A Data Mining Approach for Detecting Collusion in Unproctored Online Exams

Janine Langerbein
University of Duisburg-Essen
Essen, Germany
janine.langerbein@vwl.uni-due.de

Dr. Till Massing
University of Duisburg-Essen
Essen, Germany
till.massing@vwl.uni-due.de

Jens Klenke
University of Duisburg-Essen
45117 Essen, Germany
jens.klenke@vwl.uni-due.de

Michael Striewe
University of Duisburg-Essen
Essen, Germany
michael.striewe@uni-due.de

Michael Goedicke
University of Duisburg-Essen
Essen, Germany
michael.goedicke@s3.uni-due.de

Christoph Hanck
University of Duisburg-Essen
Essen, Germany
christoph.hanck@vwl.uni-due.de

ABSTRACT
Due to the precautionary measures during the COVID-19 pandemic many universities offered unproctored take-home exams. We propose methods to detect potential collusion between students and apply our approach on event log data from take-home exams during the pandemic. We find groups of students with suspiciously similar exams. In addition, we compare our findings to a proctored comparison group. By this, we establish a rule of thumb for evaluating which cases are "outstandingly similar", i.e., suspicious cases.

Keywords
collusion detection, unproctored online exams, clustering algorithms

1. INTRODUCTION
During the COVID-19 pandemic many universities, e.g., in Germany, were forced to switch to online classes. Moreover, most final exams were held online. In pre-pandemic times, computer-based final exams have already proven their worth, but with the difference that they were proctored in the classroom. During the pandemic this was mostly unfeasible and students had to take the exam from a location of their choice.

There exists a wide range of supervisory measures for take-home exams. E.g., one could use a video conference software to monitor students. At many universities, however, this is legally prohibited due to data protection regulations. The exams are therefore conducted as open-book exams, i.e., students are allowed to use notes or textbooks. Yet, students must not cooperate with each other. Any form of cooperation or collusion is regarded as attempted cheating.

To our knowledge, it exists no universally-applicable method for proctoring take-home exams. It is therefore hardly feasible to stop students from illegally working together. However, one can attempt to identify colluding students post-exam. The attempt alone could have a deterring effect on students. Research in this area, however, is scarce. [3] present a method for comparing exam event logs to detect collusion. They use a simple distance measure for time series, i.e., the event logs of two different students, to quantify the similarity of these student's exams. Building on this, we propose an alternative distance measure, as well as the use of hierarchical clustering algorithms, to detect groups of potentially colluding students. We find that our method succeeds in finding groups of students with near identical exams. Furthermore, we present an approach to categorise student groups as "outstandingly similar", by providing a proctored comparison group.

The remainder of this paper is organised as follows: Section 2 provides a brief overview of related work. Section 3.1 describes the available data. Section 3.2 presents our method, including the calculation of the distance matrices. Section 4 discusses the empirical results. Section 5 concludes.

2. RELATED WORK
Due to the limited relevance of unproctored exams at universities before the pandemic, there exists little research about this topic. Recent work from [3] presents a method for analysing exam event logs for the detection of collusion in unproctored exams. They visually compare the event logs of pairs of students and quantify these by calculating a distance measure. They find some suspicious pairs of students with very similar event logs. Still, the authors remark that these findings might be purely coincidental. We enhance their approach by including a comparison group for drawing the line between "normal degree of similarity" and "outstandingly similar".

In other contexts, collusion in exams has been a relatively well studied topic. [9, 14] quantify the similarity of pro-

gramming exams. For this, they calculate distance measures based on student's keyboard patterns. [13] further provide an overview of relevant work in educational data mining in programming exams. Complementary, our work does not focus on keyboard patterns in programming but on the submissions of answers and achieved points in introductory statistics classes. Thus, our calculation of distance measures follows a different approach.

Furthermore, a major body of related literature focuses on a different methodology. E.g., [1, 10, 16, 18, 21] use surveys or interviews with students to collect data. Due to issues inherent to surveys and interviews, like nonresponse or incorrect responses, there is little knowledge on student collusion based on actual student behaviour. We attempt to bridge this gap by directly using student's exam data.

Generally, there exists a wide range of proctoring options during take-home exams. [6, 12] introduce and compare some of these options. A supervisor could, for example, use video conference software to observe students during the exam. This provides conditions similar to those at classroom exams and thus prevents students from colluding. Such actions have two drawbacks: First, [4] argue that proctoring take-home exams is relatively costly, so that the costs exceed the potential benefits. Second, as mentioned before, most proctoring options are strictly illegal in some countries, e.g., Germany.

On the other hand, e.g., [17] advise against unsupervised online exams. They argue that, logically, with no supvervision there is no way to prevent students from colluding during the exam. To date, there are only few studies examining the impact of unattended online examinations on the integrity of students, see, e.g., [15]. [7] use a regression model that predicts final exam scores to detect collusion in unproctored online exams. Their findings suggest that collusion took place when the final exam was not proctored. [11] compares the Grade Point Average (GPA) of students who wrote a proctored exam and students who wrote an unproctored exam. There was no evidence of a significant difference in the mean GPA between the two groups, which, however, does not establish that the students did not collaborate illegally. [5] also compare the GPA in a proctored vs. an unproctored online exam and use a regression analysis to measure student collusion. The data used in these studies were final exam scores or the GPA. None of them uses data collected during the exam.

3. METHODOLOGY
In the following we give a brief overview about the data used in our analysis. We further describe our approach to build a suitable distance metric.

3.1 Data
The data we use stems from the introductory statistics course "Descriptive Statistics" at the Faculty of Business Administration and Economics at the University Duisburg-Essen, Germany.[1] The exam of our test group was taken unproc-

Table 1: This table gives summary statistics for all students considered in our empirical analysis

Year	Minutes	Points	Subtasks	Students
Comparison Group (18/19)	70	60	19	109
Test Group (20/21)	70	60	17	151

tored during the global COVID-19 pandemic in the winter term 2020/21. The exam of our comparison group took place in the winter term 2018/19, i.e., before the pandemic.[2] It was a proctored exam located in a PC-equipped classroom at the university. Both exams use the e-assessment platform JACK [20].

Both exams consist mainly of arithmetical problems, where students are expected to submit numerical results. Moreover, there exist some tasks where students are obliged to use the programming language R [19]. The test group also had to answer a short essay task which should contain 4-5 sentences. All but the free-text tasks are evaluated automatically by JACK. The latter is manually graded by the examiner.

During the exam, the students' activities are stored in said event logs. Hence, these contain the exact time for all inputs in all tasks. For all tasks students can change and re-submit their entries. The last submission will be evaluated. For this reason, one task can list multiple events in the event log.

In addition to the event logs we also use the points achieved per task for our analysis.

Table 1 displays the basic data for both exams. Namely, these are the duration and maximum points to achieve, as well as the number of subtasks and participants per group. The wide disparity in student participants between both exams can be explainend by a change in examination regulations. During the COVID-19 pandemic, ergo in the test group, students were allowed to fail exams without any penalties. In order to prevent this from biasing our results, we removed students who attended the exam for only a few minutes and those who achieved merely a fraction of the maximum points.[3] We also removed twelve students from the test group who reported internet problems during the exam.

---

[2]The course is jointly offered by two chairs and therefore held on a rotating basis. Hence, the exam data is only comparable every two years.
[3]We also conducted the analysis without the removal of these students, with no effect but a reduced interpretability of the following clustering algorithms.

From our perspective, the setup is reasonably comparable in both groups. Although the lecture of the comparison group was held in presence and the lecture of the test group was held online, both groups shared the same content and learning goals. Both times students were given the opportunity to ask questions during the lecture. The amount of those questions remained approximately stable. Due to the sheer size of the course, with more students attending classes than participating in the exam, direct discussions were sparse even pre-pandemic.

3.2 Model

We adopt an exploratory approach for finding clusters of students with similar event patterns and points achieved during the exam. For this, we use agglomerative, i.e., bottom-up, hierarchical clustering algorithms. The results are depicted in a dendrogram. We build on previous work by [8] and [2].

In general, clustering algorithms attempt to group N objects according to some predefined dissimilarity measure. Those objects have measurements x_{ij} for $i = 1, 2, \ldots, N$, on attributes $j = 1, 2, \ldots, h$. The global pairwise dissimilarity $D(x_i, x_{i'})$, with x_i being x_{ij} over all j, between two objects i and i' is defined as

$$D(x_i, x_{i'}) = \frac{1}{h} \sum_{j=1}^{h} w_j \cdot d_j(x_{ij}, x_{i'j}); \sum_{j=1}^{h} w_j = 1, \quad (3.1)$$

with $d_j(x_{ij}, x_{i'j})$ the pairwise attribute dissimilarity between values of the jth attribute and w_j the weight of the attribute. The clustering algorithm therefore takes a distance matrix as input.

In our case, the students are the objects to be clustered, with $N = 151$ students. As attributes we use the dissimilarities in the student's event patterns and the dissimilarities in their points achieved. Both need to be calculated differently, but over all subtasks. Hence, we split $d_j(x_{ij}, x_{i'j})$ into two parts.

We call the attribute dissimilarity for the points achieved $d_j^P(s_{ij}, s_{i'j})$ with w_j^P its corresponding weight. s_{ij} denotes the points achieved by student i in the jth subtask. Since there are 17 subtasks we obtain a total of $h = 34$ attributes. To receive a dissimilarity measure we calculate the absolute differences

$$d_j^P(s_{ij}, s_{i'j}) = |s_{ij} - s_{i'j}|. \quad (3.2)$$

Next, $d_j^L(v_{ij}, v_{i'j})$ describes the dissimilarities in the event patterns per subtask. To calculate these we divide the examination time into $m = 1, \ldots, K$ intervals of one minute. Since both exams each took 70 minutes, we obtain $K = 70$ intervals. We count each student's answer per interval. The count is denoted with v_{ijm}.[4] To obtain a pairwise attribute dissimilarity measure for all subtasks, we calculate the Manhattan metric over all counted quantities

$$d_j^L(v_{ij}, v_{i'j}) = \sum_{m=1}^{K=70} |v_{ijm} - v_{i'jm}|. \quad (3.3)$$

---

[4]We consider this an enhancement of the distance measure used in [3], as it enables us to analyse exams with more than one answer per task.

The corresponding weight is denoted by w_j^L. To ensure better transparency, we provide a detailed explanation of each variable in Appendix A.

Finally, we modify (3.1) so that

$$D(s_i, s_{i'}, v_i, v_{i'}) = \frac{1}{h} \sum_{j=1}^{h} \Big(w_j^P \cdot d_j^P(s_{ij}, s_{i'j})$$
$$+ w_j^L \cdot d_j^L(v_{ij}, v_{i'j}) \Big)$$
$$\text{with} \sum_{j=1}^{h} w_j^P + w_j^L = 1. \quad (3.4)$$

The attribute weights w_j control the influence of each attribute on the global object dissimilarity. If all 34 attributes are to be weighted equally, each attribute would be assigned a weight of $\frac{1}{34}$. Here, however, we weight the attributes with regard to our research question. Specifically, we observe that students submit entries more often in the case of R-tasks, viz. subtasks $6a, 6b$ and $6c$. One possible interpretation of this is that students submit their code more often to check its executability. Furthermore, task 7 demands the answer to be a short text which was corrected manually. This could lead to insufficient comparability between students due to accidental arbitrariness during correction. Based on these aspects, it appears reasonable to reduce the weight of said subtasks.

We further reduce the influence of the points achieved during the exam by decreasing their weight. This follows from the fact that prior to the exam we must define all (partially) correct answers in JACK. In doing so, it is not feasible to anticipate all types of mistakes resulting from, e.g., calculation errors made by students.[5] Students might receive no points due to careless mistakes, while still having employed a correct solution strategy. In our view, this might impede the detection of colluding students, e.g., if there exist large differences in points as one student makes more frequent careless mistakes due to the random numbers in the tasks. On this account, we assign smaller weight to the points achieved. For greater clarity, an overview with all exact final weights for all attributes can be found in Appendix B.[6]

The influence of each attribute on object dissimilarity further depends on its scale. We therefore normalise each attribute.

From these pairwise object dissimilarities, we create the distance matrices. We then apply agglomerative hierarchical clustering. This builds a hierarchy by merging the most similar pairs of students, viz. those with the lowest object dissimilarity $D(x_i, x_{i'})$, into a cluster. This is repeated $N-1$ times, until all students are merged into one single cluster. The merging process is implemented with different linkage methods. These differ in their definition of the shortest distance between clusters. Here, we use single, average and complete

---

[5]The tasks are randomised, i.e., there exist variations so that sharing exact results is not expedient for the students.
[6]A robustness check regarding the object weights can be found in Appendix D. In this analysis, the weights of the objects are equal. The results are basically identical.
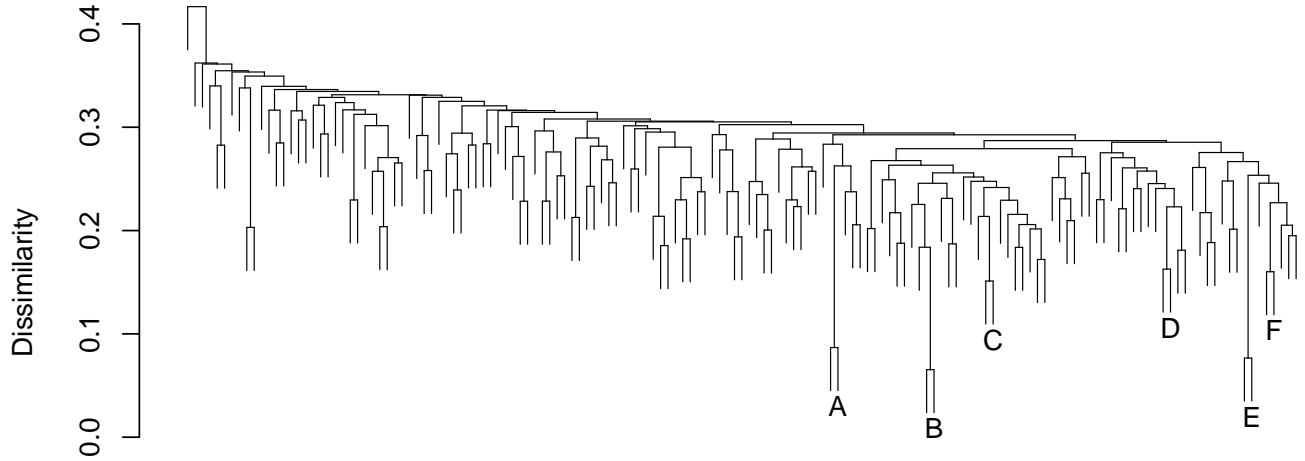
Figure 1: Dendrogram produced by average linkage clustering for the test group 2020/21. The dissimilarity of each group's node is displayed on the y-axis. Its value corresponds to the dissimilarity of the group's left and right member. A - F denotes the six lowest dissimilarity clusters. Of these especially, clusters A, B, and E are notable.

linkage. The former merges clusters with the closest minimum distance, the latter uses the closest maximum distance. The average linkage method (here: unweighted pair group method with arithmetic mean) defines the distance between any two clusters as the average distance among all pairs of objects in said clusters.

4. EMPIRICAL RESULTS

We present the results of the hierarchical clustering algorithms in a dendrogram. This provides a complete visual description of the results from the agglomerative hierarchical clustering algorithm. A dendrogram resembles a tree structure where each object is represented by one leaf. In a bottom-up approach, the objects are merged into groups one by one according to their dissimilarity. Hence, each level of the tree corresponds to one step of the clustering algorithm. The junction of a group is called a node.

There exist various hierarchical clustering algorithms. Each has a different definition of the distance between groups of observations as a function of the pairwise distances. We calculate the cophenetic correlation coefficient to assess how faithfully each algorithm represents the original structure in the data. Appendix C, Table 5 gives an overview of the cophenetic correlation coefficient for the different linkage methods for the test and comparison groups. Based on this, we deem average linkage clustering the most suitable algorithm. Figure 1 shows the corresponding dendrogram. The dissimilarity of each group's node is plotted on the vertical axis. Its value corresponds to the dissimilarity of the group's left and right member.

It is important to note that a dendrogram only gives an

indication of clusters which best fit the data. It is up to the analyst to decide which are to be examined in further detail.

The dendrogram has a slightly elongated form. Still, compact clusters were produced at medium dissimilarities. This general shape is typical for the underlying algorithm, as average linkage clustering combines the long form of single linkage clustering with the smaller, tighter clusters of complete linkage clustering. Additionally, we observe three notable clusters (A, B and E) which form at a significantly lower height. Each of these three clusters consists of two students. Prima facie, this indicates the absence of collusion in larger groups.

As explained above, the hierarchical algorithm does not cluster the data itself, but imposes a structure according to the students' dissimilarities. There exist various formal methods to decide on an optimal number of clusters given this established hierarchy. Since our primary interest lies in the detection of clusters at low dissimilarities, instead of the general structure of the data, we exemplary investigate the six lowest clusters (A - F) in Figure 1.

Figure 2 shows the exact course of events for the described selection of clusters. Each scatterplot plots all answers of the students in the cluster against their time of submission. We expect students' chronology to be more similar if their cluster's node is a lower height, i.e., lower dissimilarity. We also add the points achieved on top in a barchart.

As expected, all scatterplots show some kind of similarity. In particular, clusters A, B and E bear a striking resemblance.
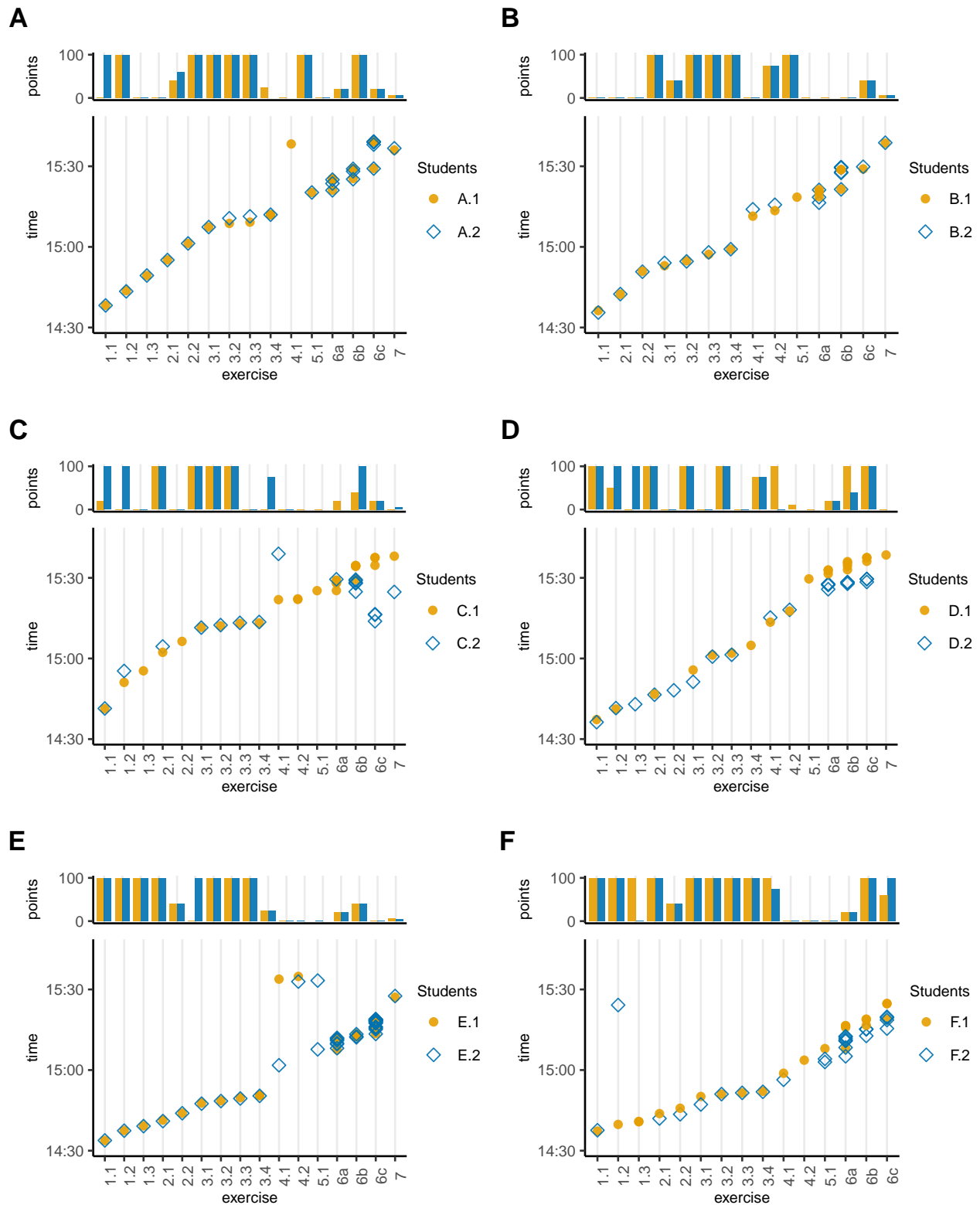
Figure 2: Comparison of the event logs and achieved points for each of the test group's (2020/21) six lowest dissimilarity clusters (A - F). The letters of the sub-figures correspond to the marked clusters in Figure 1. The number behind the letters refers to the node's left and right arm, respectively. At the bottom of each sub-figure, the sub-tasks are plotted against the clock time. Above the scatterplot, a bar chart is added to compare the points per subtask.

Their respective barchart further reveals these students to almost always achieve an equal number of points per subtask. In direct comparison, the plots of the remaining clusters C, D and F look less similar. This shows that clusters with lower node heights indeed contain more similar exams.

To assess whether these similarities are the result of collusion or coincidence, we repeat our approach on the comparison group, the final exam of the same course from two years ago. The plots were created analogously to the plots of the test group.

For the comparison group we also focus on average linkage clustering. The associated dendrogram, however, has a slightly different shape (see Figure 3). The most prominent difference, in contrast to the test group, is the absence of any visually outstanding clusters. We rather observe most nodes at a comparatively similar height.

Figure 4 shows the scatter- and barplots of the six lowest clusters in the comparison group. We find there to be significantly less similarities not only in the chronological aspect, but also in the points achieved.

The results from the comparison group support our assumption that widespread collusion over the entire exam is hardly achievable in presence. Moreover, the clear visual differences between comparison- and test group indicate that our findings in the latter might not be coincidental.

In the test group it is relatively simple to identify at least three suspicious clusters. In the less obvious cases it can be challenging to decide which clusters to investigate further, as there exists no clear rule on where to draw the line between suspicious and unsuspicious cases. We address this issue by defining a "normal degree" of similarity, which can be used as a bound to classify whether a pair of students is deemed suspicious or not. For our data of the test group, there is no indication of the existence of suspicious clusters of more than two students. Hence, we refocus on the global pairwise dissimilarity $D(x_i, x_{i'})$.

To assess the "normal degree" of similarity, we first standardise both distributions to improve their comparability. For the comparison group, we define a lower bound below which we categorise observations as extreme outliers. This bound is then used on the lower tail of the distribution of the test group. We want to identify cases in the unproctored test group which are rather extreme compared to the proctored comparison group. For this we calculate the lower bound as $Q_1 - 3 * IQR$ with Q_1 being the first quartile of the data and IQR being the interquartile range.

The boxplots in Figure 5 show the distributions of the global pairwise dissimilarity $D(x_i, x_{i'})$ of all students in the comparison- and test group. A boxplot provides a graphic overview of location and dispersion of a distribution. The eponymous box marks the upper and lower quartile of the data. Outliers are displayed by individual points.

On the left hand side of Figure 5 we observe that both distributions posess a similar shape, but a different median. The median value of the test group is significantly lower, indi-cating a lower average global pairwise dissimilarity in this group. Furthermore, we discover a high number of outliers in both groups, albeit at different positions in their respective distribution. In the test group, more outliers lie on the lower side of the box, with a greater distance to the main part of the distribution. We also find three observations with extremely small values on the lower tail of the test group's distribution. Unsurprisingly, these belong to the clusters A, B and E.

The right side of Figure 5 shows the normalised distributions. It is apparent that the normalised distribution of the test group still contains more outliers.

We apply the above mentioned lower bound on the test group's distribution to identify groups of students which are "outstandingly similar". Here, the before mentioned three cases (clusters A, B and E) fall below the lower bound for extreme outliers. While it is no surprise that these clusters were detected, our approach still aids us in deciding on when to stop inspecting further groups of students, as their level of similarity might as well occur in the comparison group.

To summarise, our approach offers a rule of thumb for narrowing down the number of suspicious cases. This is particularly useful if the visual distinction of cases is not clear-cut.

5. CONCLUSIONS AND DISCUSSION

During the COVID-19 pandemic many exams at universities had to be converted into unproctored take-home exams. We propose a method for detecting potentially colluding students in said exams. For this, we calculated a distance measure based on the students' event logs and their points achieved from the exam. Compared to former approaches adressing this topic, we use a distance measure which also applies if there exist multiple events per task. Subsequently, we use hierarchical clustering algorithms to detect clusters of potentially colluding students. The results show that our method is able to detect at least three clusters with near identical exams.

To decide which degree of similarity might be more than a coincidence we compare the normalised distributions of the distance measures of our test and comparison group. We find pairs of students in the test group with values below the minimum of the comparison group. Thus, our approach provides a basis for deciding on which clusters are to be examined further. A limitation of this approach is that we do not know the ground truth in our groups and only be able to back up our reasoning on a comparison.

In summary, we have been successful in providing an opportunity to detect colluding students after the exam. We cannot say if this is sufficient evidence to initialise legal consequences. Nevertheless, we are confident that the higher chance of getting caught has a deterring effect on students. This would be an interesting direction for further research. Moreover, one could collect complementary evidence. By doing so, we found at least two of our suspicious students confirmed.

6. ACKNOWLEDGMENTS
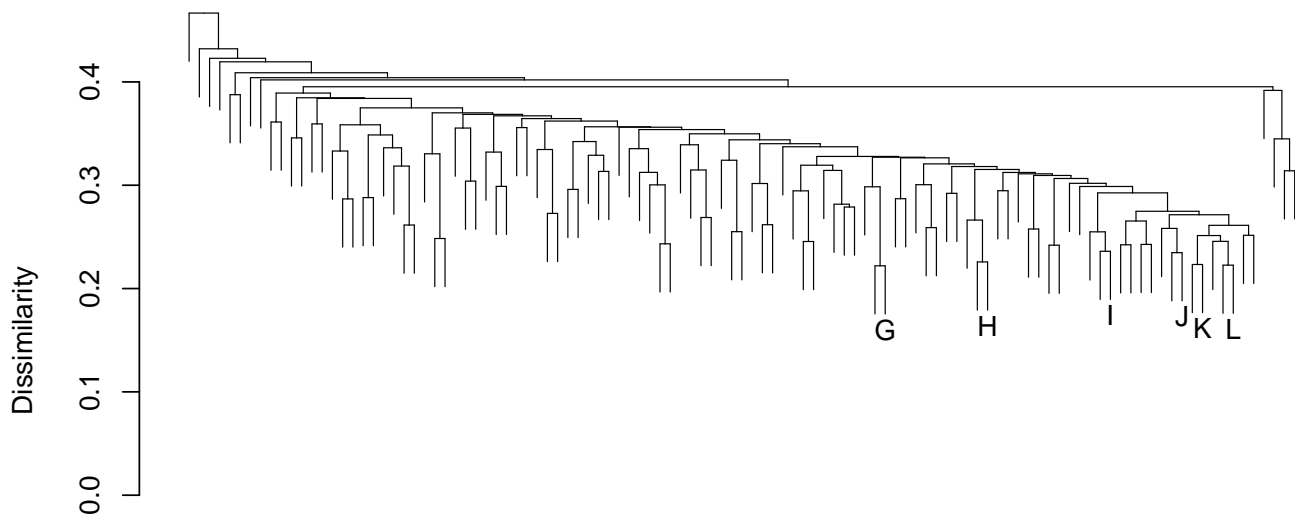
Figure 3: Dendrogram produced by average linkage clustering for the comparison group 2018/19. The dissimilarity of each group's node is displayed on the y-axis. Its value corresponds to the dissimilarity of the group's left and right member. G - L denotes the six lowest dissimilarity clusters.

7. ADDITIONAL AUTHORS

Additional authors: Natalie Reckmann (University of Duisburg-Essen, email: `natalie.reckmann@vwl.uni-due.de`).

8. REFERENCES

[1] W. Bowers. Student dishonesty and its control in college, 1964.

[2] R. Chatterjee. Similarity/clustering methods for temporal event data, 2015 (accessed 2021).

[3] C. Cleophas, C. Hoennige, F. Meisel, and P. Meyer. Who's cheating? mining patterns of collusion from text and events in online exams. *Mining Patterns of Collusion from Text and Events in Online Exams (April 12, 2021)*, 2021.

[4] G. Cluskey Jr, C. R. Ehlen, and M. H. Raiborn. Thwarting online exam cheating without proctor supervision. *Journal of Academic and Business Ethics*, 4(1):1–7, 2011.

[5] A. Fask, F. Englander, and Z. Wang. On the integrity of online testing for introductory statistics courses: A latent variable approach. *Practical Assessment, Research, and Evaluation*, 20(1):10, 2015.

[6] D. Goldberg. Programming in a pandemic: Attaining academic integrity in online coding courses. *Communications of the Association for Information Systems*, 48(1):6, 2021.

[7] O. R. Harmon and J. Lambrinos. Are online exams an invitation to cheat? *The Journal of Economic Education*, 39(2):116–125, 2008.

[8] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction.* Springer, 2 edition, 2009.

[9] A. Hellas, J. Leinonen, and P. Ihantola. Plagiarism in take-home exams: Help-seeking, collaboration, and systematic cheating. page 238–243, 2017.

[10] A. Hemming. Online tests and exams: lower standards or improved learning? *The Law Teacher*, 44(3):283–308, 2010.

[11] K. K. Hollister and M. L. Berenson. Proctored versus unproctored online exams: Studying the impact of exam environment on student performance. *Decision Sciences Journal of Innovative Education*, 7(1):271–294, 2009.

[12] M. J. Hussein, J. Yusuf, A. S. Deb, L. Fong, and S. Naidu. An evaluation of online proctoring tools. *Open Praxis*, 12(4):509–525, 2020.

[13] P. Ihantola, A. Vihavainen, A. Ahadi, M. Butler, J. Börstler, S. H. Edwards, E. Isohanni, A. Korhonen, A. Petersen, K. Rivers, et al. Educational data mining and learning analytics in programming: Literature review and case studies. *Proceedings of the 2015 ITiCSE on Working Group Reports*, pages 41–63, 2015.

[14] J. Leinonen, K. Longi, A. Klami, A. Ahadi, and A. Vihavainen. Typing patterns and authentication in practical programming exams. pages 160–165, 2016.

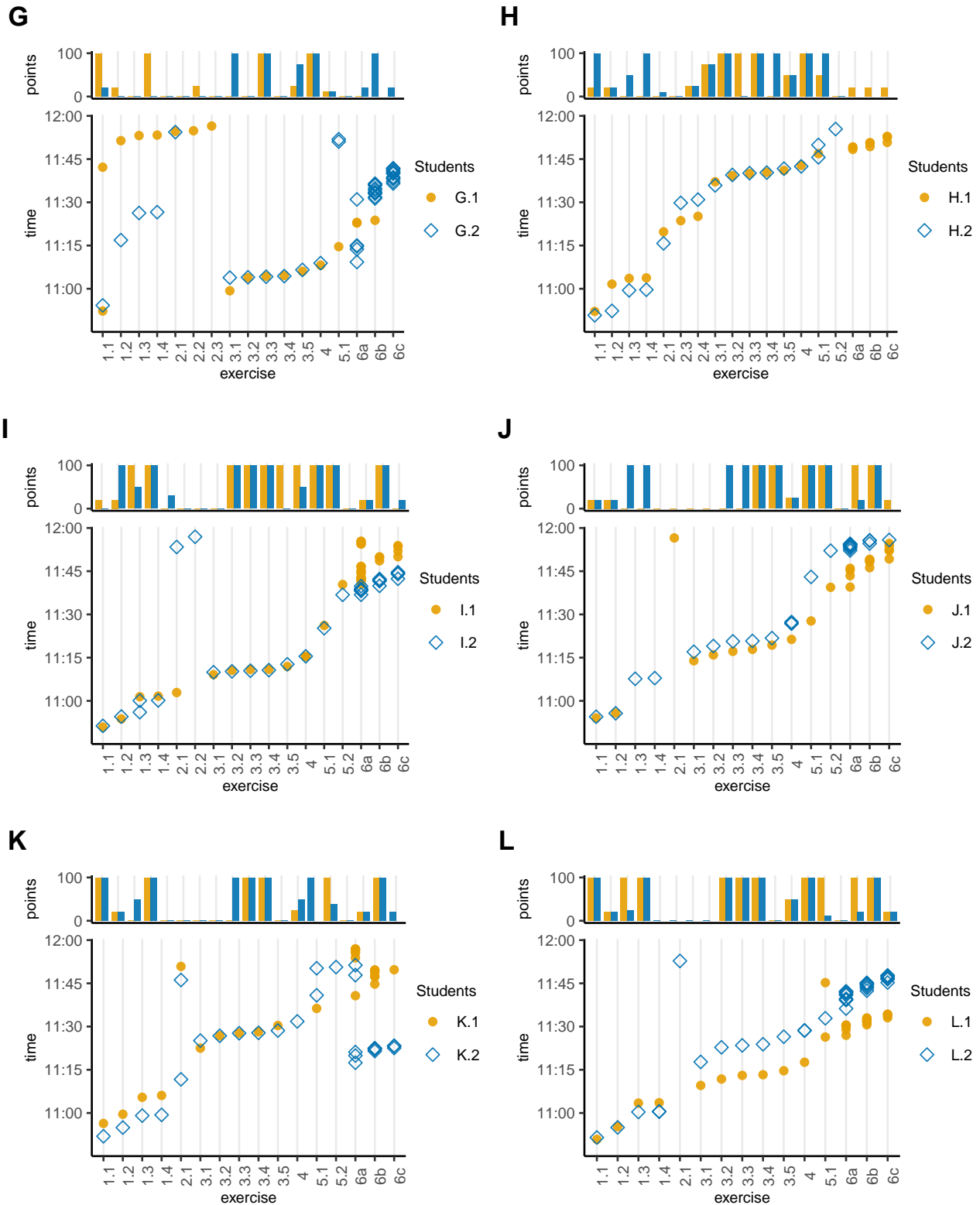[15] S. Manoharan and X. Ye. On upholding academic integrity in online examinations. In *2020 IEEE*

Figure 4: Comparison of the event logs and achieved points for each of the comparison group's (2018/19) six lowest dissimilarity clusters (G - L). The letters of the sub-figures correspond to the marked clusters in Figure 3. The number behind the letters refers to the node's left and right arm, respectively. At the bottom of each sub-figure, the sub-tasks are plotted against the clock time. Above the scatterplot, a bar chart is added to compare the points per subtask.
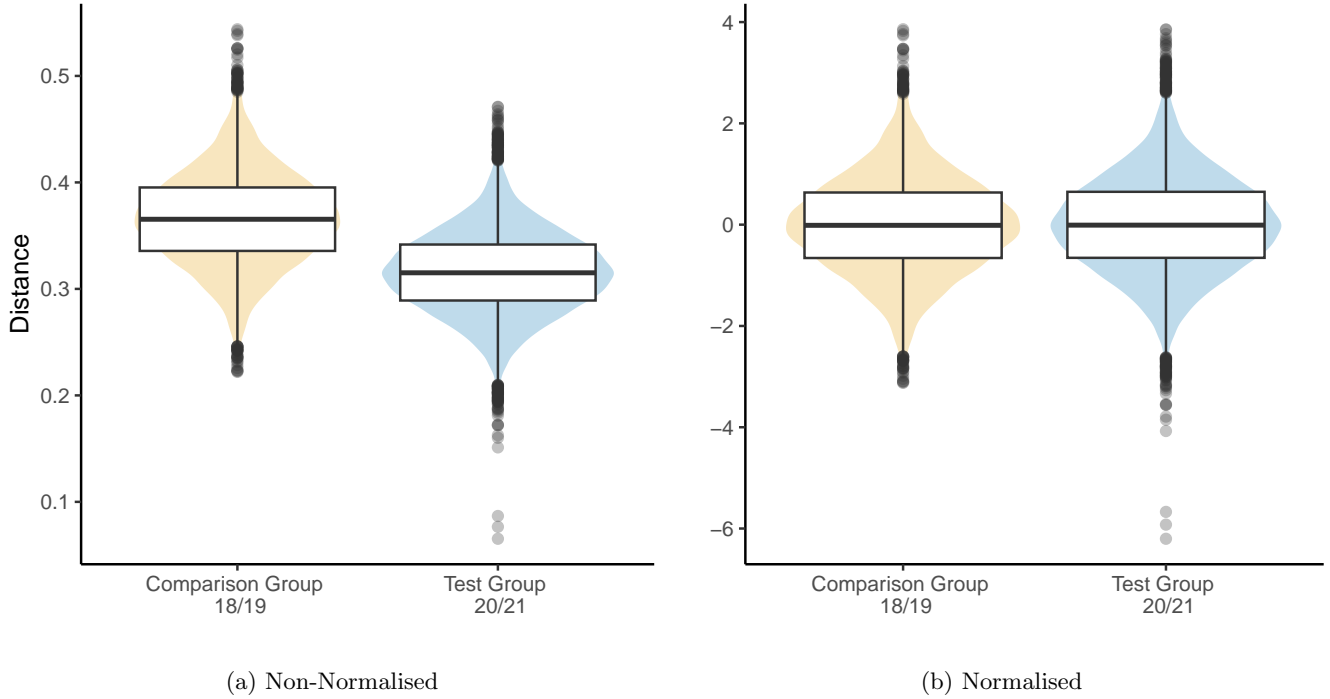
(a) Non-Normalised   (b) Normalised

Figure 5: Boxplot of the pairwise distance measures.

Conference on e-Learning, e-Management and e-Services (IC3e), pages 33–37. IEEE, 2020.

[16] D. L. McCabe, L. K. Treviño, and K. D. Butterfield. Cheating in academic institutions: A decade of research. *Ethics &Behavior*, 11(3):219–232, 2001.

[17] J. Miltenburg. Online teaching in a large, required, undergraduate management science course. *INFORMS Transactions on Education*, 19(2):89–104, 2019.

[18] M. R. Olt. Ethics and distance education: Strategies for minimizing academic dishonesty in online assessment. *Online journal of distance learning administration*, 5(3):1–7, 2002.

[19] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2021.

[20] N. Schwinning, M. Striewe, T. Massing, C. Hanck, and M. Goedicke. Towards digitalisation of summative and formative assessments in academic teaching of statistics. In *Proceedings of the Fifth International Conference on Learning and Teaching in Computing and Engineering*, 2017.

[21] P. C. Shon. How college students cheat on in-class examinations: Creativity, strain, and techniques of innovation, 2006.

APPENDIX
A. VARIABLE DESCRIPTION

Table 2: Variable Description.

Variable	Description
s_{ij}	Points achieved in the jth subtask by the ith student.
v_{ij}	Even patterns for the jth subtask by the ith student.
x_{ij}	Measurement for the ith object and jth attribute. Here, x_{ij} only functions as a variable for explaining the general clustering approach.

B. ATTRIBUTE WEIGHTS

Below we list the attribute weights used for building the global object dissimilarity.

Table 3: The weights for all attributes in the test group (2020/21), rounded to three decimal places.

	weights	
(sub-)tasks	event patterns	points
1.1 - 5.2	0.052	0.013
6a - 6c	0.026	0.013
7	0.026	0.013

Table 4: The weights for all attributes in the comparison group (2018/19), rounded to three decimal places.

	weights	
(sub-)tasks	event patterns	points
1.1 - 5.2	0.045	0.011
6a - 6c	0.022	0.011

C. DENDROGRAM AND COPHENETIC CORRELATION COEFFICIENT

Table 5: The cophenetic correlation coefficient for all three linkage methods for the comparison (2018/19) and test (2020/21) group.

	C	
Linkage method	2020/21	2019/18
single	0.6610	0.6659
complete	0.4424	0.5051
average	0.6964	0.7595

We must consider that clustering algorithms enforce a hierarchical structure on the data. This structure, however, does not have to exist. There exist a good amount of methods to assess how faithfully each algorithm represents the original distances in the data. Here, we use the cophenetic correlation coefficent (C). This is defined as the linear correlation between the pairwise dissimilarity $D(x_i, x_{i'})$ from the original distance matrix and the corresponding *cophenetic* dissimilarity from the dendrogram $t(x_i, x_{i'})$, i.e., the height of the node of the cluster. Let $\overline{D}$ be the mean of $D(x_i, x_{i'})$ and $\bar{t}$ be the mean of $t(x_i, x_{i'})$. Then, C can be written as

$$C = \frac{\sum_{i<i'} \left(\left(D(x_i, x_{i'}) - \overline{D} \right) \left(t(x_i, x_{i'}) - \bar{t} \right) \right)}{\sqrt{\sum_{i<i'} \left(D(x_i, x_{i'}) - \overline{D} \right)^2 \sum_{i<i'} \left(D(x_i, x_{i'}) - \bar{t} \right)^2}}. \tag{B.1}$$

Table 5 shows C for all three linkage methods. The clustering with the complete linkage method seems to be the most unsuitable. The results of the single and average linkage clustering seem to be an adequate representation, with the latter a slightly better fit. We therefore proceed with the average linkage method in all further steps.

D. ROBUSTNESS CHECK

We repeat our analysis on the same data, but we assign the same weight to each attribute while calculating the global object dissimilarity matrix. In simple terms, we replace the weighted arithmetic mean in equation 3.1 in chapter 3.2 with the ordinary arithmetic mean.
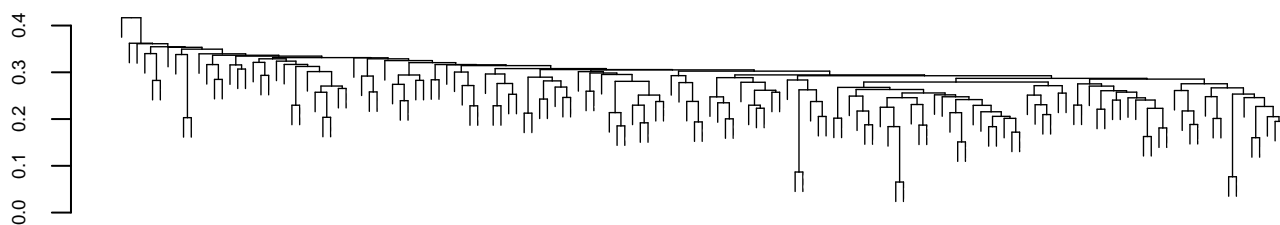
Figure 7 shows the dendrogram for the test data with equal weights. The algorithm still manages to identify the three suspicious clusters. Furthermore, these clusters are still merged at a comparatively low height.
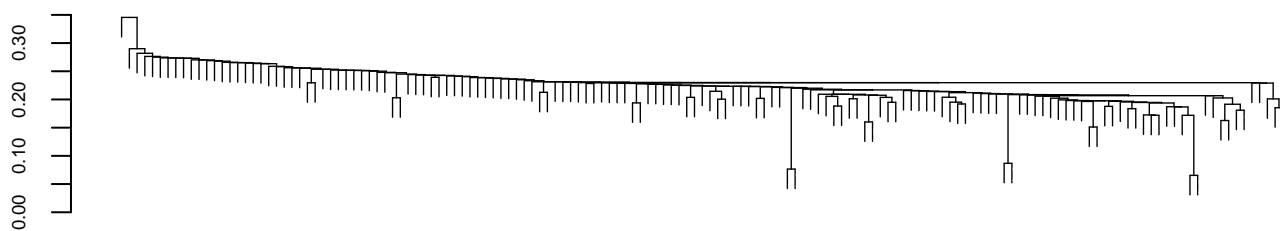
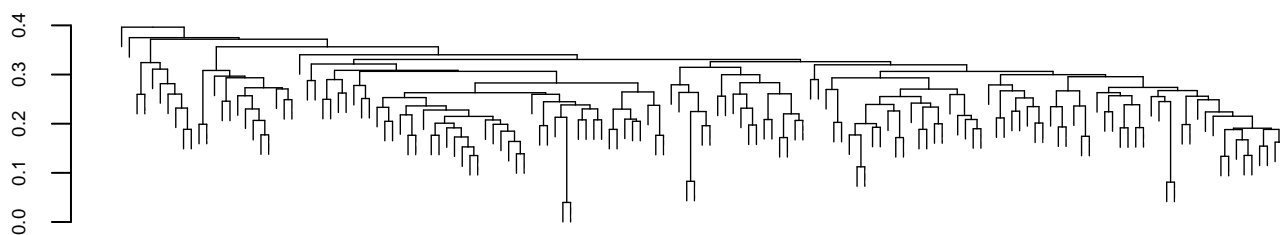(a) Single Linkage Clustering
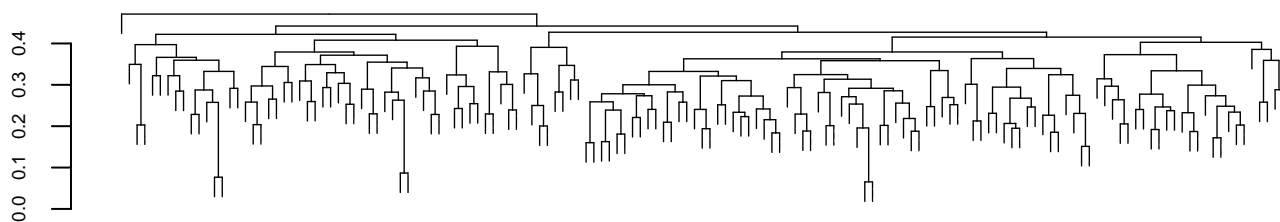
(b) Complete Linkage Clustering

(c) Average Linkage Clustering

Figure 6: Comparison of the dendrograms for all three linkage methods in the test group 2020/21.

Figure 7: Dendrogram of average linkage clustering with the test data and equals weights.