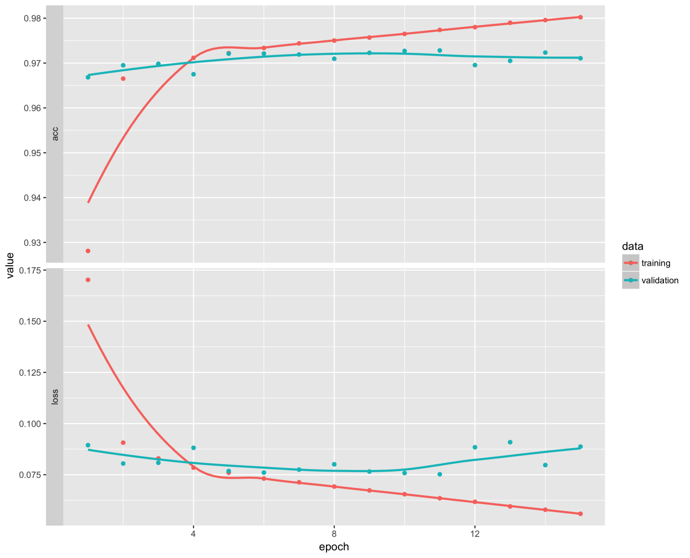


Figure 12: Proportion of correct answers for each lineup question against the

Second Experiment: Heteroskedasticity vs. Null

Comparing results

Table 3: Accuracy of testing the 27 data plots evaluated by human computer and the conventional white-test.

Rank	Tests	No. of correct	Accuracy
1	Computer 2%	25	92.59%
2	Human 5%	17	62.96%
2	White-test 5%	17	62.96%
3	White-test 2%	16	59.26%
4	Human 2%	15	55.56%

Materials

- The thesis, code and data is available on the github repository https://github.com/shuofan18/ETF5550
- Software used to conduct this research is R, Tensorflow, keras, tidyverse