AMSI
VACATION
RESEARCH
SCHOLARSHIPS
2018-2019

Visual Inference for Linear Mixed Models

John Su
Supervised by Dr. Emi Tanaka
The University of Sydney

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Australian Government
Department of Education and Training

Abstract

Linear mixed models can be used to account for complex, correlated structures between observations. Consequently, they are used widely in many scientific disciplines, including psychology, medicine and agriculture. Model diagnostics, and more specifically, model selection are important components in any model fitting process. However, these components are poorly understood and challenging in comparison to the diagnosis of the fit of linear (fixed) models due to questionable asymptotic properties and biased estimations. Conventional model inference or diagnosis often rely on hypothesis testing or examination of residual plot but these may fail to provide a meaningful understanding of the data-generation process. Buja et al. (2009) suggested the use of visual inference methods for model diagnostics as an alternative. Visual inference involves randomly embedding a plot of the true data within a line-up of null plots, generated from a carefully chosen null generating mechanism that mimics the data generation process under the null hypothesis. A number of human observers will then attempt to find the odd-plot out. This visual inference has added benefits of pin-pointing characteristics of the "odd-plot" and a more graspable diagnosis of the fitted model. In this report, we give key reviews of linear mixed models and the use of the variogram as a model diagnostic tool for detection of spatial dependency. The original contribution of this report is the qualitative comparison of the lineup protocol to a conventional hypothesis testing approach based on a simulation study from the analysis of two wheat breeding trials.

1 Introduction

1.1 Motivation

The conventional approach to model selection for linear models is to use a stepwise, forward or backward procedure under a certain criterion. These include the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and hypothesis tests. However, such an approach necessarily leads to a narrow conclusion which only informs the analyst whether or not the null hypothesis may be rejected. Furthermore, these tests may also require unrealistic assumptions about the asymptotic distribution of the test statistic. Although graphical plots offer an alternative solution, a single graphical plot can be easily misinterpreted (Loy et al., 2017) especially in the context of linear mixed models were correlation between data points is frequently encountered.

Section 2 will define linear mixed models as well as provide the background of both visual inference

methods and conventional tests. Section 3 will demonstrate application of both the residual maximum likelihood ratio test and the lineup protocol to select datasets from the 2017 CAIGE wheat yield trials. Section 4 will provide a qualitative comparison and discussion on the efficacy of both methods based on results from Section 3.

1.2 Linear Mixed Models

Linear models typically only contain a single random effect - the error term. Linear mixed models extend this model to include random effects which are capable of modelling complex correlation structures. These models are also known by other names such as linear mixed-effects model, hierarchical models and so on. Hereafter in this report, we will refer to them as linear mixed models. These are given in the form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{b} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{y}$ is an $n \times 1$ vector of observations (n is the number of observations for each trial); $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, $\mathbf{X}$ is a $n \times p$ design matrix corresponding to the fixed effects; $\mathbf{b}$ is a $q \times 1$ vector of random effects; $\mathbf{Z}$ is the $n \times q$ design matrix for the random effects and $\boldsymbol{\varepsilon}$ is the $n \times 1$ vector of random errors. Often, we assume that $\mathbf{b}$ and $\boldsymbol{\varepsilon}$ are uncorrelated and both are normally distributed with zero mean and variance matrix:

$$\begin{bmatrix} \boldsymbol{G}(\gamma) & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}(\phi) \end{bmatrix},$$

where $\boldsymbol{G}(\gamma)$ is a $q \times q$ covariance matrix for the random effects, $\boldsymbol{R}(\phi)$ is a $n \times n$ covariance matrix for the error terms, γ and ϕ are vectors of variance parameters. Usually there are k sets of random effects $\mathbf{b} = \{\mathbf{b}_i\}$ where $\mathbf{b}_i$ is a $q_i \times 1$ vector of the i-th set of random effects for $i = 1, ..., k$ and $\sum_{i=1}^{k} q_i = q$. Typically, it is assumed that the k sets of random effects are mutually independent so that $\boldsymbol{G}$ is block diagonal. For this report we assume, as done in many applications, that $\text{Var}(\mathbf{b}_i) = \boldsymbol{G}_i$ has a scaled identity structure, i.e. $\boldsymbol{G}_i = \gamma_i \boldsymbol{I}_{q_i}$. Likewise, we assume that the errors are independent and identically distributed (i.i.d) $\boldsymbol{R} = \sigma^2 \boldsymbol{I}_n$ and in this case, $\phi = \sigma^2$. A consequence of the above assumptions is that the distribution of the data is normal with mean $\boldsymbol{X}\boldsymbol{\beta}$ and variance $\text{Var}[\boldsymbol{y}] = \boldsymbol{H} = \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^\top + \boldsymbol{R}$.

1.3 Inference

Determination and verification of the fit of a statistical model to a given data is critical in evaluating the reliability of statistical inference using the model. Although a multitude of tests and procedures have been proposed for linear mixed models, there remain issues which continue to plague model selection and model diagnostic processes.

1.3.1 Model selection

In this report, we examine only the selection of random effects for a linear mixed model. For a hypothesis testing approach, suppose for one set of random effects, $\mathbf{b}_i \sim N(\mathbf{0}_{q_i}, \gamma_i \boldsymbol{I}_{q_i})$, we test $H_0 : \gamma_i = 0$ vs. $H_1 : \gamma_i > 0$. Under the null hypothesis, this implies that $\mathbf{b}_i = \mathbf{0}_{q_i}$. The test statistic is formulated based on the residual maximum likelihood ratio defined below.

Definition 1.1. For a comparison of two nested models M_0 and M_1 where M_1 has r more parameters than M_0, the residual maximum likelihood ratio test (REMLRT) test statistic is given by

$$D = 2(l_{M_1} - l_{M_0}),$$

where l_{M_1} and l_{M_0} are the residual log-likelihood for models M_1 and M_0 respectively. The REMLRT test is only appropriate if the models M_1 and M_0 share the same fixed effects.

In general, the test statistic has an approximate chi-squared distribution with r degrees of freedom. However, when testing for the inclusion of a variance parameter, the parameter itself lies on the boundary of the parameter space and the asymptotic distribution of the likelihood ratio test is no longer chi-squared (Smith (1999), Loy et al. (2017)). Approximations have been suggested to resolve the issue in some cases such as the use of a 50:50 mixture of χ^2_{r+1} and χ^2_r distributions where r is the number of variance parameters in the null hypothesis. The p-value in this case is then calculated as

$$p = 0.5\delta_0 + 0.5P[\chi^2_1 > D],$$

where $\delta_0 = 1$ when $D = 0$ and $\delta_0 = 0$ when $D \neq 0$.

Reduction in bias from using this approximation when compared with the naive test is small. However, in certain complicated scenarios where many random effects are being introduced simultaneously, the degree of asymptotic bias of the naive test can be substantial (Stram and Lee, 1994).

Ultimately, no single approximation works for all situations and simulation studies would need to be done to make the proper adjustments to the reference distribution each time (Loy et al., 2017).

1.3.2 Model diagnostics

Model diagnostics focus on assessing the underlying assumptions used in the formulation of the model. This includes homogeneity of residual variance, linearity and normality of the random effects. Traditional approaches include use of single residual plots, QQ plots or frequentist tests such as the Shapiro-Wilk test and the Anderson-Darling test which rely on asymptotic distributions. Although this report will not focus on problems in model diagnostic methods, they bear a striking similarity to problems in model selection methods which are examined here.

1.4 Visual Inference

Visual inference and classical statistical inference share common principals but diverge in what outcomes they can achieve. Whilst there are different protocols in use, they follow the same principles. In visual inference, null plots are drawn from data simulated using a model consistent with the null hypothesis. This set of null plots will then constitute the 'distribution'. The 'test statistic' for visual inference corresponds to the null and true plots shown to the observer. Human observers are then asked to compare the true data plot with the null plots. Figure 1 provides a succinct visual comparison of conventional tests and visual tests. The idea is that if the human observer is unable to identify the true plot amongst the null plots then the null model is sensible (Loy et al., 2017).

Visual inference uses human cognition in place of statistical tests but more importantly there is no pre-specification of the range of visual discoveries possible whereas in quantitative tests there is an explicit prior specification of the possible 'discoveries' in the form of the null hypothesis (Buja et al., 2009). This provides visual inference with an unparalleled degree of flexibility to further explore hidden structures in the data as well as assess which part of a null hypothesis is violated. Unlike conventional quantitative tests, visual inference do not rely on asymptotic reference distributions and thus bypass the need to make assumptions which may not be realistic or arbitrary decisions regarding degrees of freedom and group sizes.

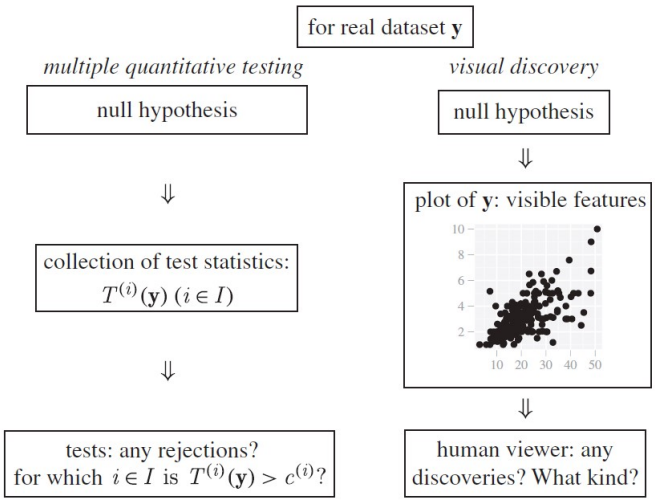


Figure 1: Comparison of visual inference tests with conventional tests (Buja et al., 2009).

1.4.1 Lineup Protocol

The lineup protocol involves generating 19 null plots and randomly inserting the true data plot amongst them. All contextual information such as graph title and axes markings are removed. The set of 20 plots is then presented to a human observer who is asked to identify which plot is the most different to others. The human observer may also be asked to explain their choice though this is not necessary and depends on the aim of the exercise. Majumder et al. (2013) proposed a method to calculate a visual p-value which is defined below.

Definition 1.2. Let m be the total number of plots shown to the observer, K be the number of independent observers and X be the number of observers picking the test statistic from the lineup. Under the null hypothesis $X \sim \text{Binom}_{K,1/m}$ since each observer has a $1/m$ chance of picking the correct plot from the lineup. Therefore, the p-value of a lineup of size m evaluated by K observers is given as:

$$P(X \geq x) = 1 - \text{Binom}_{K,1/m}(x-1) = \sum_{i=x}^{K} \binom{K}{i} \left(\frac{1}{m}\right)^{i} \left(\frac{m-1}{m}\right)^{K-i}.$$

Calculation of the visual p-value requires recruiting a large pool of human observers to evaluate the lineup plots. Loy et al. (2017) described one option to accomplish this through the recruitment of observers using the Amazon MTurk service which sends out lineup plots as HITs (Human Intelligence Tasks) to participants. Resourcing issues were a major barrier to our ability to recruit enough participants.

Definition 1.3. Let V_θ be the lineup protocol visual test. The power of a visual test is defined as

the probability of rejecting the null hypothesis for a given parameter θ. The lineup protocol depends on the observers' evaluation. Therefore X, the number of observers who identify the true plot, affects the estimation of power which is estimated by

$$\widehat{\text{Power}}_{V,K}(\theta) = 1 - F_{X,\theta}(x_\alpha - 1),$$

where $F_{X,\theta}(x_\alpha - 1)$ is the distribution of X and x_α such that $P(X \geq x_\alpha) \leq \alpha$. Since the null hypothesis is $X \sim \text{Binom}_{K,1/m}$ we have:

$$\text{Power}_V(\theta, K) = 1 - \text{Binom}_{K,1/m}(x_\alpha - 1).$$

The above definition is only suitable for comparing visual tests with conventional tests. Analysts who choose to use the lineup protocol are often confronted with the problem of choosing which plots to use in the protocol. This is similar to choosing which test statistic to use and often one type of plot will be better than another. In this case, power is a measurement of the relative ease of the plot type to distinguish the true plot from the null plots from the perspective of the observers.

2 Simulation study with CAIGE wheat trial data

2.1 The data

The CAIGE wheat-durum yield trials are an annual evaulation of different germplasms of wheat conducted at various sites in Australia. Datasets from the Balaklava and Roseworthy trials from the 2017 CAIGE trials were used. The Balaklava trial evaluated the yield of wheat on a South Australian farm with plots organised into 28 rows and 12 columns. The Roseworthy trial was also conducted in South Australia and the plots were arranged into 16 rows and 24 columns. The purpose of yield evaluation trials is to identify the best performing genotype for wheat yield. Data collected include the genotype of the wheat and the row-column coordinate of the plot within the field trial. A random row or random column effects with a simple i.i.d. variance structure is commonly significant in these trials owing to trial management practices, such as harvesting in a serpentine manner in a row or column direction. More complex spatial analysis typically include a seperable autoregressive processes, however these are omitted in this report due to time and software restrictions.

2.2 The Balaklava Trial

In this part we performed extensive simulations to compare the effectiveness of the lineup protocol with the REMLRT. To facilitate this process, ad hoc functions were written in R (see Appendix A).

	Models under comparison		
p-value threshold	$M_0 : y_{ijk} = \tau_0 + \mu_i$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j$	$M_0 : y_{ijk} = \tau_0 + \mu_i$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_k$	$M_0 : y_{ijk} = \tau_0 + \mu_i + \mu_k$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k$
0.05	9	196	16
0.01	4	193	32

Table 1: Null model (M_0), alternative model (M_1), yield (y_{ijk}), overall mean (τ_0), genotype random effect (μ_i), row random effect (μ_j), column random effect (μ_k). The number of datasets which passed the REMLRT are listed. Datasets are simulated using a model based on the Balaklava trial with the original standard deviations for the random effects.

As it is unrealistic to perform the lineup protocol on all simulated datasets, a random dataset was chosen for testing using the lineup protocol.

The first simulation most closely reflects the variations that may be present in random effects found in agricultural datasets as it makes use of the original standard deviations obtained by fitting the following model to the data from the Balaklava trial:

$$y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k + \varepsilon_{ijk},$$

where y_{ijk} is the yield response variable, τ_0 is the overall mean, μ_i is the genotype random effect, μ_j is the row random effect, μ_k is the column random effect and ε_{ijk} is the error term. This model was then used as the "true" model to generate the 200 simulated datasets for use in analysis. Appendix B.1 shows the code used to efficiently simulate large number of datasets. Table 1 gives the number of datasets which are able to satisfy the p-value threshold when performing a comparison between a null model and an alternative model using the REMLRT. Only 15 of the 200 datasets registered the Row and Column random effect as significant under REMLRT at the 0.01 p-value threshold. This gives the REMLRT an experimental power of 0.075 when testing with the 0.01 p-value threshold.

For comparison, we investigated the 36^{th} simulated dataset to perform the lineup protocol to compare the models $M_0 : y_{ijk} = \tau_0 + \mu_i$ and $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_k$. Dataset 36 was chosen because it was the dataset with the lowest p-value (0.0127) without passing the 0.01 threshold (see Appendix C for the lineup plot codes). Figure 2 shows the resulting lineup. Instead of scatter plots, we chose boxplots to enhance the underlying trends. The true plot is in panel $\sqrt{11^2 + 23}$. This plot is not

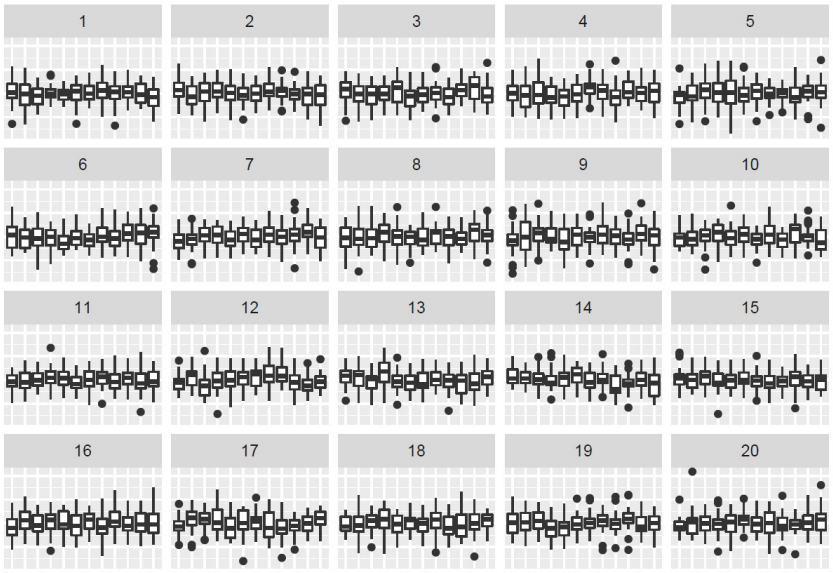


Figure 2: Lineup testing for the Column variance parameter where the null model only had a Genotype random effect. Level-1 residuals are plotted on the y axis while the Column numbers are plotted as factors on the x axis. The true plot is located at $\sqrt{11^2 + 23}$.

easily identifiable and therefore matches the result given by the REMLRT. Despite this conclusion, one interesting feature to note in the true plot is that there appears to be an underlying wave-like trend. This could be suggestive of a spatial dependency in the columns which would favour the inclusion of a random column effect. However, the trend is not obvious and in the absence of a large pool of independent observers it is difficult to assess the significance of this visual characteristic. On the other hand, Figure 3 presents the lineup protocol of dataset 160 which is the dataset with the lowest p-value and which failed to pass the 0.01 threshold for the test between the models $M_0 : y_{ijk} = \tau_0 + \mu_i + \mu_k$ and $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k$. Unlike Figure 2, Figure 3 makes use of a scatter plot as opposed to boxplots. Here, the render of more individual data points greatly reduces the readability of the plots. Therefore, although the scatter plot is able to represent the data more completely, it appears that boxplots are much better at showing underlying trends. In both instances, the lineup protocol was able to match the REMLRT result.

The second simulation utilised the same model as the first simulation but with inflated Row and Column random effects (the standard deviation of both random effects was set to 0.09). Table 2 shows the number of datasets which were able to pass the relevant p-value threshold for the REMLRT. This time, a dataset which passed the REMLRT at the 0.01 p-value threshold was chosen for comparison using the lineup protocol. Figure 4 shows the lineup protocol for dataset 1 which satisfies this criteria.

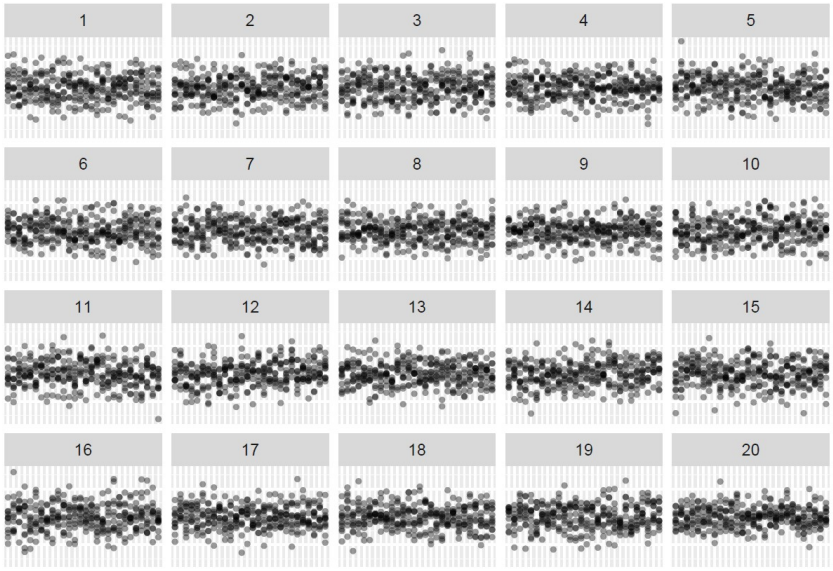


Figure 3: Lineup testing for Row variance parameter where the null model only had a Genotype and Column random effect. Level-1 residuals are plotted on the y axis while the Row numbers are plotted on the x axis. The true plot is located at $\sqrt{169} - 1$.

	Models under comparison	
p-value threshold	$M_0 : y_{ijk} = \tau_0 + \mu_i$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_k$	$M_0 : y_{ijk} = \tau_0 + \mu_i + \mu_k$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k$
0.05	163	140
0.01	143	115

Table 2: Results for the REMLRT on the 200 simulated datasets from the Balaklava trial with inflated Row and Column random effects. Notation used is the same as Table 1.

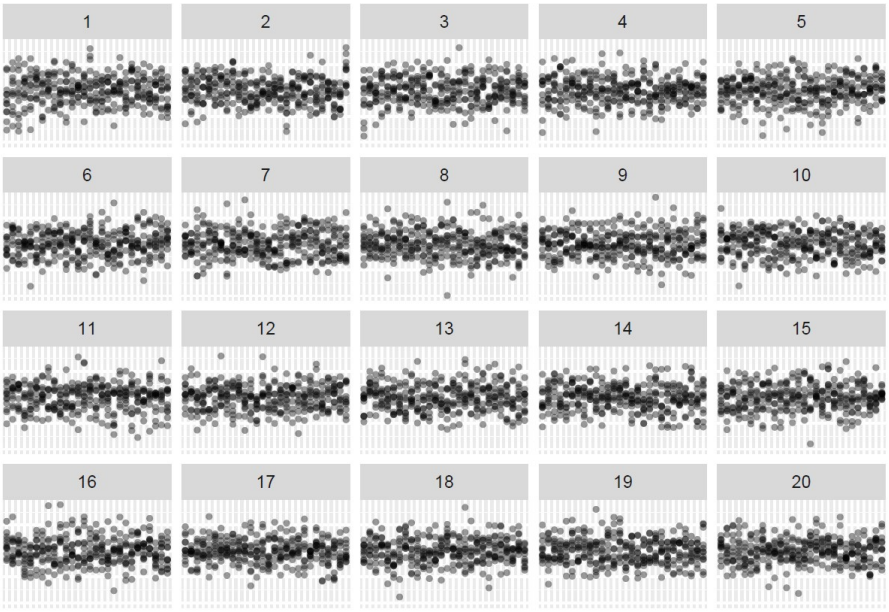


Figure 4: Lineup testing for both Row and Column random effects. The true position is at $\frac{143+1}{72}$*. The Row numbers are plotted on the* x *axis while the level-1 residuals are plotted on the* y *axis.*

Again, even with inflated values for both Rows and Columns, the power of the lineup protocol appears to be hindered by the plot type. The lineup for the same dataset with columns on the x axis is not included but a similar result was observed.

2.3 The Roseworthy Trial

The results of the simulations based on the Balaklava trial have shown that plot choice is closely tied with the power of the lineup protocol. In this part we performed further simulations based on the Roseworthy Trial with a different graphic choice. The plot type we chose was the *variogram* (see Appendix D.1 for definition). At the time of writing, there was no existing code for the calculation of the experimental variogram for linear mixed models fitted using the `lme4` package so an ad hoc function was written (see Appendix D.2). Raster plots were chosen to represent the experimental variogram (see Appendix D.3 for more details and code). The simulation approach in this part is similar to that used for the Balaklava trial.

In the first simulation, the following model:

$$y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k + \varepsilon_{ijk},$$

was used to simulate a dataset with an inflated Row and Column random effect (standard deviation

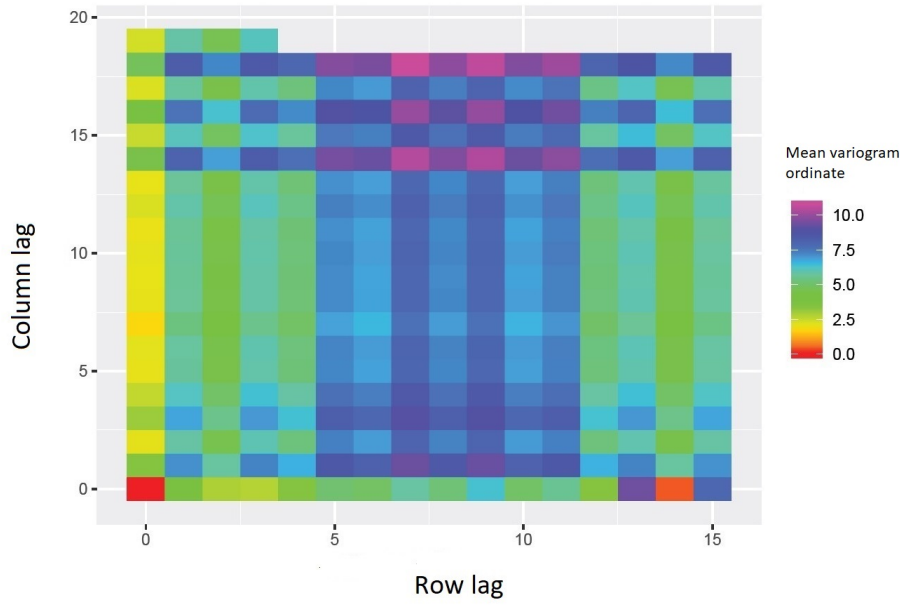


Figure 5: Variogram of simulated dataset with the original Genotype random effect and inflated Row and Column random effects. The top 20% *of the datapoints with the highest lags were excluded.*

increased to 2). Using inflated random effects would clearly reveal to us what a variogram would look like if a very obvious Row and Column random effect was present in a dataset. Figure 5 shows the resulting variogram. An immediately obvious feature is the grid-like pattern that is present throughout the whole figure. Furthermore, there appears to be a repetition of the range of colours present across the displacements. This suggests that the sample variogram has an underlying wave-like trend across the displacements which is confirmed by looking at scatter plots of the profiles of the variogram (see Appendix E). Furthermore, Appendix F shows that this grid-like pattern can be decomposed into separate striations and provides evidence for the proposition that the addition of random effects to the model correspond to an additive combination of the individual visual characteristics of the random effects in the variogram.

In our second set of simulations, the same model as the first simulation was used with the original Genotype, Row and Column standard deviations for their respective random effects. This model was used to generate a single data set which we denote as the "true" dataset. Table 3 shows the results from applying the REMLRT to this simulated dataset. Although the REMLRT was able to recognise that the Column effect was significant, the REMLRT failed to recognise that the row effect was significant at both the 0.01 and 0.05 p-value threshold. However, it must be noted that this failure is a very borderline conclusion. Figure 6 depicts the lineup protocol produced where the null model only

	Models under comparison		
	$M_0 : y_{ijk} = \tau_0 + \mu_i$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j$	$M_0 : y_{ijk} = \tau_0 + \mu_i$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_k$	$M_0 : y_{ijk} = \tau_0 + \mu_i + \mu_k$ $M_1 : y_{ijk} = \tau_0 + \mu_i + \mu_j + \mu_k$
p-value	0.1753	7.370×10^{-8}	0.05307

Table 3: Results for the REMLRT test on the simulated dataset from Roseworthy with original Genotype, Row and Column random effects. Notation used is the same as Table 1.

contains the Genotype random effect (see Appendix G for follow-up lineup plots of the column profile of the variograms). Notice the discernible grid-like pattern in the true plot which is not present in the null plots. This pattern bears a resemblance to the pattern we saw in Figure 5. The relative lack of visible patterns and uniform distribution of colours in the null plots enhance the ease with which the true plot stands out. Without the null plots, it may be more difficult to recognise this visual pattern and come to the same conclusion. Unlike the REMLRT results which only give us a suspicion that perhaps we should not be too quick to accept the null hypothesis, the lineup protocol provides visual evidence which can be used to point us in the right direction for model selection especially when ambiguous data is present. In this case, the pattern observed may hint at an underlying spatial dependency in the data. This is far more useful for model building then simply deciding whether the null hypothesis is to be rejected.

In our third and final set of simulations, a "true" dataset was again simulated using the same set of random effects present in the first set of simulations. Instead of assuming normality for all the random effects, we instead assumed that the random effects are distributed according to a chi-squared distribution with the degrees of freedom determined by the original estimated standard deviations. The error terms are still assumed to be normally distributed. The purpose of this final simulation is to compare the efficacy of the lineup protocol and the REMLRT in situations where the normality assumption is no longer appropriate. To perform the simulation itself, a modified version of the earlier `mansim` function was used (see Appendix B.3). Figure 7 shows the variogram for the lineup protocol where the null model assumes normality for all random effects while the "true" dataset was generated with the Column and Row random effects distributed as a chi-squared distribution. The true position is at $2^2 + 5$. Notice how even after the true plot's position is revealed it is very difficult to distinguish between the true dataset and the null data sets. Figure 8 depicts the variogram which compares the "true" dataset simulated with inflated Column and Row random effects with a chi-

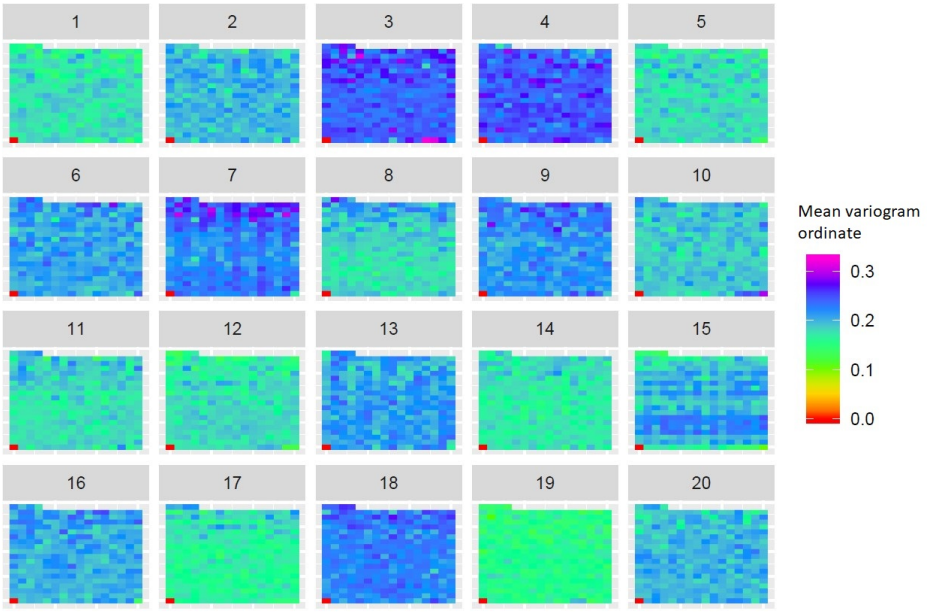


Figure 6: Lineup variogram testing for both Column and Row random effects. The true position is located at $3 \times 7 - 6$.

squared distribution. The true position is at $8^2 - 50$. Similar to Figure 5 (where the "true" dataset which assumes normality for all inflated random effects), there is also a grid-like pattern present. However, unlike Figure 5, the "true" data plot in Figure 8 is much flatter than the "true" data plot followed an underlying normal distribution. This again exemplifies the usefulness of the lineup protocol in preventing misinterpretation of single plots through comparison with the null plots.

3 Discussion

3.1 Comparison with REMLRT

The lineup protocol was able to match the result of the REMLRT in almost all of the datasets we investigated. Although we were unable to test all the simulated datasets and calculate visual p-values, we can qualitatively infer that the result of the lineup matches the REMLRT result. Further studies making use of actual participants (e.g. the Amazon MTurk service) would be beneficial to derive actual p-values from the lineup protocol for comparison with REMLRT. There was only one instance where the lineup protocol was able to qualitatively detect a random effect which was missed by the REMLRT (see Figure 6 and Table 3). Although the result is very borderline, it is nevertheless a glimpse of the potential of the lineup protocol as an alternative testing method.

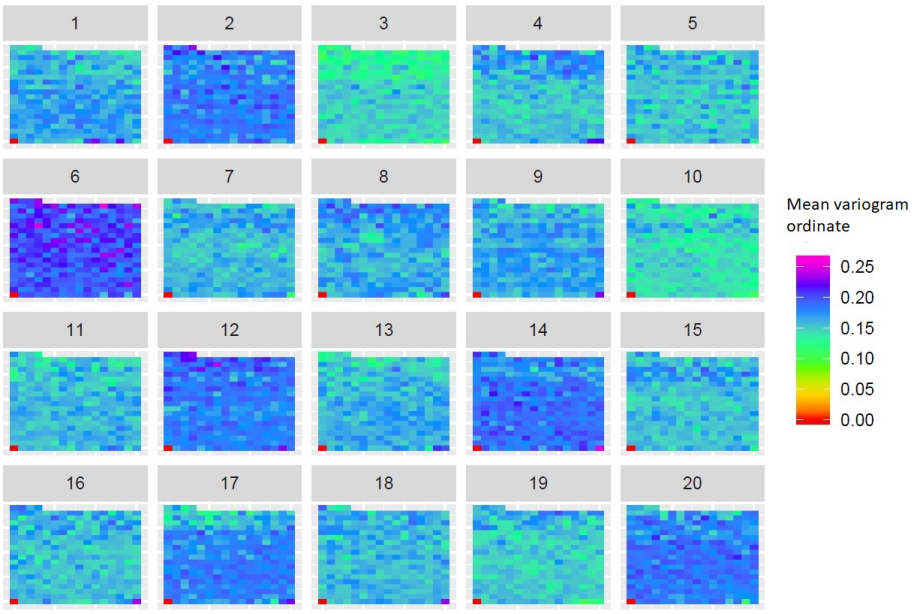


Figure 7: Lineup variogram where the Column and Row random effects follow a chi-squared distribution. The true position is at $2^2 + 5$.

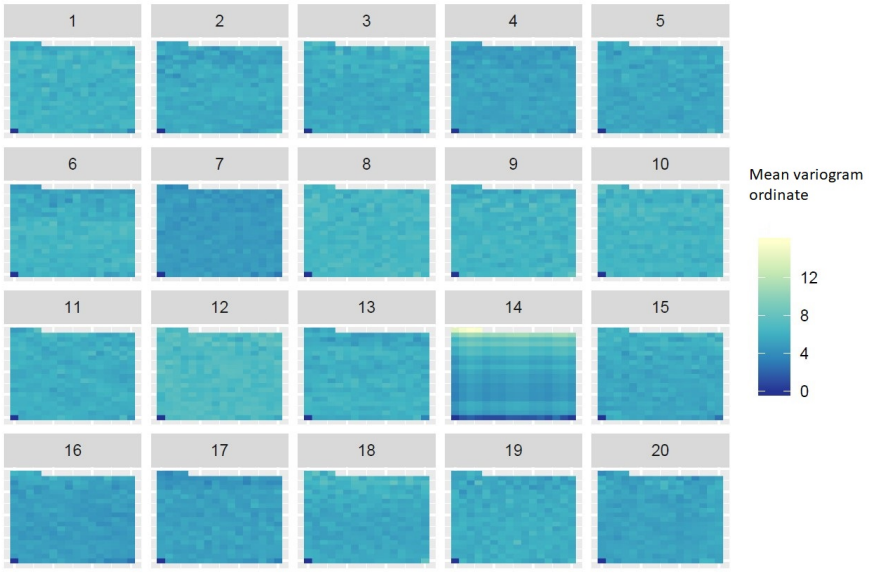


Figure 8: Lineup variogram where the Column and Row random effects are inflated and follow a chi-squared distribution. The true position is at $8^2 - 50$.

3.2 Plot choice

Across all the simulations, it is clear that the power of the lineup protocol is closely related to plot choice. Without the correct choice of plots to suit the situation or context, even if the random effect is very obvious (i.e. has high standard deviation), the visual representation of this effect will not be clear. Although Majumder et al. (2013) gave an extensive list of different types of plots to be used to test various aspects of a linear regression model, it was not extended to linear mixed models. As of writing, there is still no unified approach to measure how the power of the lineup protocol changes when different types of plots are selected (Majumder et al., 2013). Further studies into the power of the tests and their dependence on plot choice are needed.

4 Conclusion

This article carried out a selective comparison of the lineup protocol with the REMLRT. Two separate simulation studies were carried out on the Balaklava and Roseworthy datasets from the CAIGE 2017 Wheat Yield trials. A qualitative comparison was then made by applying these two testing methods to simulated datasets. Although the results were promising, they are by no means conclusive. The aim of this report is not to rebuke conventional tests in favour of the lineup protocol or any other visual inference method. Rather, visual inference methods should form another safety net for analysts. Further work should explore the application of these tests to model diagnosis problems and carry out quantitative comparisons of visual inference methods with conventional tests.

5 Acknowledgements

I would like to thank the support and guidance given by my supervisor Dr. Emi Tanaka without whom this project would not have been possible. I would also like to thank the University of Sydney and AMSI for providing undergraduate students with the opportunity to undertake a research project and present it at AMSIConnect in Melbourne. R (R Core Team, 2018) was used to conduct all statistical simulations and analysis. lme4 (Bates et al., 2015) and HLMdiag (Loy and Hofmann, 2014) were used for model fitting and calculation of diagnostics, respectively. ggplot2 (Wickham, 2016) and nullabor (Wickham et al., 2018) were used to produce visualisations.

References

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E. K., Swayne, D. F., and Wickham, H. (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.

Loy, A. and Hofmann, H. (2014). HLMdiag: A suite of diagnostics for hierarchical linear models in R. *Journal of Statistical Software*, 56(5):1–28.

Loy, A., Hofmann, H., and Cook, D. (2017). Model Choice and Diagnostics for Linear Mixed-Effects Models Using Statistics on Street Corners. *Journal of Computational and Graphical Statistics*, 26(3):478–492.

Majumder, M., Hofmann, H., and Cook, D. (2013). Validation of visual statistical inference, applied to linear models. *Journal of the American Statistical Association*, 108(503):942–956.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, 58(8):1246–1266.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Smith, A. (1999). *Multiplicative mixed models for the analysis of multi-environment trial data*. PhD thesis.

Stram, D. and Lee, J. W. (1994). Variance Components Testing in the Longitudinal Mixed Effects Model. *Biometrics*, 50(4):1171–1177.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

Wickham, H., Chowdhury, N. R., Cook, D., and Hofmann, H. (2018). *nullabor: Tools for Graphical Inference*. R package version 0.3.5.

Appendices

A Code for REMLRT

R function to test a model with only the genotype random effect against a model with genotype and row random effects. The code returns the number of datasets which pass the REMLRT at the 0.01 p-value threshold and the 0.05 p-value threshold. Similar R functions were also written to perform the same function to compare other nested models. In all REMLRT tests used in this paper the naive unadjusted degrees of freedom is used.

```
p.gr = function(dat){
        nonaddrow = function(sdat){
                nullmod= lmer(Yield~1+(1|Geno),data=sdat)
                altmod = lmer(Yield~1+(1|Geno)+(1|Rowf),data=sdat)
                val = 2*(logLik(altmod)-logLik(nullmod))
                p.val=pchisq(val,df=attr(logLik(altmod),"df"), lower.tail=F)
                return(as.numeric(p.val))
        }
        n = length(unique(dat$.n))
        pvals=vector(mode="numeric",length=n)
        for (i in 1:n){
                simdat = dat[which(dat$.n==i),]
                p = nonaddrow(simdat)
                pvals[i] = p
        }
        o = length(which(pvals<0.01))
        f = length(which(pvals<0.05))
        newlist = list("zerofive" = f, "zeroone" = o, "pvals" =pvals)
        return(newlist)
}
```

B Code for data simulation

B.1 repsim

R function to simulate data. This function allows the user to choose the model from which the data is simulated (`mod`), set the number of simulated datasets required (`r`), choose whether or not they want to return the level-1 residuals of the model after fitting it to the simulated datasets and set the seed.

```
repsim = function(mod, r, dat, resid = F, seed){
        set.seed(seed)
        df.sim.raw = simulate(mod, nsim = r)
        if (resid == F){
                df.sim = df.sim.raw
                df.sim = do.call("cbind",df.sim)
                df.sim = melt(df.sim)[,-1]
                names(df.sim)=c(".n","Yield")
                df.sim$.n = as.numeric(str_extract(df.sim$.n,"\\d+"))
                df.sim$Rowf = rep(dat$Rowf, rep=r)
                df.sim$Columnf = rep(dat$Columnf, rep =r)
                df.sim$Geno = rep(dat$Geno, rep=r)
                return(df.sim)
        }
        if (resid == T){
                lm.refit = lapply(df.sim.raw, refit, object = mod)
                df.sim.res = lapply(lm.refit, HLMresid, level=1,
                 type="EB", standardize =T)
                df.sim.res = do.call("cbind",df.sim.res)
                df.sim.res = melt(df.sim.res)[,-1]
                names(df.sim.res) =c(".n","res")
                df.sim.res$.n = as.numeric(str_extract(df.sim.res$.n,
                 "\\d+"))
                df.sim.res$Rowf = rep(dat$Rowf,rep=r)
                df.sim.res$Columnf = rep(dat$Columnf,rep=r)
                df.sim.res$Geno = rep(dat$Geno, rep=r)
```

```
                return(df.sim.res)
        }
}
```

B.2 mansim

R function to simulate data. This function allows the user to set the number of simulated datasets required (`r`), set the standard deviation of each of Genotype, Row and Column random effects or remove them completely from the model and set the seed. Note the user must insert the "full" model into `mod` before being able to pick which variables they want to remove for the generation process. By default, the function will assume that the variable will be included with the original estimated standard deviation.

```
mansim = function(mod, r, dat, G=0, R=0, C=0, seed){
        ngeno = nlevels(dat$Geno)
        nrow = nlevels(dat$Rowf)
        ncol = nlevels(dat$Columnf)
        mu = fixef(mod)
        vc = VarCorr(mod)
        ressd <- attr(vc, "sc") # Extract out residual sd
        df.sim.raw = data.frame()
        for (i in 1:r){
                set.seed(seed+i)
                if (G>0) {
                        geno_eff = rnorm(ngeno, 0, G)
                } else if(G==0) {
                        geno_eff = rnorm(ngeno, 0, vc$Geno[1])
                } else {
                        geno_eff = rep(0, ngeno)
                }
                if (R> 0) {
                        row_eff = rnorm(nrow, 0, R)
                } else if(R==0) {
                        row_eff = rnorm(nrow, 0, vc$Rowf[1])
```

```
                } else {
                        row_eff = rep(0, nrow)
                }
                if (C>0) {
                        col_eff = rnorm(ncol, 0, C)
                } else if(C==0) {
                        col_eff = rnorm(ncol, 0, vc$Columnf[1])
                } else {
                        col_eff = rep(0,ncol)
                }
                df1 = dat
                df1 = mutate(df1,Yield = mu +geno_eff[as.numeric(Geno)]+
                        row_eff[Rowf]+col_eff[Columnf]+
                        rnorm(nrow*ncol,0,ressd))
                        df.sim.raw=rbind(df.sim.raw, df1)
        }
        df.sim.raw$.n = rep(1:r, each=(nrow*ncol))
        df.sim = df.sim.raw
        return(df.sim)
}
```

B.3 chi_sim

A modified version of `mansim` which simulates data that assumes chi-squared distributions for all random effects.

```
chi_sim = function(mod, r, dat, G=0, R=0, C=0, seed){
        ngeno = nlevels(dat$Geno)
        nrow = nlevels(dat$Rowf)
        ncol = nlevels(dat$Columnf)
        mu = fixef(mod)
        vc = VarCorr(mod)
        ressd <- attr(vc, "sc") # Extract out residual sd
        df.sim.raw = data.frame()
```

```
for (i in 1:r){
        set.seed(seed+i)
        if (G>0) {
                geno_eff = rchisq(ngeno, df = 1/2 * G^2)
        } else if(G==0) {
                geno_eff = rchisq(ngeno, df = 1/2 * (vc$Geno[1])^2)
        } else {
                geno_eff = rep(0, ngeno)
        }
        if (R> 0) {
                row_eff = rchisq(nrow, df = 1/2*R^2)
        } else if(R==0) {
                row_eff = rchisq(nrow, df = 1/2*(vc$Rowf[1])^2)
        } else {
                row_eff = rep(0, nrow)
        }
        if (C>0) {
                col_eff = rchisq(ncol, df = 1/2*C^2)
        } else if(C==0) {
                col_eff = rchisq(ncol, df = 1/2*(vc$Columnf[1])^2)
        } else {
                col_eff = rep(0,ncol)
        }
df1 = dat
df1 = mutate(df1,Yield = mu +
        geno_eff[as.numeric(Geno)]+
        row_eff[Rowf]+
        col_eff[Columnf]+
        rnorm(nrow*ncol,0,ressd))
df.sim.raw=rbind(df.sim.raw, df1)
}
df.sim.raw$.n = rep(1:r, each=(nrow*ncol))
```

```
        df.sim = df.sim.raw
        return(df.sim)
}
```

C Code for lineup protocol plots

Example of R code to produce the lineup plots. The code below makes use of the `nullabor` package. Similar code was used to produce all lineup figures in this paper.

```
set.seed(111)
true.pos.5 = sample(20,1)
ggplot(lineup(true=sc1.sim.trures, samples=sc1.sim.res, pos = true.pos.5),
        aes(x=Columnf,y=res)) + facet_wrap(~.sample, ncol=5) +
        geom_boxplot()+
        xlab(NULL)+ylab(NULL)+
        theme(axis.text.y=element_blank(),
                axis.text.x = element_blank(),
                axis.ticks.x = element_blank(),
                axis.ticks.y = element_blank())
```

D The Variogram

D.1 Definition

The variogram provides a description of the spatial dependency of data and is used extensively in geoscience to represent the degree of continuity of mineralisation (Matheron, 1963). The sample variogram ordinates are defined as:

$$v_{ij} = \frac{1}{2}(e_i - e_j)^2 \quad \forall i, j = 1, \ldots, RC; i \neq j,$$

where e_i is the level-1 residual for plot number i in the trial, R and C are the total number of rows and columns respectively. The sample variogram ordinate itself is given by:

$$(l_r, l_c, \overline{v}_{rc}),$$

where l_r and l_c are the displacement between rows and columns respectively and $\overline{v}_{rc}$ is the sample mean of observed variogram ordinates with the same row and column displacements. In theory, if there is no spatial dependency then the variogram should flatten out.

D.2 Code for Variogram calculation

The following R function was written to calculate the experimental variogram for linear mixed models fitted using the `lmer` function in the `lme4` package. The function first calculates the experimental variogram ordinates for all pairwise combinations of plots in the trial. At the same time, a row displacement matrix and a column displacement matrix is created to record the appropriate row and column displacement between each pairwise combination of plots. A filter is then applied to extract the variogram ordinates for every possible combination of row and column displacements to calculate the mean experimental variogram ordinate. The function returns a table with the mean experimental variogram ordinate for every pairwise combination of row and column displacement. Note that this function is not general and only works for data recorded in the manner of the CAIGE trials. This function is only applicable to a single site trial.

```
vario.lme4 = function(dat){
        r = length(unique(dat$Row))
        c = length(unique(dat$Column))
        varmat = function(dat){
                res = matrix(nrow= length(dat$res), ncol = length(dat$res))
                i=1
                while(i<= length(dat$res)){
                        for(j in (i:length(dat$res))){
                                res[i,j]=1/2*(dat$res[i]-dat$res[j])^2
                                res[j,i] = res[i,j]
                        }
                        i= i+1
                }
        return(res)
        }
        locmat_row = matrix(nrow = r*c, ncol = r*c)
        disp_row = 1:r
```

```
        for (i in 1:(r*c)) {
                locmat_row[i,] =c(rep(0,i-1),
                rep(disp_row, times = c, length.out = r*c-i+1))
        }
        locmat_row = locmat_row + t(locmat_row)
        diag(locmat_row) = 1
        locmat_row = locmat_row - 1
        locmat_col = matrix(1:c, nrow = c, ncol = c, byrow = T)
        locmat_col = (locmat_col - t(locmat_col))
        locmat_col = kronecker(locmat_col, matrix(1, nrow = r, ncol = r))
        locmat_col[lower.tri(locmat_col, diag = F)] =
                locmat_col[lower.tri(locmat_col, diag = F)]*-1
        ordmat = varmat(dat)
        var_x = vector()
        var_y = vector()
        var_z = vector()
        for (i in unique(as.numeric(locmat_col))){
                var_y = c(var_y, rep(i, times = r))
                for (j in unique(as.numeric(locmat_row))){
                        var_x = c(var_x, j)
                        m = mean(ordmat[intersect(which(locmat_col == i),
                                which(locmat_row == j))])
                        var_z = c(var_z,m)
                }
        }
        vario_dat = cbind(var_z, var_x, var_y)
        vario_dat = data.frame(vario_dat)
        names(vario_dat) = c("z", "row_d","column_d")
        vario_dat = vario_dat[(1:ceiling(length(var_z)*0.8)),]
        return(vario_dat)
}
```

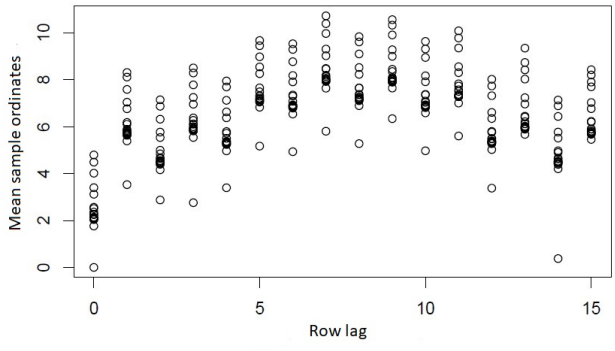
D.3 Visualisation of variogram

Although variograms are normally depicted as 3-dimensional objects, this is impractical for the purposes of producing lineups. Therefore, raster plots were used. The following R code was used to generate the lineups of the variograms.

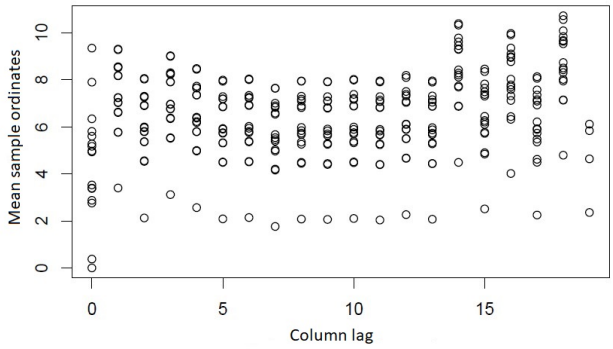
```
ggplot(lineup(true=vario.rose.f.tru,
                samples=vario.rose.f.sim, pos = true.pos.1),
                aes(x=row_d, y = column_d ,fill=z)) +
        facet_wrap(~.sample, ncol=5) +
        geom_raster()+
        scale_fill_gradientn(colours=rainbow(7))+
        xlab(NULL)+ylab(NULL)+
        theme(axis.text.y=element_blank(),
                axis.text.x = element_blank(),
                axis.ticks.x = element_blank(),
                axis.ticks.y = element_blank())
```

E Profile of inflated variogram

The following are scatter plots of the variogram when viewed "horizontally" from the side.



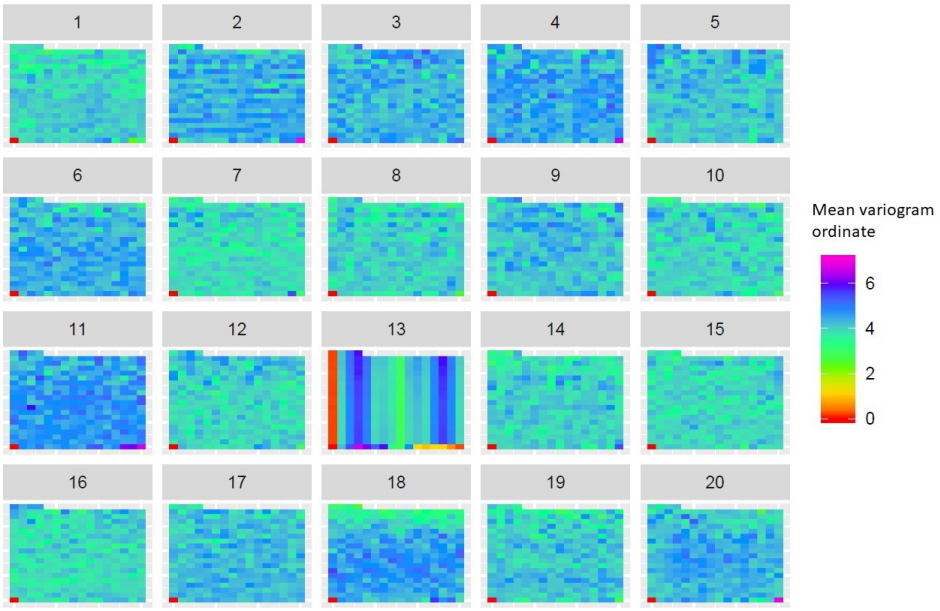
Profile scatter plot of the sample variogram. Row lags are plotted on the x axis.



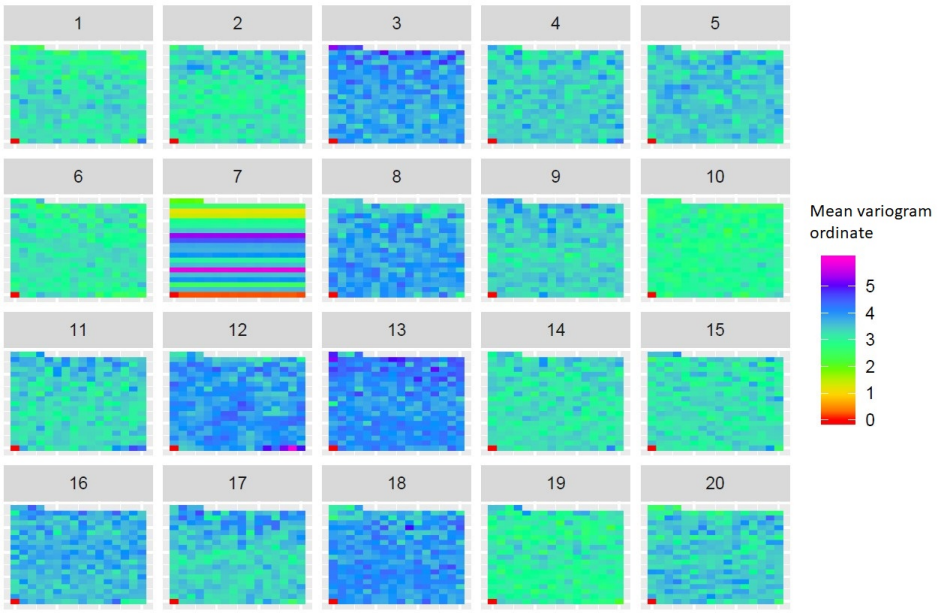
Profile scatter plot of the sample variogram. Column lags are plotted on the x axis.

F Further variograms of row and column only models

The following are lineups of sample variograms produced from models which only had an inflated Row random effect or Column random effect on top of the normal Genotype random effect. In both cases, the standard deviation of either the Row or Column effect was inflated to 2.



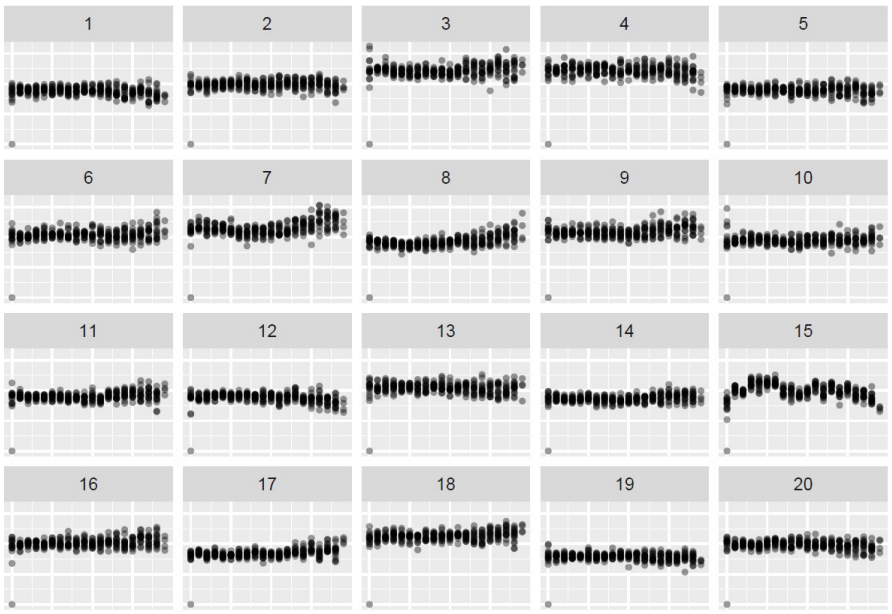
Null plots generated using $y_{ijk} = \tau_0 + \mu_i$ while the true plot was generated using $y_{ijk} = \tau_0 + \mu_i + \mu_j$.



Null plots generated using $y_{ijk} = \tau_0 + \mu_i$ while the true plot was generated using $y_{ijk} = \tau_0 + \mu_i + \mu_k$.

G Further lineups of simulated data with normal random effects

The following are lineups of the profiles of the "true" dataset and the null datasets from the Roseworthy trial. Here, the "true" dataset is simulated from the full model using the original estimated standard deviations for the Genotype, Row and Column random effects. The true plot is still located at $3 \times 7 - 6$.



The Column number is plotted on the y axis.