Leading Learning in Communities

Hal Kolb

May 29, 2022

1. Introduction

To understand how demographics, pupil identities and the strengths of the local community effect learning in my classroom I designed a survey to identify where attitudes and opinions on computer science were varying between groups of pupils. I then used pupil data gathered from the school's SIMS system to cross reference pupil feedback with the data about them available to the school. Overall, this analysis was successful in identifying gender as the biggest indicator of interest in continuing computer science (with 52% of boys expressing interest in continuing study compared to only 24% of girls). The further survey responses allowed me to identify that while girls believed computer science would lead to good, diverse jobs they also felt that computer science was "not for people like them", that they "could not get jobs as programmers if they wanted" and ultimately lacked interest in computer science careers. By analysing recent assessment results I concluded that gender was not effecting attainment and therefor targeted the intervention around improving the girls' perception and opinion of computer science in my classroom.

A review of the literature identified exposure to positive role models and group work as two evidence-supported strategies for improving the interest of women and girls in computer science, with the greatest impact coming from early intervention Happe et al. (2020) (Accenture, 2016) (Nash, 2017). These interventions were then implemented in my lessons in the form of peer programming in Python with my Year 7 classes, and a classroom display to expose students to positive and relatable role models. The same classes were then asked to fill in the survey again and showed a measurable improvement towards more positive opinions. These changes affected all genders but had a greater impact on the girls, resulting in a reduction of the gender-based discrepancy in interest and feelings of belonging.

2. School Context

The school is a mixed-gender academy located in the London borough of Harrow, catering to pupils from 11 to 18. The latest published data (from the 2018/19 academic year) gives an enrolment of

1197 pupils across these year groups. The school has a 0,74 ratio of girls to boys, compared to a national average of 0,99. The national average percentage of pupils eligible for free school meals (FSM) is 27,7% with this school having 38,6% eligibility. The larger number of boys and FSM eligible pupils are one of the unique aspects of the school's demographics. This will be discussed later. Another noteworthy statistic is that 60,6% speak English as an additional language (EAL). This is significantly higher than the national average of 16,9%. The ways that these particular demographic differences manifest in my classroom is discussed at greater length in later sections.

(Hatch End High School, 2022)

3. Research Rationale & Intervention Methodology

Computer science careers and the tech sector currently suffer from a considerable lack of diversity and a very narrow stereotype of the kind of person who fills this role. As such I set out to identify how these patterns and attitudes manifest in my classroom and what I can do to ensure all pupils feel they can succeed in the subject. I codified these aims in the form of the following four research questions:

- Where is student's interest and engagement in computer science dropping?
- Are there patterns in the students who are lacking engagement?
- What research-supported intervention can be designed to improve engagement among these groups?
- How effective are these strategies when implemented in my classroom?

3.1. Where is interest dropping?

3.1.1. Gathering Data

To investigate attitudes towards computer science further education and careers I created a survey for students in the form of a Likert. The Likert consisted of eight statements with 5 optional responses: Strongly disagree, Disagree, Neutral, Agree and Strongly agree. These eight statements were: "Learning Computer Science will be useful to me in the future.", "Studying Computer Science leads to good jobs.", "I am interested in a career involving Computer Science.", "Computer Science is easy.", "If I wanted to I could get a job as a programmer.", "There are lots of different types of jobs you can get by studying Computer Science.", "Computer Science is interesting." and "Computer Science is for people like me.". I also included the question: "Do you plan on continuing to study Computer Science beyond your current level (e.g. as a GCSE, A-Level or Degree)?" with a simple yes-no response.

Each of these statements were designed to target a particular aspect of pupil opinion e.g.

- Broad belief in the value of computer science: “Learning Computer Science will be useful to me in the future.”
- Belief in the employment value of computer science: “Studying Computer Science leads good jobs.”
- Career interest: “I am interested in a career involving Computer Science.”
- Subject knowledge confidence: “Computer Science is easy.”
- Confidence in ability in further life: “If I wanted to I could get a job as a programmer.”
- Beliefs on career prospects and diversity: “There are lots of different types of jobs you can get by studying Computer Science.”
- Subject interest: “Computer Science is interesting.”
- Feelings of belonging in the subject: “Computer Science is for people like me.”

Analysing the responses to each question helped me target the highest leverage action to increase engagement in the subject.

To gather responses I created a small website where the form was hosted (see Appendix 1-4). I used a small PHP file to send each response to a mySQL database where the results were collected for analysis. This survey is viewable at the address: http://hkolb.co.uk. To cancel bias in the framing of the questions and to improve the validity of results I randomly assigned either a positive or negative framing to each pupil on a per question basis. It is likely that any bias in the framing of the question would effect different groups equally and therefor bias would not be important when comparing responses between groups. However, to improve the validity of the absolute value of the responses (when not comparing between groups) and to mitigate the possible effects of different groups responding to bias differently, two versions of each statement were included. The initial version of each question was framed in the positive sense. I therefore contrasted these with versions with a negative framing: “Learning Computer Science will not be useful to me in the future.”, “Computer Science does not lead to good jobs.”, “I do not want a career involving Computer Science”, “Computer Science is hard.”, “I could not get a job as a programmer if I wanted to.”, “There are only a small number of types of jobs you can get if you study Computer Science.”, “Computer Science is boring.” and “Computer Science is not for people like me.”. When the website loads, a JavaScript file randomly selected one of the two versions for each question. These selections were saved as cookies to avoid refreshing and getting different questions.

When analysing the results each response was given a weight from -2 to 2, with strongly disagree having a value -2, disagree a value of -1, neutral a value 0, agree 1 and strongly agree a value of 2. For the negative framing of the questions these values had their signs flipped and were combined with the responses to the positive framing to average out possible bias. Ultimately, all the patterns that were obseved in the combined results were also present in the individual results of each framing.

The data were then matched with pupil data form the school records, using pupil's first names to search for the responses that matched that student. The school data included information such as pupil's target grades, special educational needs (SEN) status, FSM status, EAL status, gender, ethnicity and prior assessment scores.

One bug that was unfortunately noticed too late to fix was the fact that name collisions can occur when a name is contained within another name. For example 'Aria' could be found as a match for 'Daria'. This can also occur for identical names.

In total, 221 responses were collected from students from a range of year groups. These data were combined using a Python script (see Appendix 5) to create the visualisations shown in the following sections.

3.2. Patterns in the Data

Each question's responses are represented using double bar graphs to compare two groups (below). Each response's weight (from -2 to 2) is averaged together to quantify the average opinion of that group. The two groups averages are plotted as vertical lines on the bar graphs. The values of these averages and the absolute difference between them are also shown below the graph. The full set of graphs are available in the appendix (section 7), however, only relevant graphs are shown here.

Graphs were created for each question comparing groups from the following categories: FSM status, EAL status, gender and ethnicity. As FSM and EAL status are binaries they were easy to represent using a dual bar graph, however, other attributes presented some challenges. As it was impractical to create readable visualisations with more than two groups, some reductionism was unfortunately necessary. Only binary male or female gender identities were represented on the graph, with data from students with other gender identities not included. Ethnicity also involved cutting corners; white backgrounds made up the plurality of the data and were therefore compared to the sum of the data from all other backgrounds. These data are included for the sake of completeness however, due to the extent to which nuance in this data has been stripped, the validity of broad interpretation is subject to question.

3.2.1. English as an Additional Language Status

As discussed, computer science and tech careers are dominated by a homogenous group and a diversity of backgrounds can be lacking. One possible barrier can be EAL status. EAL is a very valuable asset for students, however, it can also be a challenge for some in accessing classroom learning (Johnson, 2008) (El-Lahib et al., 2011). Despite this possibility, looking at the data from the survey responses, the EAL students I teach are: a) more likely to be interested in computer science careers, b) more likely to agree with the statement "computer science is easy", c) more confident in agreeing they could get a job as a programmer if they wanted, and d) more likely to respond that computer science is for people like them. As noted earlier, this school has a majority EAL population which is well above the national average. Taking these data points together it is encouraging to see that there is evidence that EAL students feel confident in studying computer science and have a sense of belonging in the subject and field. The EAL students I teach have strong confidence in the subject and future employment prospects and as such no intervention was planned for this group. The data may also indicate that pupils who speak English as a first language may be being left behind, however, the discrepancy in these data is minimal compared to other differences shown below, and as such this was also not a focus for my intervention.

Dual bar graph of responses to the statement "Computer science is for people like me" comparing EAL and non EAL students.

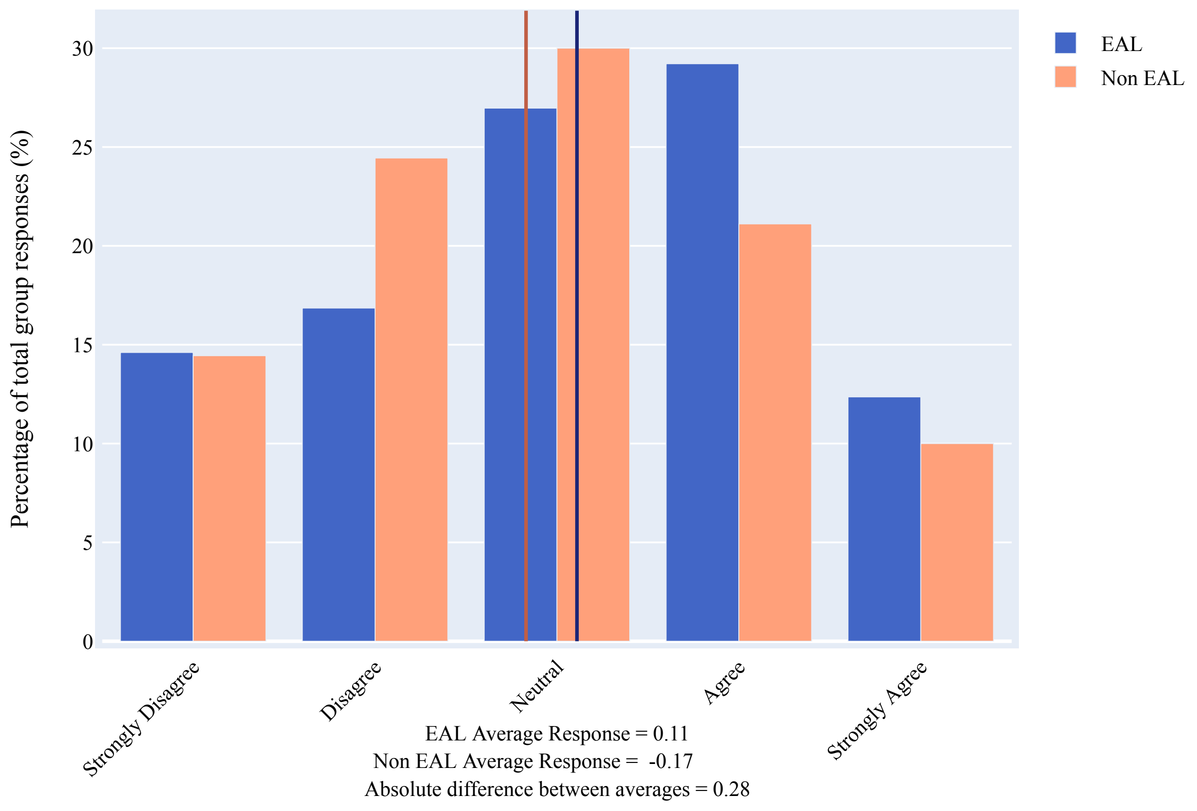


Figure 1: Dual bar graph showing the percentages of each group by responses.

Dual bar graph of responses to the statement “I am interested in a career involving computer science” comparing EAL and non EAL students.

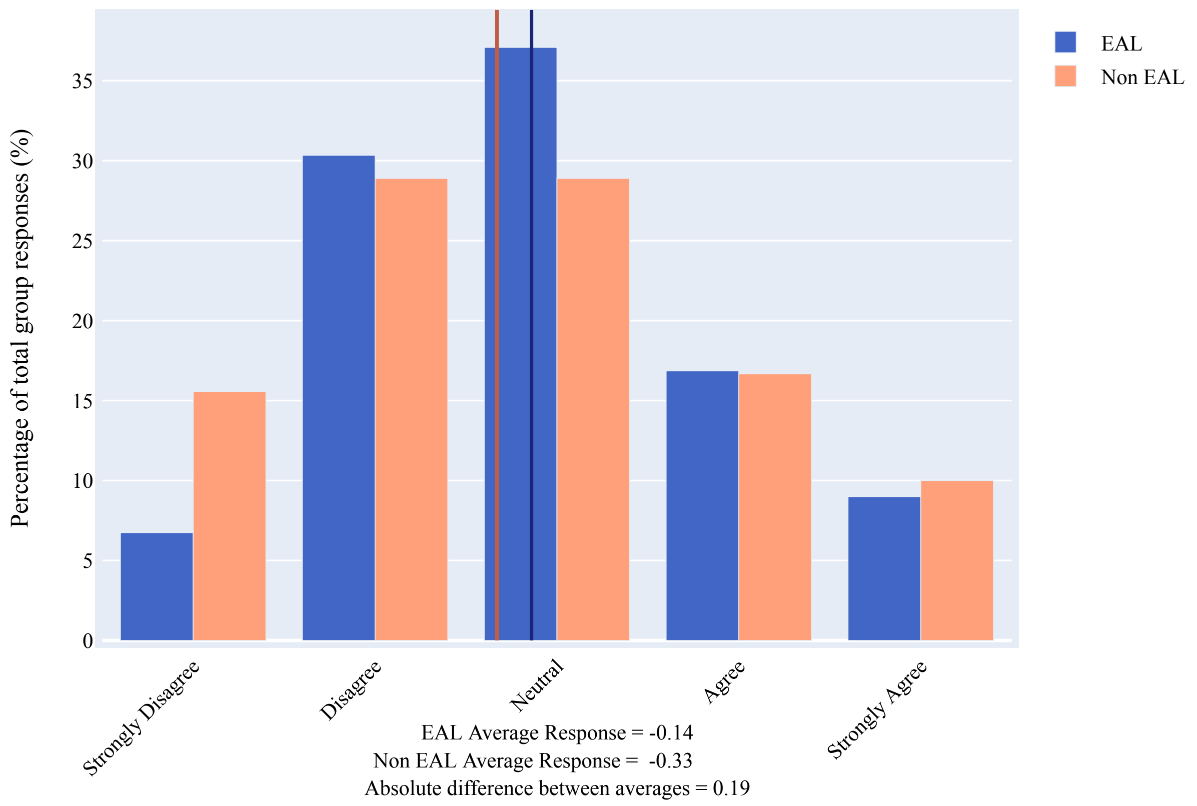


Figure 2: Dual bar graph showing the percentages of each group by responses.

Dual bar graph of responses to the statement "I am interested in a career involving computer science" comparing EAL and non EAL students.

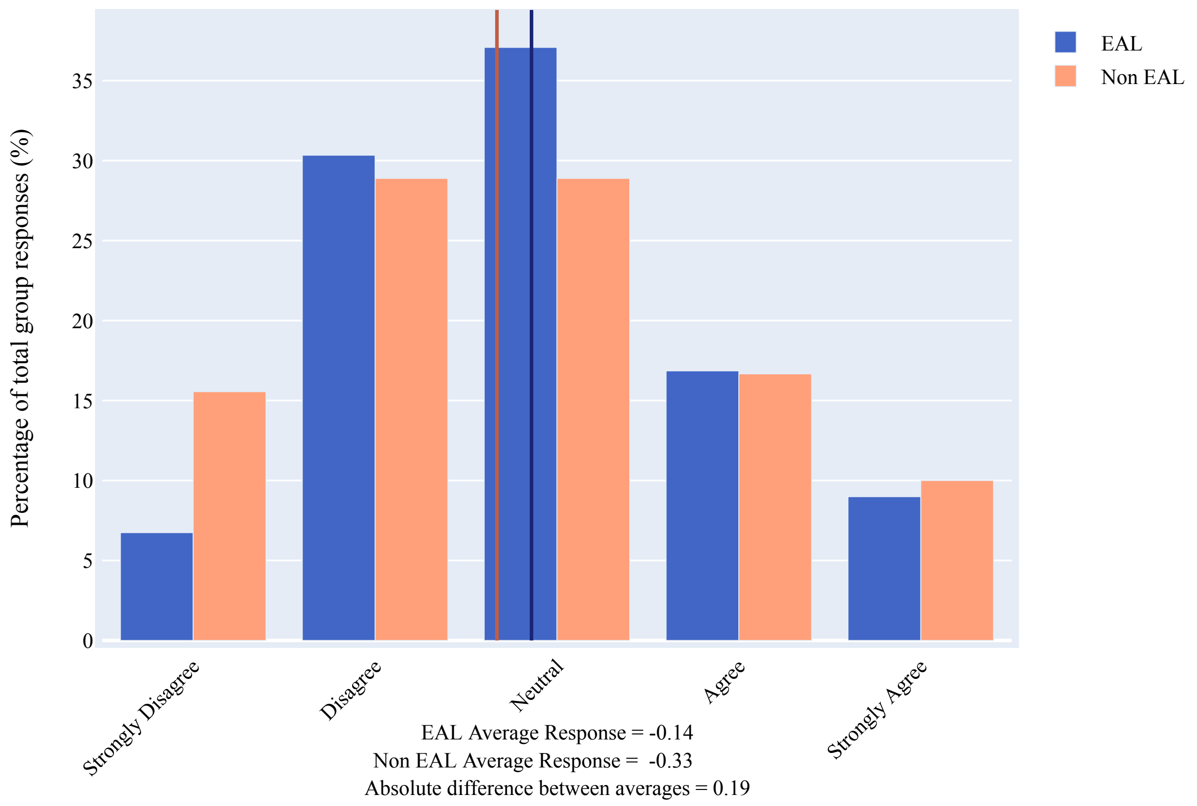


Figure 3: Dual bar graph showing the percentages of each group by responses.

3.2.2. Free School Meal Status

For each statement the average responses from FSM students was marginally lower than that of students not eligible for FSM. While this difference was constant across 7 of the 8 statements, the absolute differences in the average response was much smaller than group differences for other groups. The choice was therefore made to focus on the area of most significant discrepancy: gender.

3.2.3. Gender

Across the statements the largest discrepancy between responses was between genders, with the largest differences being a significantly lower in interest in computer science careers (Figure 4) and in feeling that computer science was "for people like them" (Figure 5) among girls. Girls were also less confident they could get jobs as programmers if they wanted to (Figure 6).

Dual bar graph of responses to the statement “I am interested in a career involving computer science” comparing male and female students.

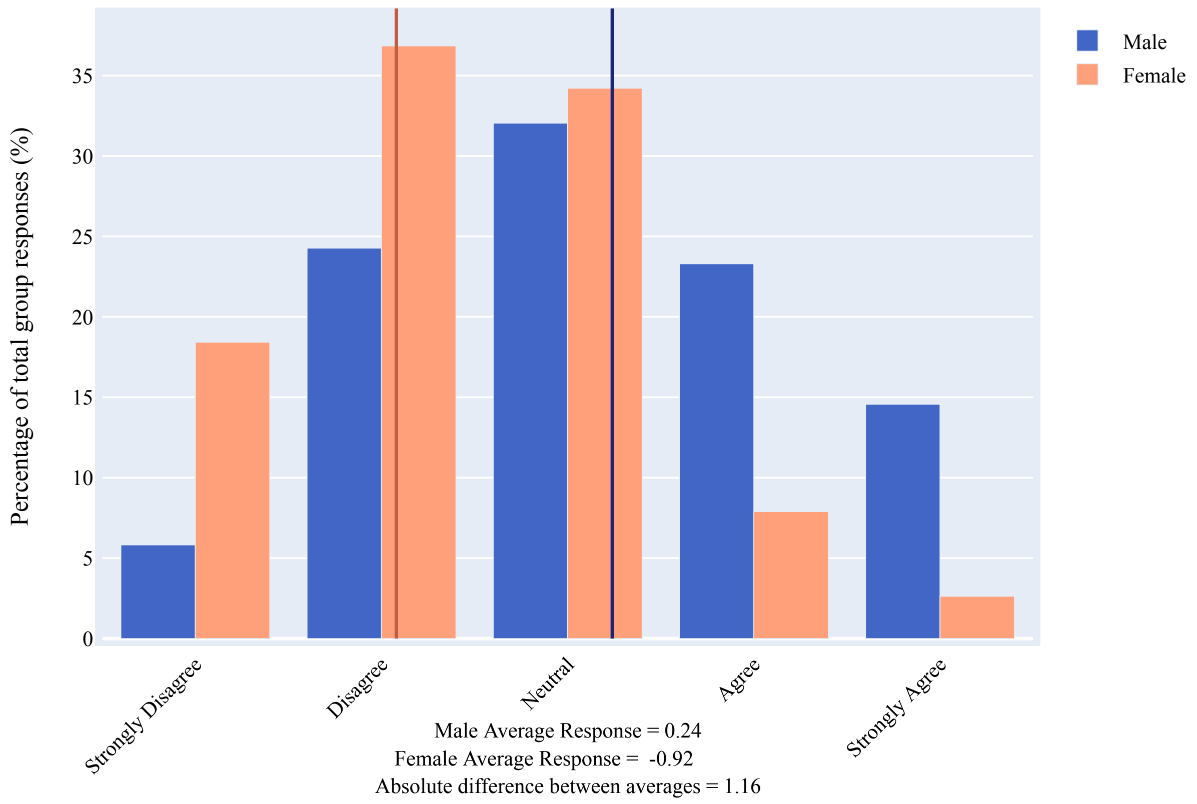


Figure 4: Dual bar graph showing the percentages of each group by responses.

Dual bar graph of responses to the statement "Computer science is for people like me" comparing male and female students.

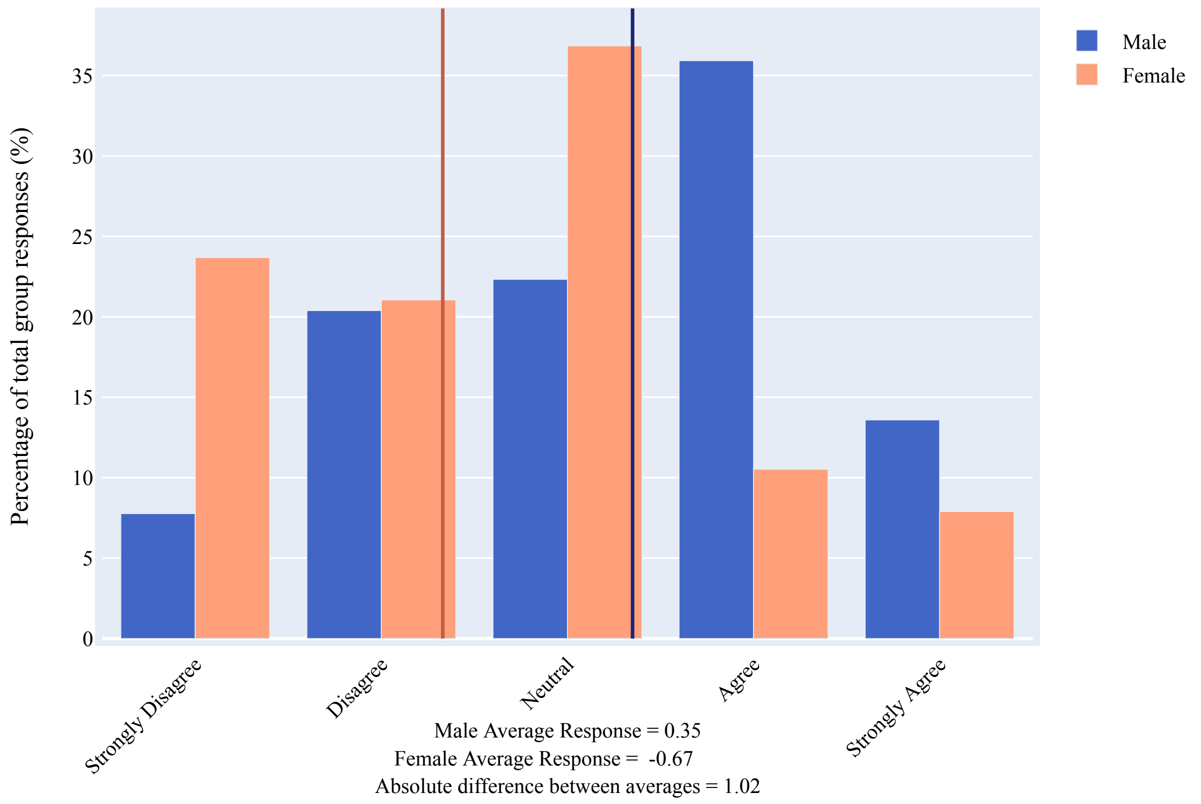


Figure 5: Dual bar graph showing the percentages of each group by responses.

Dual bar graph of responses to the statement "I could get a job as a programmer if I wanted to" comparing male and female students.

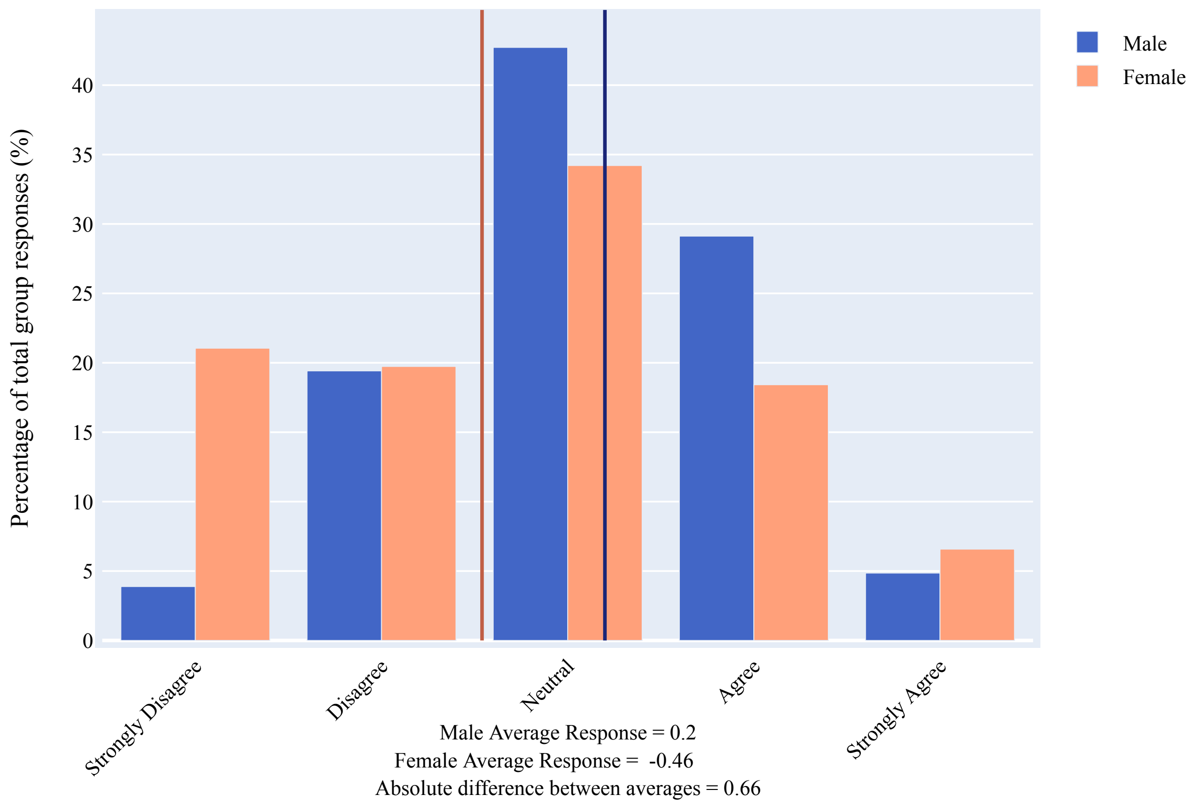


Figure 6: Dual bar graph showing the percentages of each group by responses.

Despite these differences, belief that studying computer science would lead to good jobs were high for all students and showed almost no gendered difference (Figure 7), with a similar trend present for responses to "There are lots of different types of jobs you can get by studying Computer Science" (Figure 8).

Dual bar graph of responses to the statement "Studying Computer Science leads to good jobs" comparing male and female students.

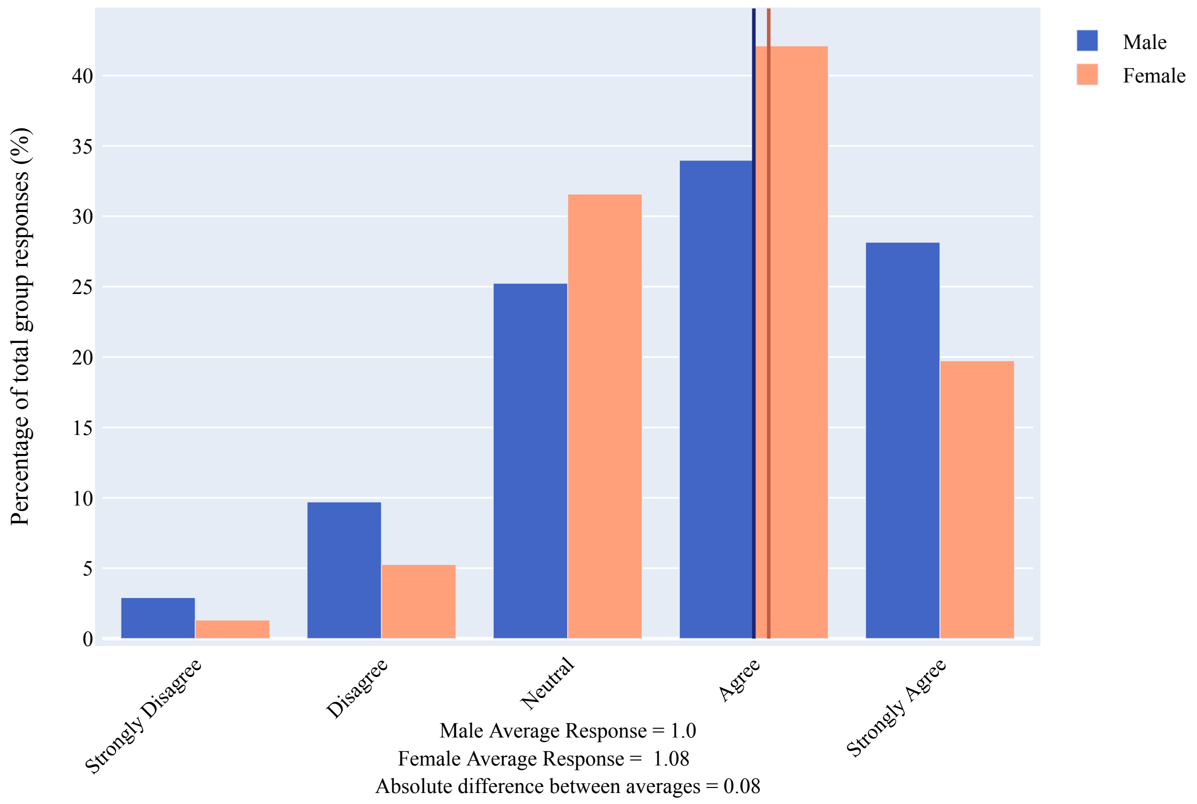


Figure 7: Dual bar graph showing the percentages of each group by responses.

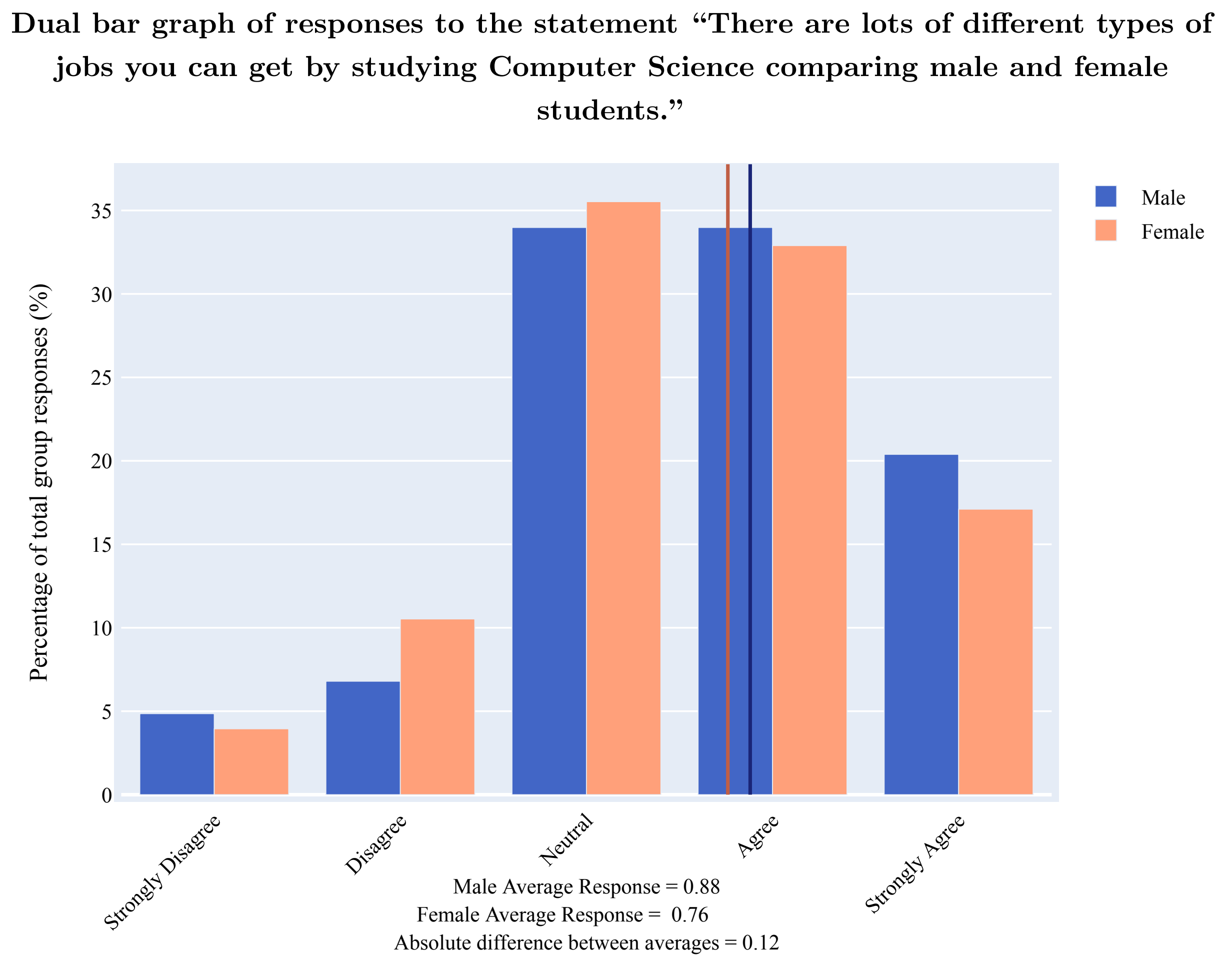


Figure 8: Dual bar graph showing the percentages of each group by responses.

Taking these data together suggests that girls are less confident in their skills, are less likely to feel they belong in the field and have lower interest in the subject and further careers. Comparing these data it is clear the strongest patterns in computer science engagement are occurring around gender.

To further examine the impact of gender I also analysed the students' target grades and attainment in the last summative assessment using data from SIMS. The average target grades of boys and girls showed no difference and the assessment grades showed a difference of less than one percent. This small difference may be accounted for by the girls' lower confidence in the subject.

Based on the negligible difference in attainment between the groups, the interventions were focused on building the confidence and interest of girls and tackling stereotypes of computer science, rather than targeting academic outcomes.

3.3. National Comparison

For the last decade there has been a vast and growing discrepancy in employment between genders in computer science. Women are greatly outnumbered in the digital economy, particularly in the highest earning sectors. This discrepancy has been recognised globally and there is a wealth of data outlining the extent of its prevalence. A 2022 US DHI Group report found an "average annual pay difference of nearly $8 600 between men and women" in tech (DHI Group, 2022) and only 19% of tech workers in the UK are women (Tech Nation, 2018). This is particularly concerning as the proportion of computing jobs in the economy is projected to grow substantially over the next decade. A 2021 report by the US Bureau of Labor Statistics stated that: "Computer occupations as a group are projected to grow about 3 times as fast as the average between 2019 and 2029; at 11,5 percent. This growth will result in slightly more than half a million new computer jobs over the 10-year period" (Bureau of Labor Statics, 2021).

This trend is pervasive from employment to degrees, A-Levels and GCSE entries (Frenkel, 1990) with the vast majority of institutions having women making up fewer than one in five computer science graduates (UNESCO, 2015). Below are the gender demographic breakdowns for students completing GCSE computer science from 2013 to 2021. These graphs show little to no progress has been made in closing the gender gap for approaching a decade (JCQ, 2021).

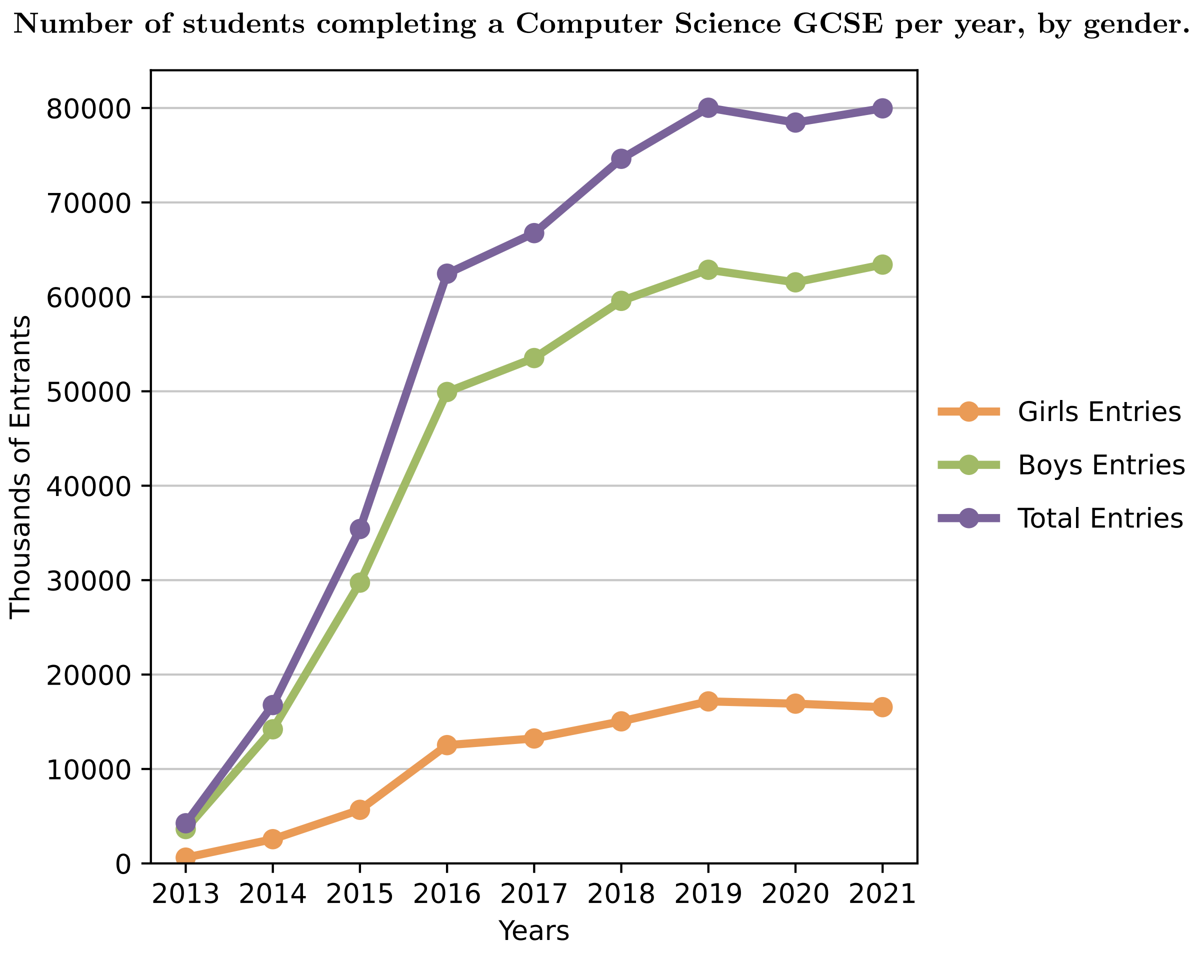


Figure 9: Graph of the percentage of overall GCSE exam completions per year in subjects: Maths, Physics and Computer Science based on JCQ results data (JCQ, 2021).

When comparing the percentage of GCSE completions by girls in computer science to other science, technology, engineering and mathematics (STEM) subjects (some of which also have a reputation for a gender imbalance e.g. economics), these subjects' discrepancies are dwarfed in comparison to the huge gap in computer science.

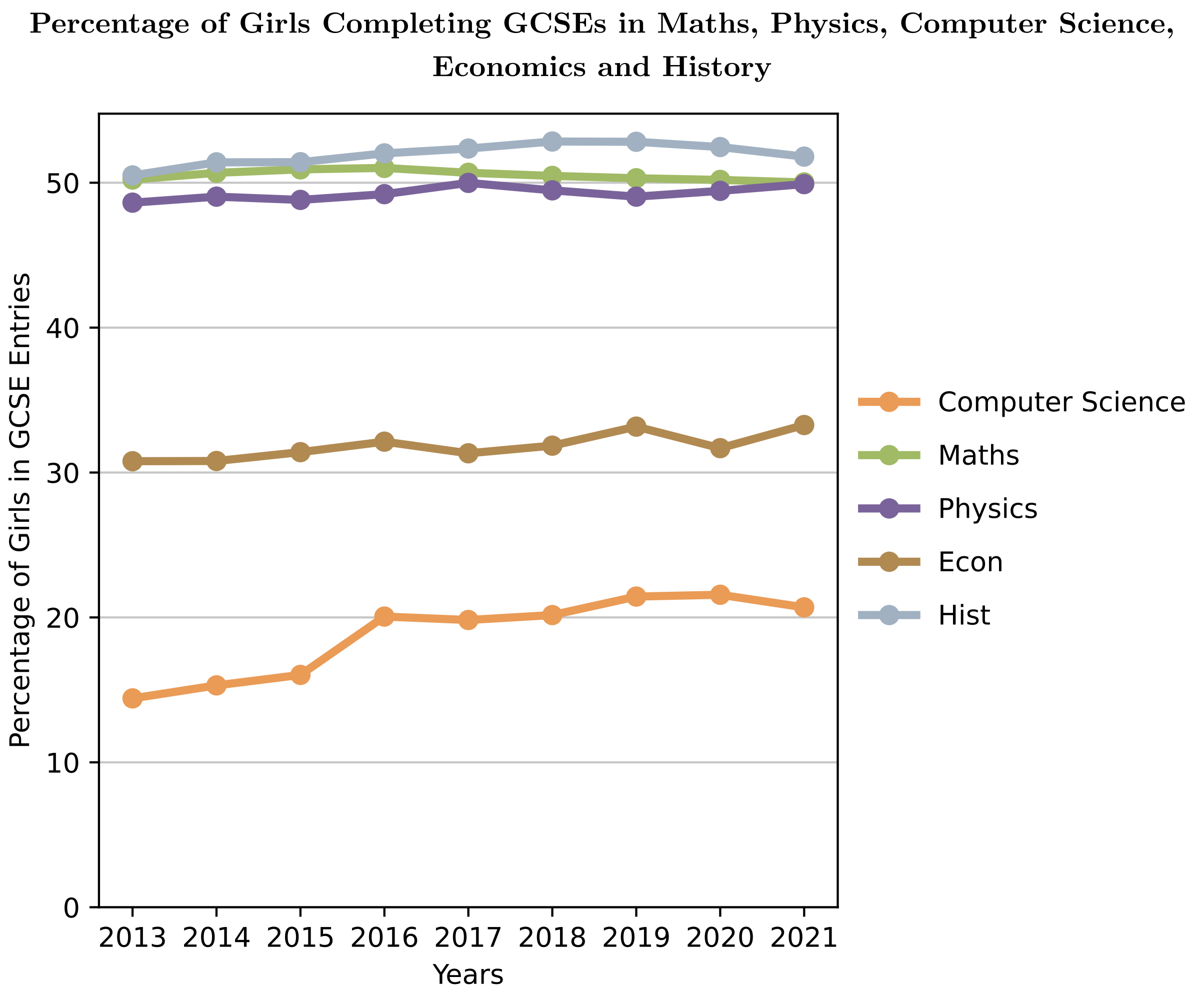
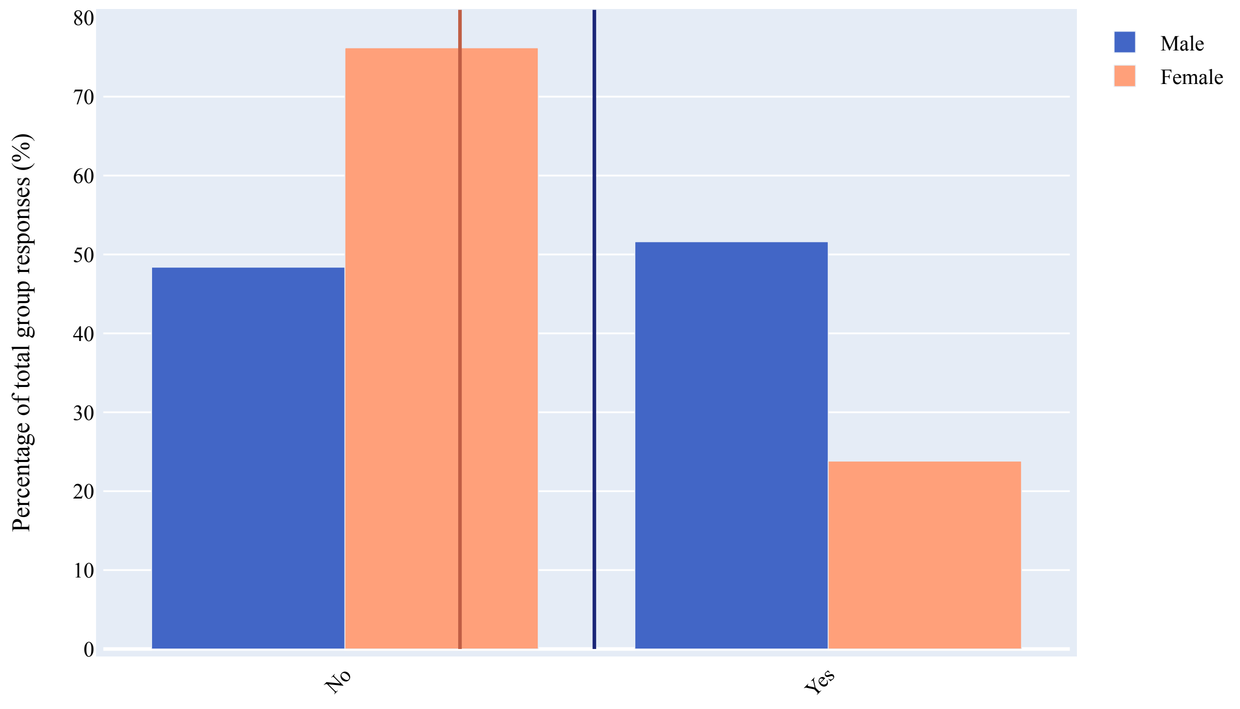


Figure 10: Graph of the percentage of overall GCSE exam completions per year in subjects: Maths, Physics, Computer Science, Echonomics and History based on JCQ results data (JCQ, 2021).

In comparison to the above, the data I collected showed that 52% of the year 9 boys who responded expressed interest in continuing computer science into GCSE while only 24% of girls did. While this is far from ideal, that would correspond to classes containing 31% girls; higher than the 2021 national rate of 20.7% (JCQ, 2021).

Dual bar graph of responses to the question "Do you plan on continuing to study Computer Science beyond your current level.



Male Average Response = 0.52
Female Average Response = 0.24
Absolute difference between averages = 0.28

Figure 11: Dual bar graph showing the percentages of each group by response.

3.4. Evidence Based Strategies to Increase Engagement

A 2016 Accenture report identified that the gender discrepancy trends originate in secondary schools, with too few girls taking computer science education past this point. However, the report further shows that well placed changes at school can improve uptake in computer science careers among girls. The report states that : "69% of the growth in the computing pipeline would come from changing the path of the youngest girls – especially those in junior high school" (students aged 12 to 15) (Accenture, 2016).

There are a range of evidence-backed strategies for building confidence and tackling stereotypes in computer science. An aggregated study examining 11 existing reviews identified a range of strategies that had seen measurable success in changing attitudes (Happe et al., 2020). The list of interventions outlined in this paper are vast and some are challenging to implement into most lessons. For example: providing all-female education programs and classes (Gürer and Camp, 2002;

García-Peñalvo et al., 2016), splitting classes by experience and not gender (Siiman et al., 2014), providing long-term self-directed projects (Brotman and Moore, 2008) and providing digital gaming and creative arts activities designed for girls (Main and Schimpf, 2017; Milam, 2012).

It was necessary to identify a small number of interventions that can be effectively planned into lessons. I selected: using collaborative assignments (Boston and Cimpian, 2018; Brotman and Moore, 2008; Nash, 2017), exposing girls to successful and relatable female role models in computing (Boston and Cimpian, 2018; Nash, 2017), providing opportunities to do computing activities as part of a group (Boston and Cimpian, 2018; Brotman and Moore, 2008; Nash, 2017) and providing non-stereotypical role models (Boston and Cimpian, 2018) (Happe et al., 2020).

I grouped the four interventions into two categories: group work (pair programming, to provide opportunities for collaborative work) and providing role models (classroom display, to promote diverse role models). The success of these techniques has been further shown in the study from Boston and Cimpian (2018) and summarised in the table below (Boston and Cimpian, 2018) (Figure 12).

Potential Strategies for Encouraging Pursuit of and Success in STEM Among (Gifted) Girls

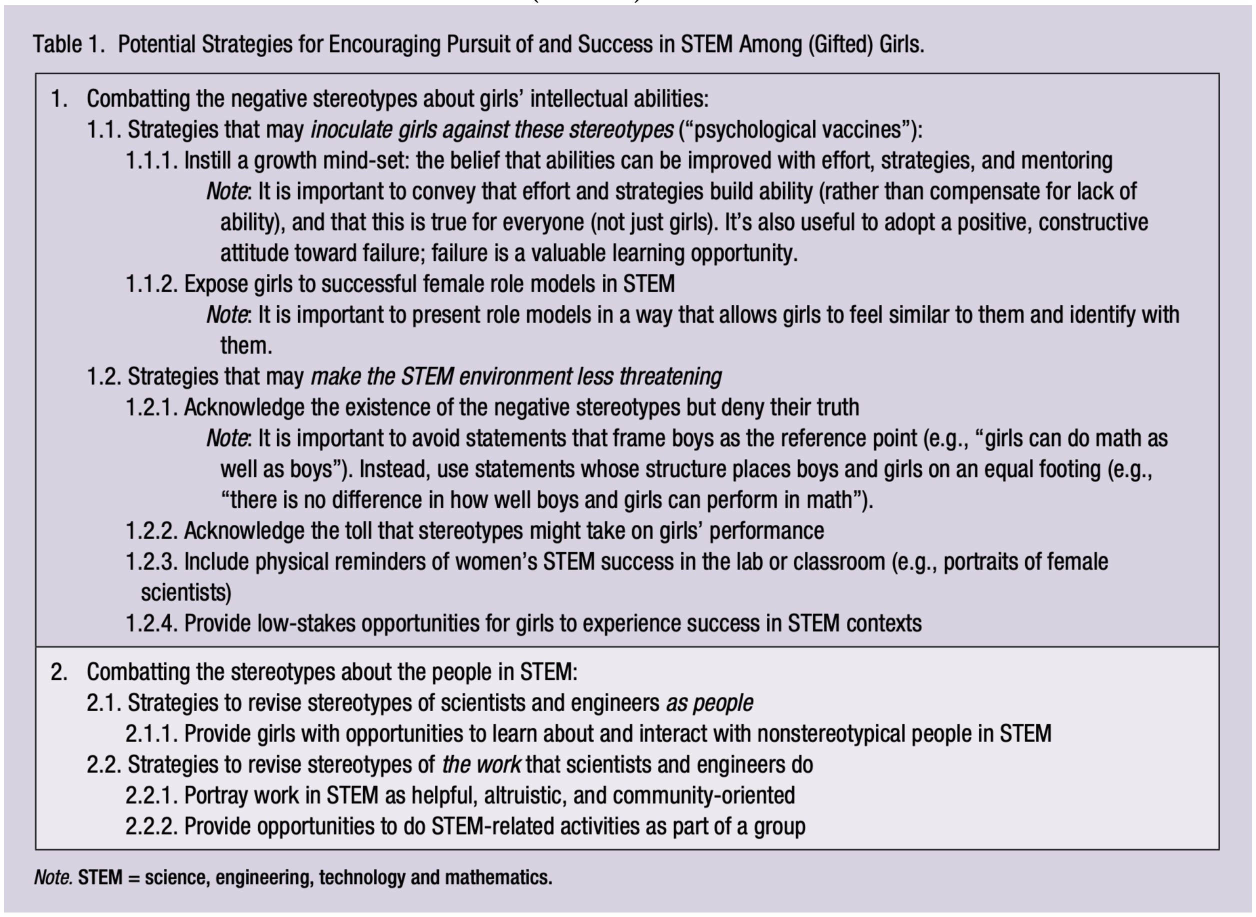


Table 1. Potential Strategies for Encouraging Pursuit of and Success in STEM Among (Gifted) Girls.

1. Combatting the negative stereotypes about girls' intellectual abilities:
    1.1. Strategies that may *inoculate girls against these stereotypes* ("psychological vaccines"):
        1.1.1. Instill a growth mind-set: the belief that abilities can be improved with effort, strategies, and mentoring
            Note: It is important to convey that effort and strategies build ability (rather than compensate for lack of ability), and that this is true for everyone (not just girls). It's also useful to adopt a positive, constructive attitude toward failure; failure is a valuable learning opportunity.
        1.1.2. Expose girls to successful female role models in STEM
            Note: It is important to present role models in a way that allows girls to feel similar to them and identify with them.
    1.2. Strategies that may *make the STEM environment less threatening*
        1.2.1. Acknowledge the existence of the negative stereotypes but deny their truth
            Note: It is important to avoid statements that frame boys as the reference point (e.g., "girls can do math as well as boys"). Instead, use statements whose structure places boys and girls on an equal footing (e.g., "there is no difference in how well boys and girls can perform in math").
        1.2.2. Acknowledge the toll that stereotypes might take on girls' performance
        1.2.3. Include physical reminders of women's STEM success in the lab or classroom (e.g., portraits of female scientists)
        1.2.4. Provide low-stakes opportunities for girls to experience success in STEM contexts

2. Combatting the stereotypes about the people in STEM:
    2.1. Strategies to revise stereotypes of scientists and engineers *as people*
        2.1.1. Provide girls with opportunities to learn about and interact with nonstereotypical people in STEM
    2.2. Strategies to revise stereotypes of *the work* that scientists and engineers do
        2.2.1. Portray work in STEM as helpful, altruistic, and community-oriented
        2.2.2. Provide opportunities to do STEM-related activities as part of a group

Note. STEM = science, engineering, technology and mathematics.

Figure 12: A table from Boston and Cimpian (2018) listing the interventions they suggest for encouraging pursuit and success of girls in STEM.

3.4.1. Pair Programming

Pair programming is a popular approach used in introducing programming where programming takes place between two students using the same computer. Each student in the pair takes one of two alternating roles: "driver" and "navigator". The driver is the student who uses the computer and types the code while the navigator assists the other verbally, giving instructions and spotting mistakes. It is also important to encourage pupils in the driver role to explain what they are doing and their thought process to allow the navigator to benefit from their modelling and spot logic errors or misconceptions. Pair programming has been shown to produce beneficial outcomes for all genders in a number of studies (Williams et al., 2000) (Williams and Upchurch, 2001). Research has particularly suggested pair programming as beneficial for women and girls (Werner et al., 2004), including Boston and Cimpian (2018), as show in bullet point 2.2.2 from the above Figure

12. Furthermore, a study of 104 student's responses to pair programming noted: "98 students expressed at least one positive sentiment about pair programming, for a total of 334 positive sentiments. Women attributed pair programming leading to a positive atmosphere, specifically by reducing negative emotions like discouragement, frustration, and stress and by creating an environment that was comforting, reassuring, and supportive" (Ying et al., 2019).

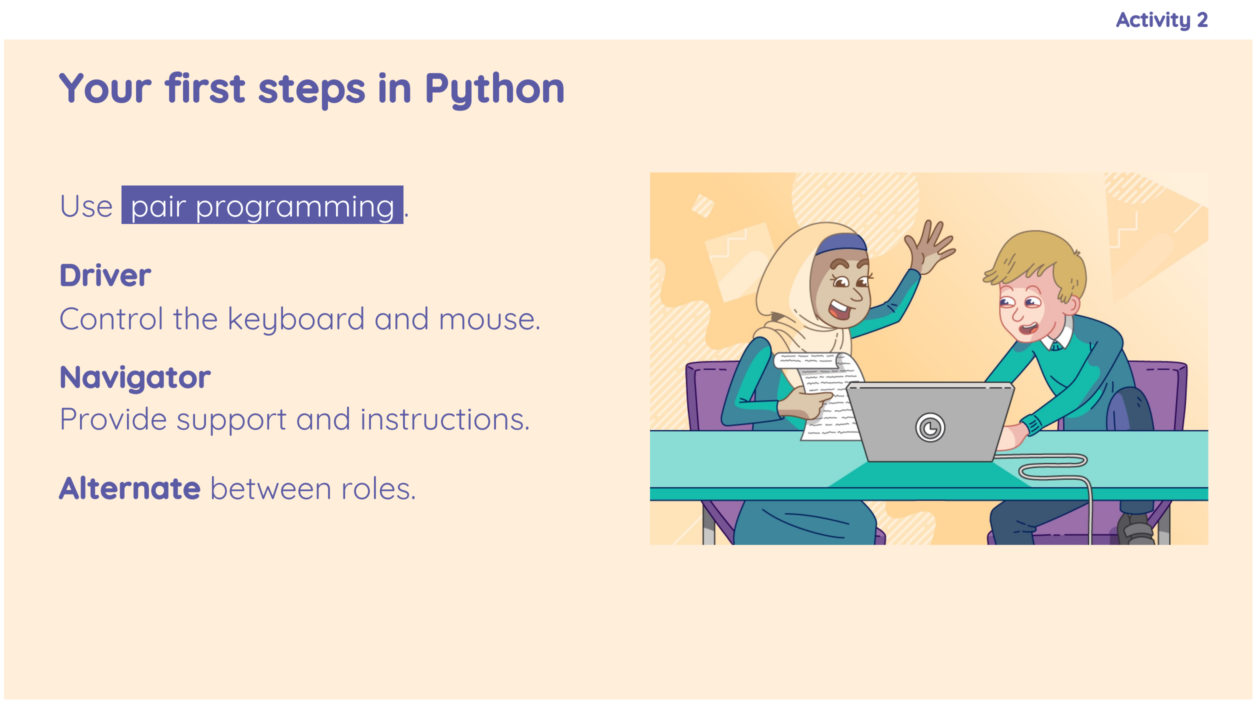
A further study showed increased positive sentiments from girls when they worked in same gendered pairs: "Same gender pairs, [FF] and [MM], exhibited significantly higher level of compatibility than mixed gender pair, [FM]." With some responses from women including: "my partner was male, and tended to not explain what he was thinking or doing which was sometimes frustrating." and "I think gender caused us to be a little more reserved around each other. If I programmed with a female I probably would have been more open." (Choi, 2015). The value of group work for women and girls in computer science was also a finding of (Boston and Cimpian, 2018) and (Nash, 2017).

Using pair programming during Python lessons is the first methodology I implemented in my lessons to improve the confidence and comfort on the girls in my classroom. Based on the above finding I aimed to pair girls together where possible in the classroom activities.

Happe et al. (2020) noted that "the interest of girls in computing drops early during primary and secondary education, with minimal recovery in later education stages." For this reason the Year 7 classes were selected as the subjects of this pair programming intervention.

When it came to planning lessons, I researched available resources that included specific use of peer programming. The National Centre for Computing Education (NCCE) provide a range of research-backed lesson resources through the Teach Computing platform. These resources are structured specifically around pair programming as an introduction to the topic (Teach Computing, 2020). I ensured I planned time for the students to swap roles and experience both sides of the pairing to reduce the potential for a single student to dominate the workload in the tasks and to necessitate a greater degree of collaboration between pupils.

Teach Computing KS3 Slide



Activity 2

Your first steps in Python

Use pair programming.

Driver
Control the keyboard and mouse.

Navigator
Provide support and instructions.

Alternate between roles.

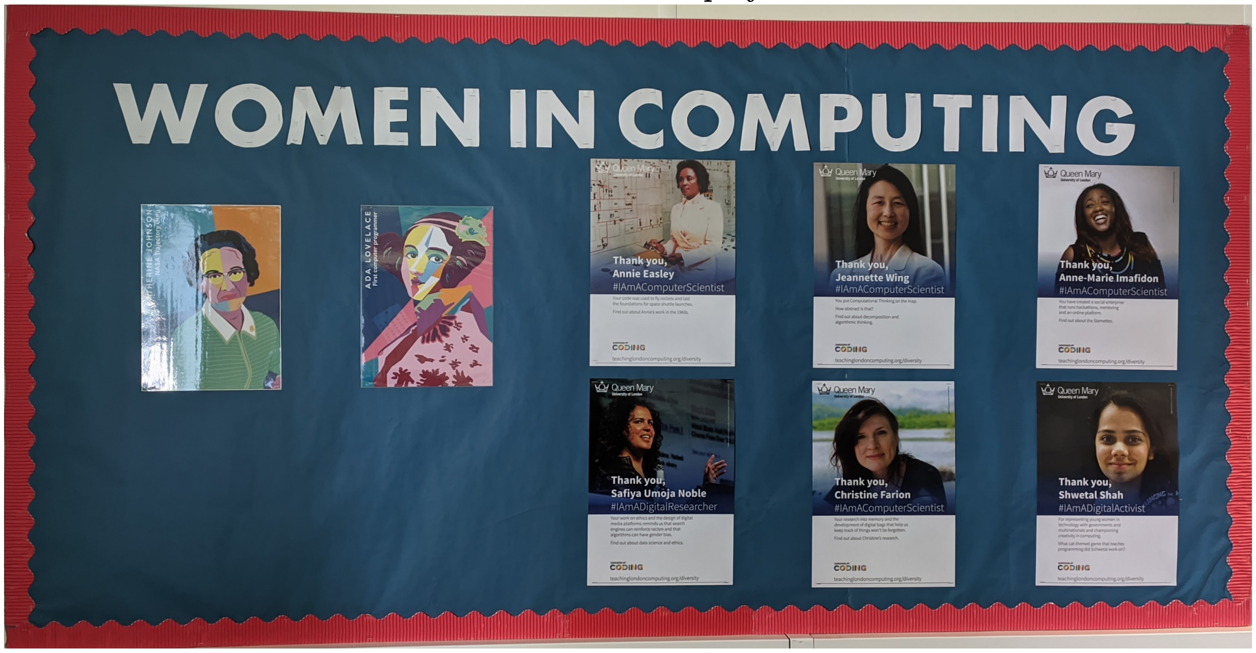
Figure 13: A slide explaining peer programming from the KS3 Teach Computing: "Introduction to Python programming" resources (Teach Computing, 2020).

3.4.2. Classroom Display

A second intervention I employed was a less direct one. Boston and Cimpian (2018) suggest the inclusion of physical reminders of women's success in computing, as seen in point 1.1.2 in Figure 12 above ('expose girls to successful female role models in STEM') and point 1.2.3 ('include physical reminders of women's STEM success in the lab or classroom (e.g. portraits of female scientists)'). The importance of selecting role models that allow girls to feel similar to and identify with is also emphasised. I chose to create a classroom display of young women in computing with photos and explanations of their contributions to the field. As such I chose to display mostly younger and contemporary figures in technological fields. I also ensured I selected women who represented a diverse range of ethnic backgrounds to best match the diversity of the students I teach and leave no students out.

The primary source I used for these images was a set of posters celebrating diversity in computing created by Queen Mary University (available at https://equalities.eecs.qmul.ac.uk/2019/01/16/posters-celebrating-diversity-in-computer-science/). These were selected as they contained an ethically diverse group of women as well as many younger women. Based on these recommendations the following display was constructed:

Classroom Display



4. Analysis

4.1. Quantitative Analysis

After displaying the new posters and using peer programming in a series of 5 lessons I asked pupils to complete the survey again to see if responses had changed. One limitation of this approach is the short series of lessons over which changes can be observed. Key Stage 3 students only have one hour of computer science per week, limiting the number of hours the students are exposed to the intervention. This is exacerbated by the deeply engrained nature of attitudes and stereotypes. These beliefs can be gathered from sources from parents, role models, television, social media and film and will develop over a period of years. As such feelings like "computer science is not for people like me" are unlikely to be meaningfully changed over a period of a few hours.

Post intervention bar graph of responses to the statement "Computer science is for people like me" comparing male and female students.

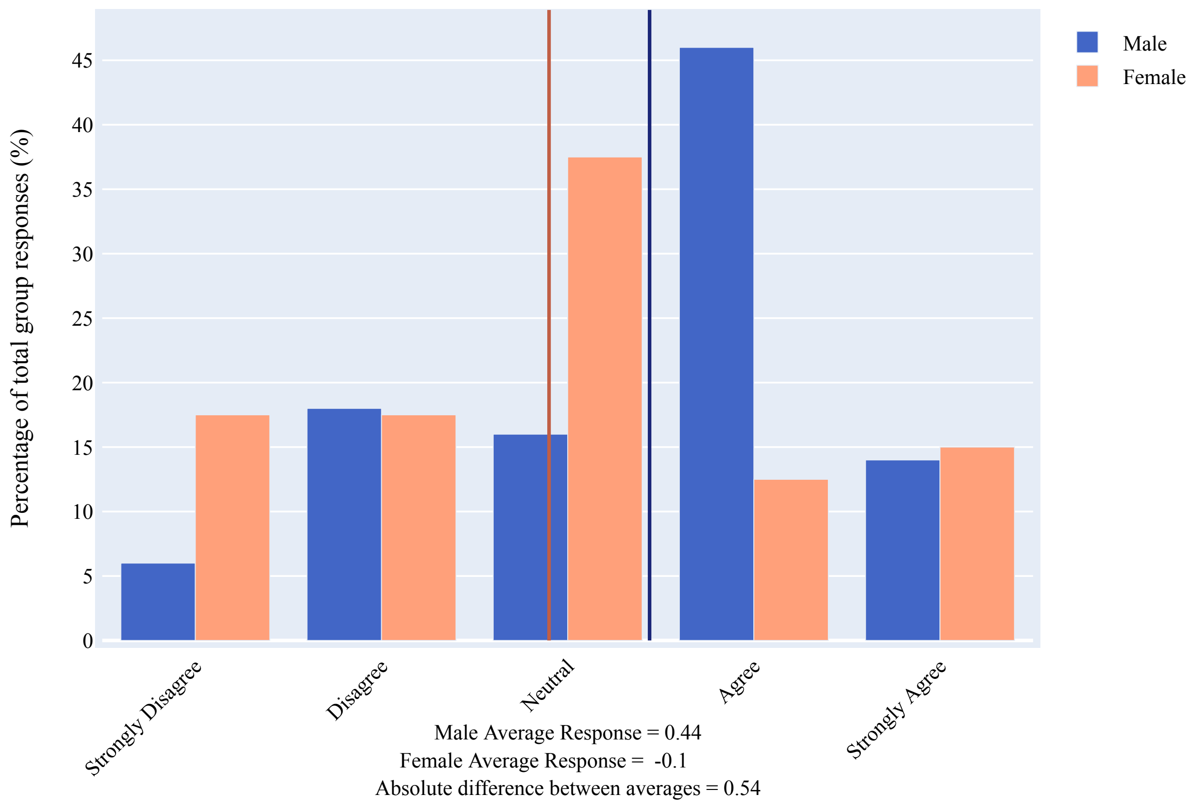


Figure 14: Dual bar graph showing the percentages of each group by responses.

Post intervention bar graph of responses to the statement "I am interested in a career involving computer science" comparing male and female students.

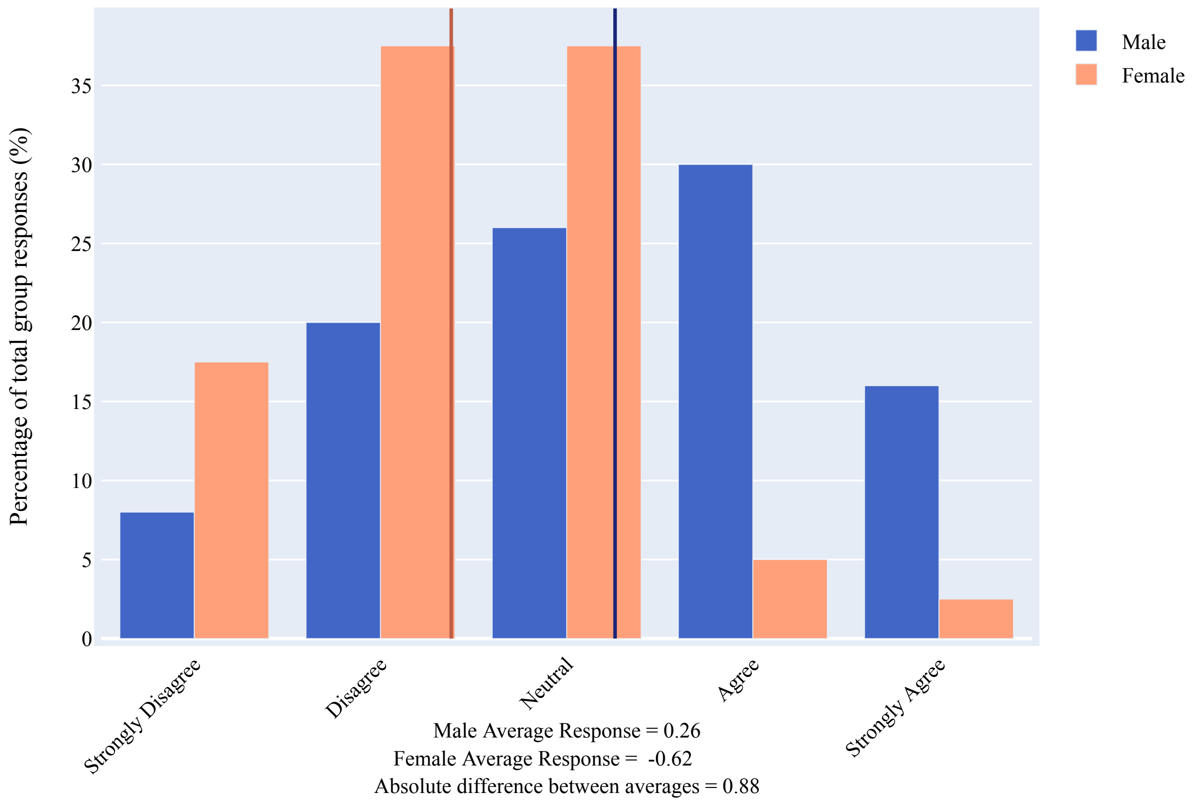


Figure 15: Dual bar graph showing the percentages of each group by responses.

Despite these limitations a moderate positive change in responses was shown in the data collected. It is difficult to attribute a causal relation between these changes due to the lack of a control group to compare to and the wealth of other factors that may have contributed to the changes. For example after the survey and putting up the display, pupils understood I was trying to specifically inspire girls. This knowledge may have changed how they answered regardless of whether their underlying attitudes had changed. Equally the influence of presenting the survey a second time is unknown. Furthermore, due to absences and exclusions the group of pupils who took the survey the first time were not identical to the group the second time. Given more time it would have been beneficial to cross reference responses and examine how individual student's responses change. To gain a better understanding of the causal relationship these pupils could have been interviewed on their opinions to ascertain the reasoning behind the change.

The boys' responses can to some extent be used as a control, with those attitudes also growing more positive but to a lesser extent than the girls. The absolute difference in average responses to

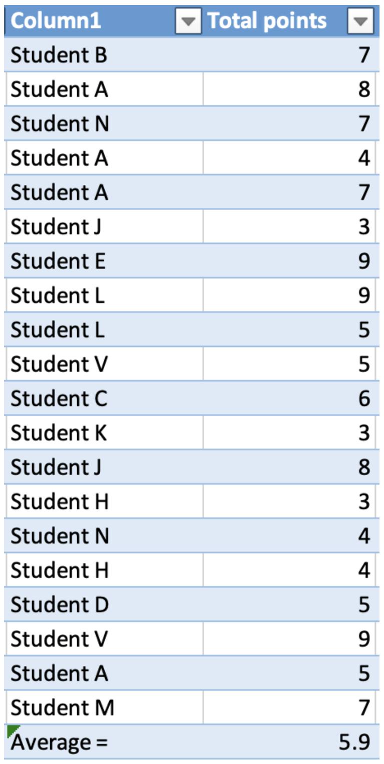
"Computer science is for people like me" was 0,54 after the intervention compared to 1,02 before intervention (where a difference of 1 represents the difference between neutral and agree or agree and strongly agree etc.). In particular the average female response changed from -0,67 to -0,1.

4.2. Qualitative Analysis

Whilst pair programming was chosen because of the proven benefit to girls, it was interesting to observe and reflect on the effects of this intervention on the whole class. While the data showed a moderate change in girls' attitudes, reflecting on the lessons themselves and the response of all the students can also be a useful tool to evaluate the success of the lessons. Almost all pupils were very enthusiastic about working collaboratively in the lesson and the standard of work was high for these students. Many of the pupils who have struggled more with Python were better able to access the lessons using the support of their partner. It was encouraging to see pupils guiding and teaching each other when they were in the role of the navigator. Despite many of these strengths there were also many challenges to delivering these lessons. For example navigating the pupil relationships so that pupils were partnered with people who they felt comfortable with but who would not be a distraction was complex. For example, in the first lesson using pair programming, one pair refused to work with each other as they did not get on. This first lesson where groups were assigned involved moving pupil's seats and some arguments from students. Ultimately this ended up wasting lesson time. Students were also required to talk in order to carry out both driver and navigator roles which lead to a rising noise level and difficulty in tracking which groups were straying off task. Given the challenging behaviour at the school generally and my more novice behaviour management skills, this may not have been the ideal setting to test this teaching approach. When employing group work in the future I would label computers with numbers and direct students to that machine at the very start to cut down on time wasted moving pupils. The routines of pair programming, such as the assignment and switching of roles and how to work effectively as pair also took time to establish, however after several lessons these routines became ingrained and the efficiency of the lessons increased greatly. This may suggest that a prolonged period of intervention may have produced a greater degree of impact on academic and attitude changes.

While the primary concern of the interventions was on girls' attitudes and opinions, it is also important to ensure that these changes do not come at the cost of academic performance and attainment. Student work is another way to evaluate the efficacy of the lessons, however student's programmes are completed in pairs and as such it can be challenging to discern whether an individual has met learning objectives. To better gather data on individual students, each lesson concluded with a quiz on the lesson topic which students were asked to complete individually. The quiz scores provided feedback on the learning of each pupil's progress while working in pairs. While the results of these quizzes demonstrated that students met lesson objectives, the lack of a control group made it impossible to discern whether the use of pair programming had a positive or negative impact compared with programming individually.

Anonymised table of one classes scores on the variables quiz (see appendix 6 for questions).



Column1	Total points
Student B	7
Student A	8
Student N	7
Student A	4
Student A	7
Student J	3
Student E	9
Student L	9
Student L	5
Student V	5
Student C	6
Student K	3
Student J	8
Student H	3
Student N	4
Student H	4
Student D	5
Student V	9
Student A	5
Student M	7
Average =	5.9

Figure 16: Anonymised marks out of 10 for the quiz covering use and apropritate naming of variables.

5. Conclusion

There are multiple conclusions that can be drawn from analysis of these data. The engagement, interest and self-belief of pupils in computer science is largely consistent across ethnicity, language and pupil premium status, with EAL status being a particular strength for my pupils. However, as has been demonstrated across computer science education and employment, the subject is failing to appeal effectively to women and girls. The data I collected showed that girls are less likely to believe the subject is for them and show less interest in both the subject and further careers. Despite these shortcomings there is strong evidence that changes to how the curriculum is delivered are effective in growing this interest (Happe et al., 2020). Overall, while a causal relationship is difficult to establish, the efficacy of these interventions demonstrated in other research were supported by the data from my classroom, with feelings of belonging growing among all pupils but more significantly with girls.

There is strong reason to believe that continuing to employ these changes over a prolonged period of time may be able to further close the interest gap between genders. Going forward there is other research-backed interventions that can be employed in curriculum areas outside of programming such as framing computer science in the context of helping people and positive societal impacts (Boston and Cimpian, 2018) (Barker and Aspray, 2006). I will aim to consider my framing in this context as I plan future lessons.

The impact on academic performance from these interventions compared to other teaching approaches could not be determined due to a lack of control group, although pupils were largely successful in meeting key lesson objectives. Challenges with pair programming were also identified, primarily in building routines, managing pupil behaviour, ensuring pairs remained on task and managing a rise in noise levels. With some changes to how group pairs are introduced and with more experience in behaviour management, the pair programming activities would likely have been more valuable to pupil learning.

References

Accenture (2016). *Cracking the gender code: Get 3x more women in computing.*

Barker, L. and Aspray, W. (2006). *1 the state of research on girls and it.*

Boston, J. S. and Cimpian, A. (2018). *How do we encourage gifted girls to pursue and succeed in science and engineering?.* Gifted Child Today **41**(4), 196–207.

Bureau of Labor Statics (2021). *Why computer occupations are behind strong stem employment growth in the 2019–29 decade.* US Bureau of Labor Statistics .

Choi, K. S. (2015). *A comparative analysis of different gender pair combinations in pair programming.* Behaviour & Information Technology **34**(8), 825–837.

DHI Group (2022). *The gender pay gap in tech.* DHI Group .

El-Lahib, Y., George, P., Pon, G. and Wehbi, S. (2011). *Challenging the myth of" studying harder": A social work response to the oppression of" eal" students.* Canadian Social Work Review/Revue Canadienne de Service Social pp. 209–223.

Frenkel, K. A. (1990). *Women and computing.* Commun. ACM **33**(11), 34–46.

Happe, L., Buhnova, B., Koziolek, A. and Wagner, I. (2020). *Effective measures to foster girls' interest in secondary computer science education.* Education and Information Technologies **26**(3), 2811–2829.

Hatch End High School (2022). *Equality information and objectives 2022-2023.*

JCQ (2021). *Jcq gcse examination results.* Joint Council of Qualifications (JCQ) .

Johnson, E. M. (2008). *An investigation into pedagogical challenges facing international tertiary-level students in new zealand.* Higher Education Research & Development **27**(3), 231–243.

Nash, J. (2017). Understanding how to interest girls in stem education: a look at how lego® education ambassador teachers engage female students in stem learning. PhD thesis. University of Florida.

Teach Computing (2020). *Introduction to python programming.*

Tech Nation (2018). *Diversity and inclusion in uk tech companies.*

UNESCO (2015). *UNESCO science report: towards 2030.* UNESCO.

Werner, L. L., Hanks, B. and McDowell, C. (2004). *Pair-programming helps female computer science students.* J. Educ. Resour. Comput. **4**(1), 4–es.

Williams, L., Kessler, R., Cunningham, W. and Jeffries, R. (2000). *Strengthening the case for pair programming.* IEEE Software **17**(4), 19–25.

Williams, L. and Upchurch, R. L. (2001). In support of student pair-programming. *in* 'Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education'. SIGCSE '01. Association for Computing Machinery. New York, NY, USA. p. 327–331.

Ying, K. M., Pezzullo, L. G., Ahmed, M., Crompton, K., Blanchard, J. and Boyer, K. E. (2019). In their own words: Gender differences in student perceptions of pair programming. *in* 'Proceedings of the 50th ACM Technical Symposium on Computer Science Education'. SIGCSE '19. Association for Computing Machinery. New York, NY, USA. p. 1053–1059.

6. Appendix

6.1. 1. Website HTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>Further CS Form</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
  <link href='https://fonts.googleapis.com/css?family=Noto Serif' rel='stylesheet'>
  <link href='https://fonts.googleapis.com/css?family=Open Sans' rel='stylesheet'>
</head>
<body>
  <div class="wrap">
  <h1 >Careers and Further Education Form</h1>
  </br>
</br>
  <p>Fill in <strong>every</strong> question below and then click the submit button at the end
</br>
  <form name="frmSubmit" action="reponses.php", method="post">
      <div id="nameClass">
          <div id="name">
              <label>Enter your name: </label>
              <input type="text" id="name" name="name" required>
            </div><div id="class">
                <label>Enter your class (e.g. 9OFL): </label>
                <input type="text" name="class" required>
            </div>
        </div>
    <label id="Question1" class="statement">Do you plan on continuing to study Computer Science
    <ul class='likert'>
        <li>
            <input type="radio" name="likert" value="yes" required>
            <label>Yes</label>
        </li>
        <li>
            <input type="radio" name="likert" value="no">
            <label>No</label>
        </li>
    </ul>
    <label id="q0" class="statement">Error - Questions not updated.</label>
    <ul class='likert'>
        <li>
            <input type="radio" name="likert0" value="strong_disagree" required>
            <label>Strongly disagree</label>
        </li>
        <li>
            <input type="radio" name="likert0" value="disagree">
            <label>Disagree</label>
        </li>
        <li>
            <input type="radio" name="likert0" value="neutral">
            <label>Neutral</label>
        </li>
        <li>
            <input type="radio" name="likert0" value="agree">
            <label>Agree</label>
        </li>
        <li>
          <input type="radio" name="likert0" value="strong_agree">
          <label>Strongly agree</label>
        </li>
    </ul>
    <label id="q1" class="statement">Error - Questions not updated.</label>
    <ul class='likert'>
        <li>
            <input type="radio" name="likert1" value="strong_disagree" required>
            <label>Strongly disagree</label>
        </li>
        <li>
            <input type="radio" name="likert1" value="disagree">
            <label>Disagree</label>
        </li>
        <li>
            <input type="radio" name="likert1" value="neutral">
            <label>Neutral</label>
        </li>
        <li>
            <input type="radio" name="likert1" value="agree">
            <label>Agree</label>
        </li>
        <li>
          <input type="radio" name="likert1" value="strong_agree">
          <label>Strongly agree</label>
        </li>
    </ul>
```

```html
<label id="q2" class="statement">Error - Questions not updated.</label>
<ul class='likert'>
    <li>
        <input type="radio" name="likert2" value="strong_disagree" required>
        <label>Strongly disagree</label>
    </li>
    <li>
        <input type="radio" name="likert2" value="disagree">
        <label>Disagree</label>
    </li>
    <li>
        <input type="radio" name="likert2" value="neutral">
        <label>Neutral</label>
    </li>
    <li>
        <input type="radio" name="likert2" value="agree">
        <label>Agree</label>
    </li>
    <li>
      <input type="radio" name="likert2" value="strong_agree">
      <label>Strongly agree</label>
    </li>
</ul>
<label id="q3" class="statement">Error - Questions not updated.</label>
<ul class='likert'>
    <li>
        <input type="radio" name="likert3" value="strong_disagree" required>
        <label>Strongly disagree</label>
    </li>
    <li>
        <input type="radio" name="likert3" value="disagree">
        <label>Disagree</label>
    </li>
    <li>
        <input type="radio" name="likert3" value="neutral">
        <label>Neutral</label>
    </li>
    <li>
```

```
        <input type="radio" name="likert3" value="agree">
        <label>Agree</label>
    </li>
    <li>
      <input type="radio" name="likert3" value="strong_agree">
      <label>Strongly agree</label>
    </li>
</ul>
<label id="q4" class="statement">Error - Questions not updated.</label>
<ul class='likert'>
    <li>
        <input type="radio" name="likert4" value="strong_disagree" required>
        <label>Strongly disagree</label>
    </li>
    <li>
        <input type="radio" name="likert4" value="disagree">
        <label>Disagree</label>
    </li>
    <li>
        <input type="radio" name="likert4" value="neutral">
        <label>Neutral</label>
    </li>
    <li>
        <input type="radio" name="likert4" value="agree">
        <label>Agree</label>
    </li>
    <li>
      <input type="radio" name="likert4" value="strong_agree">
      <label>Strongly agree</label>
    </li>
</ul>
<label id="q5" class="statement">Error - Questions not updated.</label>
<ul class='likert'>
    <li>
        <input type="radio" name="likert5" value="strong_disagree" required>
        <label>Strongly disagree</label>
    </li>
    <li>
```

```
        <input type="radio" name="likert5" value="disagree">
        <label>Disagree</label>
    </li>
    <li>
        <input type="radio" name="likert5" value="neutral">
        <label>Neutral</label>
    </li>
    <li>
        <input type="radio" name="likert5" value="agree">
        <label>Agree</label>
    </li>
    <li>
      <input type="radio" name="likert5" value="strong_agree">
      <label>Strongly agree</label>
    </li>
</ul>
<label id="q6" class="statement">Error - Questions not updated.</label>
<ul class='likert'>
    <li>
        <input type="radio" name="likert6" value="strong_disagree" required>
        <label>Strongly disagree</label>
    </li>
    <li>
        <input type="radio" name="likert6" value="disagree">
        <label>Disagree</label>
    </li>
    <li>
        <input type="radio" name="likert6" value="neutral">
        <label>Neutral</label>
    </li>
    <li>
        <input type="radio" name="likert6" value="agree">
        <label>Agree</label>
    </li>
    <li>
      <input type="radio" name="likert6" value="strong_agree">
      <label>Strongly agree</label>
    </li>
    </ul>
    <label id="q7" class="statement">Error - Questions not updated.</label>
    <ul class='likert'>
        <li>
            <input type="radio" name="likert7" value="strong_disagree" required>
            <label>Strongly disagree</label>
        </li>
        <li>
            <input type="radio" name="likert7" value="disagree">
            <label>Disagree</label>
        </li>
        <li>
            <input type="radio" name="likert7" value="neutral">
            <label>Neutral</label>
        </li>
        <li>
            <input type="radio" name="likert7" value="agree">
            <label>Agree</label>
        </li>
        <li>
          <input type="radio" name="likert7" value="strong_agree">
          <label>Strongly agree</label>
        </li>
    </ul>
    <input type="hidden" id="Q1Type" name = "Q1Type" value="na">
    <input type="hidden" id="Q2Type" name = "Q2Type" value="na">
    <input type="hidden" id="Q3Type" name = "Q3Type" value="na">
    <input type="hidden" id="Q4Type" name = "Q4Type" value="na">
    <input type="hidden" id="Q5Type" name = "Q5Type" value="na">
    <input type="hidden" id="Q6Type" name = "Q6Type" value="na">
    <input type="hidden" id="Q7Type" name = "Q7Type" value="na">
    <input type="hidden" id="Q8Type" name = "Q8Type" value="na">

    <div class="buttons">
      <button class="clear" type="reset">Clear</button>
      <button class="submit">Submit</button>
    </div>
  </form>
```

```
</div>
  <script src="script.js"></script>
</body>

</html>
```

6.2. 2. Website CSS

```
p{
    font-size: 12pt;
    font-family: "Open Sans", "Arial", "Helvetica", "sans-serif";
    text-align: left;
    margin-left: 9%;
}

h1{
    font-size: 28pt;
    text-align: center;
    background-color: #232323;
    color: #ffffff;
    margin: 0px;
    padding: 30px;
    font-family: "Noto-Serif", "Georgia", "serif";
}

body{
  color: #232323;
  background-color: #ffffff;
  line-height: 1.38;
  margin-top: 0pt;
  margin-bottom: 0pt;
  font-weight: 400;
  font-style: normal;
  font-variant: normal;
  text-decoration: none;
  vertical-align: baseline;
/*   white-space: pre-wrap; */
}
```

```css
#nameClass{
  display: flex;
  justify-content: center;
  gap: 5%;
  margin: 10px;
}


form {
  font-size: 12pt;
  color: #232323;
  font-family: "Open Sans", "Arial", "Helvetica", "sans-serif";
  /* Center the form on the page */
  margin: 0 auto;
  width: 80%;
  /* Form outline */
  padding: 1em;
  border: 1px solid #CCC;
  border-radius: 1em;
}

html,body {padding:0;margin:0;}
.wrap {
  font:15px Open Sans, Arial, Helvetica, sans-serif;
}
h1.likert-header {
  padding-left:4.25%;
  margin:20px 0 0;
}
form .statement {
  display:block;
  font-size: 16px;
  font-weight: bold;
  padding: 30px 0 0 4.25%;
  margin-bottom:10px;
}
form .likert {
  list-style:none;
  width:100%;
  margin:0;
  padding:0 0 35px;
  display:block;
  border-bottom:2px solid #efefef;
}
form .likert:last-of-type {border-bottom:0;}
form .likert:before {
  content: '';
  position:relative;
  top:11px;
  left:9.5%;
  display:block;
  background-color:#efefef;
  height:4px;
  width:78%;
}
form .likert li {
  display:inline-block;
  width:19%;
  text-align:center;
  vertical-align: top;
}
form .likert li input[type=radio] {
  display:block;
  position:relative;
  top:0;
  left:50%;
  margin-left:-6px;

}
form .likert li label {width:100%;}
form .buttons {
  font-size: 30pt;
  margin:30px 0;
  padding:0 4.25%;
  text-align:right
```

```
}
form .buttons button {
  padding: 5px 10px;
  background-color: #67ab49;
  border: 0;
  border-radius: 6px;
}
form .buttons .clear {background-color: #e9e9e9; font-size: 30pt;}
form .buttons .submit {background-color: #67ab49; font-size: 30pt;}
form .buttons .clear:hover {background-color: #ccc; font-size: 30pt;}
form .buttons .submit:hover {background-color: #14892c; font-size: 30pt; color: #fff;}
```

6.3. 3. Website JavaScript

```
var questions = [
    ["Learning Computer Science will be useful to me in the future.", "Learning Computer Scien
    ["Studying Computer Science leads good jobs.", "Computer Science does not lead to good jobs
    ["I am interested in a career involving Computer Science.", "I do not want a career involvi
    ["Computer Science is easy.", "Computer Science is hard."],
    ["If I wanted to I could get a job as a programmer.", "I could not get a job as a programme
    ["There are lots of different types of jobs you can get by studying Computer Science.", "Th
    ["Computer Science is interesting.", "Computer Science is boring."],
    ["Computer Science is for people like me.", "Computer Science is not for people like me."]
  ];

function getCookie(name) {
    // Split cookie string and get all individual name=value pairs in an array
    var cookieArr = document.cookie.split(";");

    // Loop through the array elements
    for(var i = 0; i < cookieArr.length; i++) {
        var cookiePair = cookieArr[i].split("=");

        /* Removing whitespace at the beginning of the cookie name
        and compare it with the given string */
        if(name == cookiePair[0].trim()) {
            // Decode the cookie value and return
            return decodeURIComponent(cookiePair[1]);
        }
    }

    // Return null if not found
    return null;
}
var qOptions = new Array ();
var numQs = 8;


if (getCookie("q0") == null){
    for (let i = 0; i < numQs; i++){
        qOptions[i] = Math.floor(Math.random() * 2);
        document.cookie = "q" + String(i) + "=" + String(qOptions[i]) + ";";
    }
} else{
    for (let i = 0; i < numQs; i++){
        qOptions[i] = getCookie("q" + String(i));
    }
}

for (let i = 0; i < numQs; i++){
    document.getElementById("q" + String(i)).innerHTML = (questions[i][qOptions[i]]);
    document.getElementById("Q" + String(i+1) + "Type").value = (String(qOptions[i]));
}
```

6.4. 4. Website PHP

```
<?php session_start();

if ($_SESSION["submitted"] == y){
    echo "Response already submitted";
} else {
    $dbh= new mysqli("localhost", "hkolbcou_WP6SP", "xxxxxxxxxxxxxx", "hkolbcou_formResponses"

    // Check connection
    if ($dbh->connect_error) {
        die("Connection failed: " . $dbh->connect_error);
```

```
}

// mysql_select_db ("hkolbcou_phpTest");
$name = $_POST["name"];
$class = $_POST["class"];

$txtQ0 = $_POST["likert"];
$txtQ1 = $_POST['likert0'];
$txtQ2 = $_POST['likert1'];
$txtQ3 = $_POST['likert2'];
$txtQ4 = $_POST['likert3'];
$txtQ5 = $_POST['likert4'];
$txtQ6 = $_POST['likert5'];
$txtQ7 = $_POST['likert6'];
$txtQ8 = $_POST['likert7'];


$txtQ1Type = $_POST['Q1Type'];
$txtQ2Type = $_POST['Q2Type'];
$txtQ3Type = $_POST['Q3Type'];
$txtQ4Type = $_POST['Q4Type'];
$txtQ5Type = $_POST['Q5Type'];
$txtQ6Type = $_POST['Q6Type'];
$txtQ7Type = $_POST['Q7Type'];
$txtQ8Type = $_POST['Q8Type'];

// database insert SQL code
$sql = "INSERT INTO testing2 (Name, Class, Q0, Q0Type, Q1, Q1Type, Q2, Q2Type, Q3, Q3Type,
setcookie("TestCookie", $value);

if ($dbh->query($sql) === TRUE) {
    echo "Response Submitted!";
    $value = 'something from somewhere';
    $_SESSION["submitted"] = "y";
} else {
    echo "Error: " . $sql . "<br>" . $dbh->error;
}
```

```
    $dbh->close();
}
?>
```

6.5. 5. Python script for creating the bar charts

```
import csv
import re
from urllib.response import addclosehook
import plotly.graph_objects as go
import os
import plotly.express as px
import plotly.io as pio
# pio.kaleido.scope.mathjax = None

if not os.path.exists("images"):
    os.mkdir("images")


valueDict = {
    "strong_disagree": -2,
    "disagree": -1,
    "neutral": 0,
    "agree": 1,
    "strong_agree": 2
}

flipDict = {
    "strong_disagree": "strong_agree",
    "disagree": "agree",
    "neutral": "neutral",
    "agree": "disagree",
    "strong_agree": "strong_disagree"
}

questionDict = {
    2: "Do you plan on continuing to study Computer Science beyond your current level (e.g. as
    4: "Learning Computer Science will be useful to me in the future.",
    6: "Studying Computer Science leads good jobs.",
    8: "I am interested in a career involving Computer Science.",
    10: "Computer Science is easy.",
    12: "If I wanted to I could get a job as a programmer.",
    14: "There are lots of different types of jobs you can get by studying Computer Science.",
    16: "Computer Science is interesting.",
    18: "Computer Science is for people like me."
}

responceIndex = {
     "strong_disagree": 0,
     "disagree": 1,
     "neutral": 2,
     "agree": 3,
     "strong_agree": 4
}

def calcScore(studentIndex, questionNum):
    if responce[studentIndex][questionNum + 1] == "0":
        return valueDict[responce[studentIndex][questionNum]]
    else:
        return -valueDict[responce[studentIndex][questionNum]]

def correctFlip(studentIndex, questionNum):
    if responce[studentIndex][questionNum + 1] == "0":
        return responce[studentIndex][questionNum]
    else:
        return flipDict[responce[studentIndex][questionNum]]

def addClassToLists89(tutorGroup):
    with open((tutorGroup + ".csv"), newline='') as tutorCSV:
        reader = csv.reader(tutorCSV, delimiter=',', quotechar='|')
        for row in reader:
            name = row[0][1:(row[0].find(" "))]
            if name != "Learner":
                names.append(name.lower())
                gender.append(row[1][1:2])
                ethnicity.append(row[2][1:-1])
                pupilPremium.append(row[3][1:-1])
                fsm.append(row[4][1:-1])
                sen.append(row[5][1:-1])
                eal.append(row[12][1:-1])
                cap.append(row[27][1:-1])
                csTargets.append(row[30][1:-1])

def addClassToLists7(tutorGroup):
    with open((tutorGroup + ".csv"), newline='') as tutorCSV:
        reader = csv.reader(tutorCSV, delimiter=',', quotechar='|')
        for row in reader:
            name = row[0][1:(row[0].find(" "))]
            if name != "Learner":
                names.append(name.lower())
                gender.append(row[1][1:2])
                ethnicity.append(row[2][1:-1])
                pupilPremium.append(row[3][1:-1])
                fsm.append(row[4][1:-1])
                sen.append(row[5][1:-1])
                eal.append(row[12][1:-1])
                cap.append(row[24][1:-1])
                csTargets.append(row[27][1:-1])

def addClassToLists11(tutorGroup):
    with open((tutorGroup + ".csv"), newline='') as tutorCSV:
        reader = csv.reader(tutorCSV, delimiter=',', quotechar='|')
        for row in reader:
            name = row[0][1:(row[0].find(" "))]
            if name != "Learner": # Column heading
                names.append(name.lower())
                gender.append(row[1][1:2])
                ethnicity.append(row[2][1:-1])
                pupilPremium.append(row[3][1:-1])
                fsm.append(row[4][1:-1])
                sen.append(row[5][1:-1])
                eal.append(row[8][1:-1])
                cap.append(row[15][1:-1])
                csTargets.append(row[12][1:-1])
```

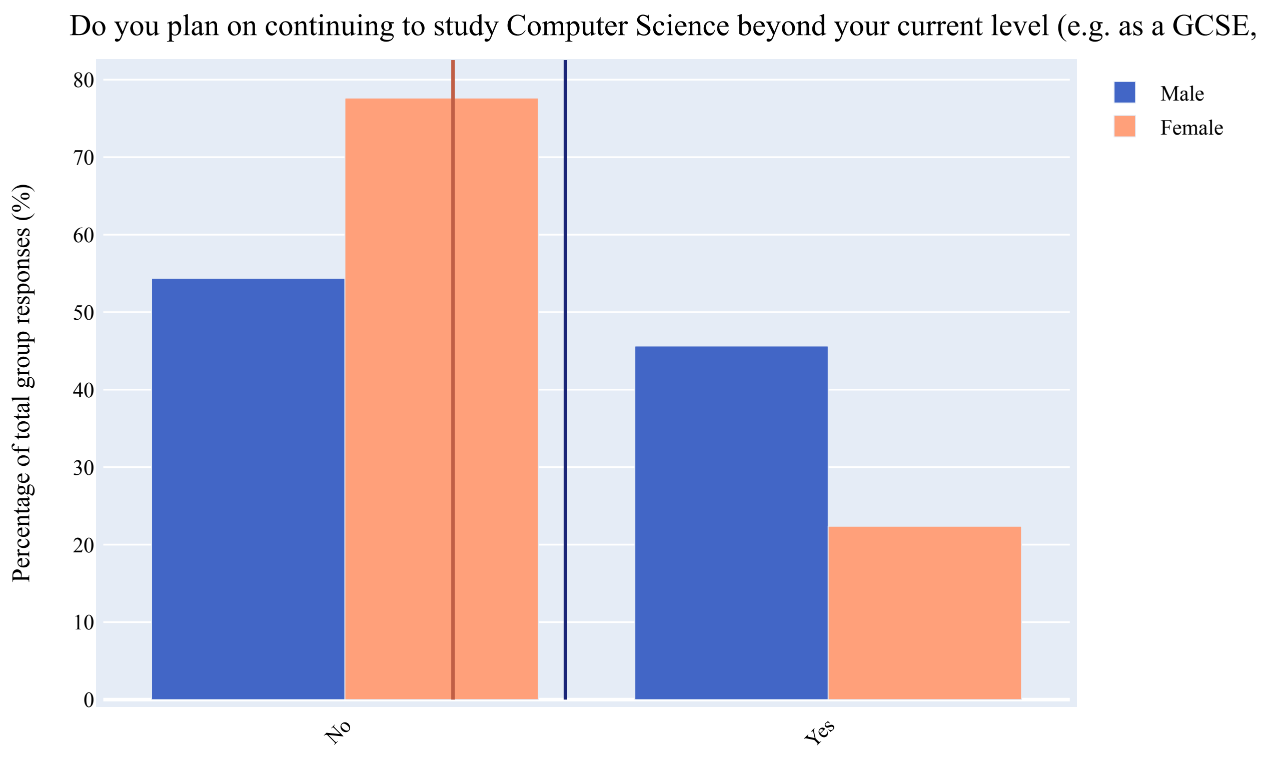
```python
def drawGraph(questionNum, studentAtribute, valToCheck, groupALabel, groupBLabel, fileLabel):
    q1FrequenciesA = [0,0,0,0,0]
    q1FrequenciesB = [0,0,0,0,0]
    q1FrequenciesA0 = [0,0,0,0,0]
    q1FrequenciesB0 = [0,0,0,0,0]
    q1FrequenciesA1 = [0,0,0,0,0]
    q1FrequenciesB1 = [0,0,0,0,0]

    for i in range(len(names)):
        if responce[i] != None:
            reponseCorrected = correctFlip(i, questionNum)
            if studentAtribute[i] == valToCheck:
                q1FrequenciesA[responceIndex[reponseCorrected]] += 1
                if responce[i][questionNum + 1] == "0":
                    q1FrequenciesA0[responceIndex[reponseCorrected]] += 1
                else:
                    q1FrequenciesA1[responceIndex[reponseCorrected]] += 1
            else:
                q1FrequenciesB[responceIndex[reponseCorrected]] += 1
                if responce[i][questionNum + 1] == "0":
                    q1FrequenciesB0[responceIndex[reponseCorrected]] += 1
                else:
                    q1FrequenciesB1[responceIndex[reponseCorrected]] += 1

    q1PercentagesA = [freq/sum(q1FrequenciesA)*100 for freq in q1FrequenciesA]
    q1PercentagesB = [freq/sum(q1FrequenciesB)*100 for freq in q1FrequenciesB]
    q1PercentagesA0 = [freq/(sum(q1FrequenciesA0))*100 for freq in q1FrequenciesA0]
    q1PercentagesB0 = [freq/(sum(q1FrequenciesB0))*100 for freq in q1FrequenciesB0]
    q1PercentagesA1 = [freq/(sum(q1FrequenciesA1))*100 for freq in q1FrequenciesA1]
    q1PercentagesB1 = [freq/(sum(q1FrequenciesB1))*100 for freq in q1FrequenciesB1]

    xAxis = ["Strongly Disagree", 'Disagree', 'Neutral', "Agree", "Strongly Agree"]

    opacity = 0.7
    fig = go.Figure()
    fig.add_trace(go.Bar(
        x=xAxis,
        y=q1PercentagesA,
        name=groupALabel,
        marker_color='#4266c6'
    ))
    fig.add_trace(go.Bar(
        x=xAxis,
        y=q1PercentagesB,
        name=groupBLabel,
        marker_color='lightsalmon'
    ))
    # fig.add_trace(go.Bar(
    #     x=xAxis,
    #     y=q1PercentagesB0,
    #     name=groupBLabel,
    #     marker_color='lightsalmon',
    #     opacity=opacity,
    #     offset=0.15,
    #     width=0.3
    # ))
    # fig.add_trace(go.Bar(
    #     x=xAxis,
    #     y=q1PercentagesB1,
    #     name=groupBLabel,
    #     marker_color='#ff2248',
    #     opacity=opacity,
    #     offset=0.15,
    #     width=0.3
    # ))

    def sumResponses(freqList):
        weightedSum = -2*freqList[0] - freqList[1] + freqList[3] + 2*freqList[4]
        return round(weightedSum / (sum(freqList)), 2)

    averagesText = f"{groupALabel} Average Response = {sumResponses(q1FrequenciesA)}<br>{group
        <br>Absolute difference between averages = {round(abs(sumResponses(q1FrequenciesA) - su

    fig.update_layout(barmode='group', xaxis_tickangle=-45, title=questionDict[questionNum], fo
            family="Computer Modern",
            size=12,
            color="Black"
        ),  xaxis_title="", yaxis_title="Percentage of total group responses (%)")
    fig.add_annotation(x = 2, y = -0.26, text = averagesText,
                     yref = 'paper', showarrow = False,
                     font=dict(color="Black",size=12))
    # fig.add_annotation(x = -0.1, y = 0.5, text = "test", yref = 'paper',
    #                    xref = 'paper', showarrow = False,
    #                    font=dict(color="Black",size=12))
    # fig.add_vrect(x0=0.9, x1=2, annotation_text="vertical rectangle",
    # annotation_position="top left", annotation_textangle = -90)

    fig.add_shape(type = 'line', x0 = sumResponses(q1FrequenciesA) + 2, x1 = sumResponses(q1Fre
              line = dict(color = "#1a2577"))
    fig.add_shape(type = 'line', x0 = sumResponses(q1FrequenciesB) + 2, x1 = sumResponses(q1Fre
              line = dict(color = "#c05e45"))
    fig.update_layout(margin=dict(l=75,
                                r=25,
                                b=110,
                                t=35,
                                pad=4),
              paper_bgcolor="White", barmode='group')

    # fig.show()
    fig.write_image(f"images2/{fileLabel}_Q{str(questionNum)}.pdf")

names = []
gender = []
ethnicity = []
pupilPremium = []
fsm = []
sen = []
eal = []
cap = []
csTargets = []

addClassToLists7("7AAM")
addClassToLists7("7CDO")
```

```
addClassToLists89("8ALY")
addClassToLists89("8ST")
addClassToLists89("8SEV")
addClassToLists89("9ANB")
addClassToLists89("9MAA")
addClassToLists89("9OFL")

addClassToLists11("11A")
addClassToLists11("10B")


responce = [None] * len(names)
with open('testing2.csv', newline='') as responsesCSV:
    reader = csv.reader(responsesCSV, delimiter=',', quotechar='|')
    for row in reader:
        for name in names:
            if name[0:].lower() in row[0].lower():
                responce[names.index(name[0:].lower())] = [rowEntry.replace("\"", "") for rowEn

for i in range(4,19, 2):
    drawGraph(i, gender, "M", "Male", "Female", "gender")
    drawGraph(i, eal, "Y", "EAL", "Non EAL", "EAL")
    drawGraph(i, fsm, "Y", "FSM", "Non FSM", "FSM")
    drawGraph(i, sen, "", "Non SEN", "SEN", "SEN")
```

6.6. 6. Quiz

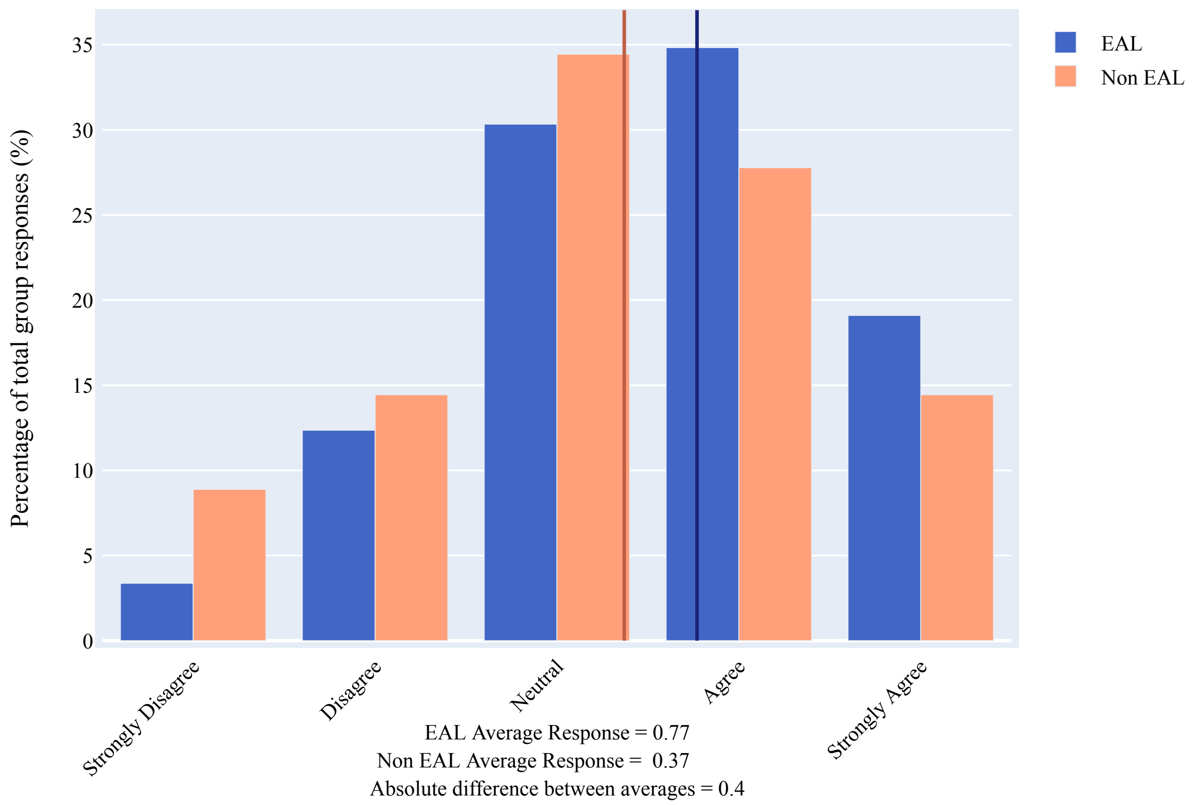
https://forms.office.com/Pages/ShareFormPage.aspx?id=YG418z6cUUOvrLY9SldgyY9vrtwLKGRBq6
KjUdAOAmBUMTJOWTRGSTIyRlNXMTI0NEdZN1hPSExQOC4u&sharetoken=lD2llKh9LrrEjl4F0OiD

6.7. 7. Graphs



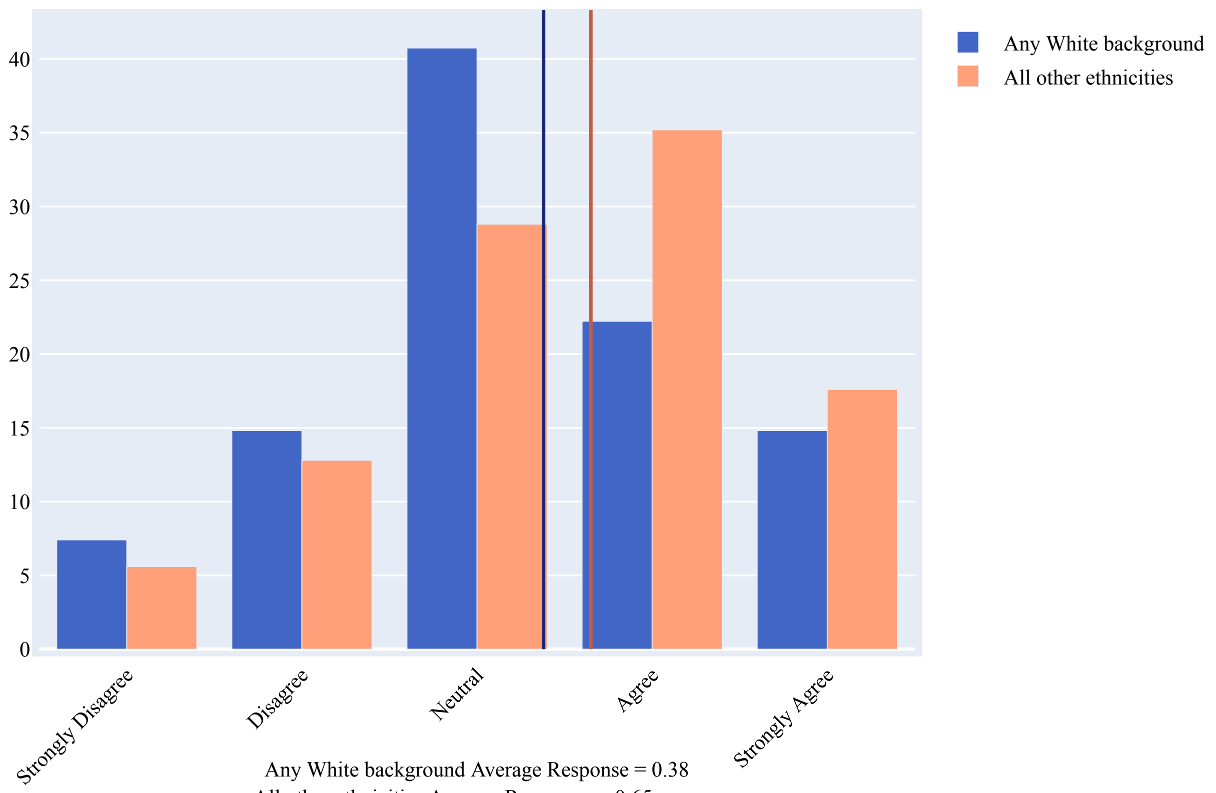
Male Average Response = 0.46
Female Average Response = 0.22
Absolute difference between averages = 0.23

Learning Computer Science will be useful to me in the future.



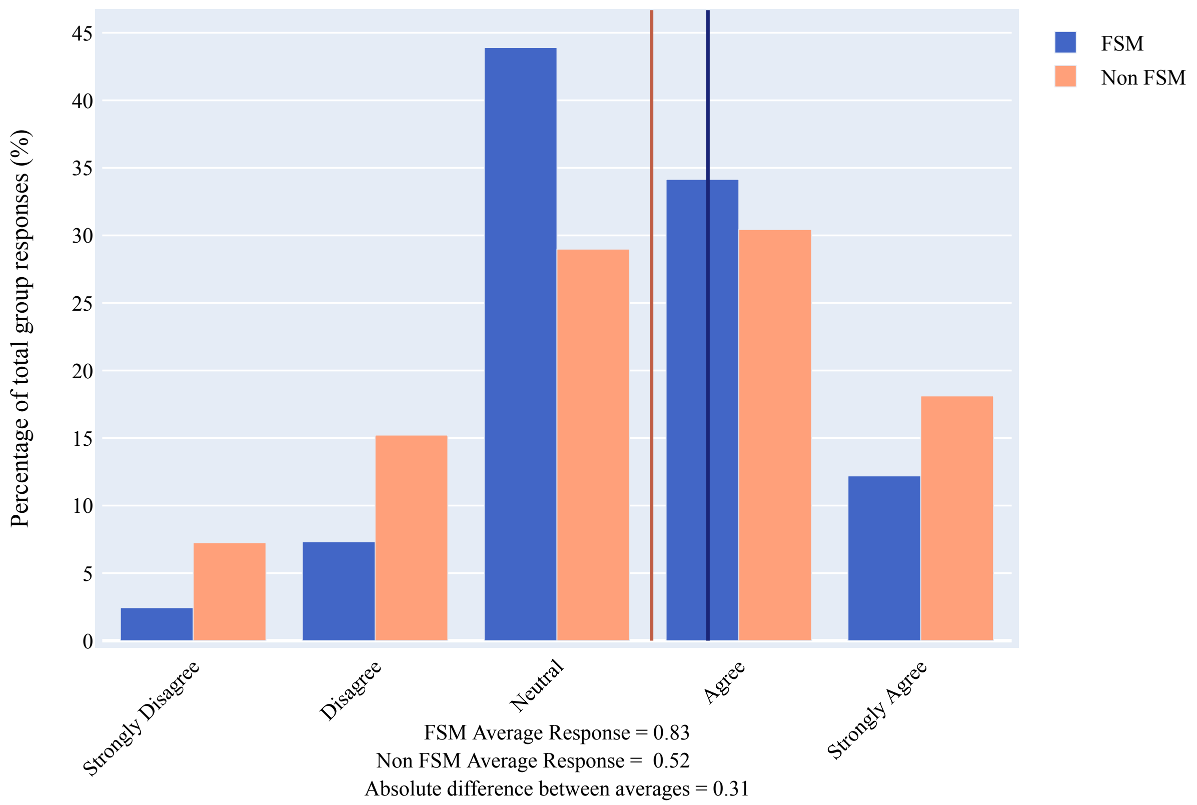
EAL Average Response = 0.77
Non EAL Average Response = 0.37
Absolute difference between averages = 0.4

Learning Computer Science will be useful to me in the future.



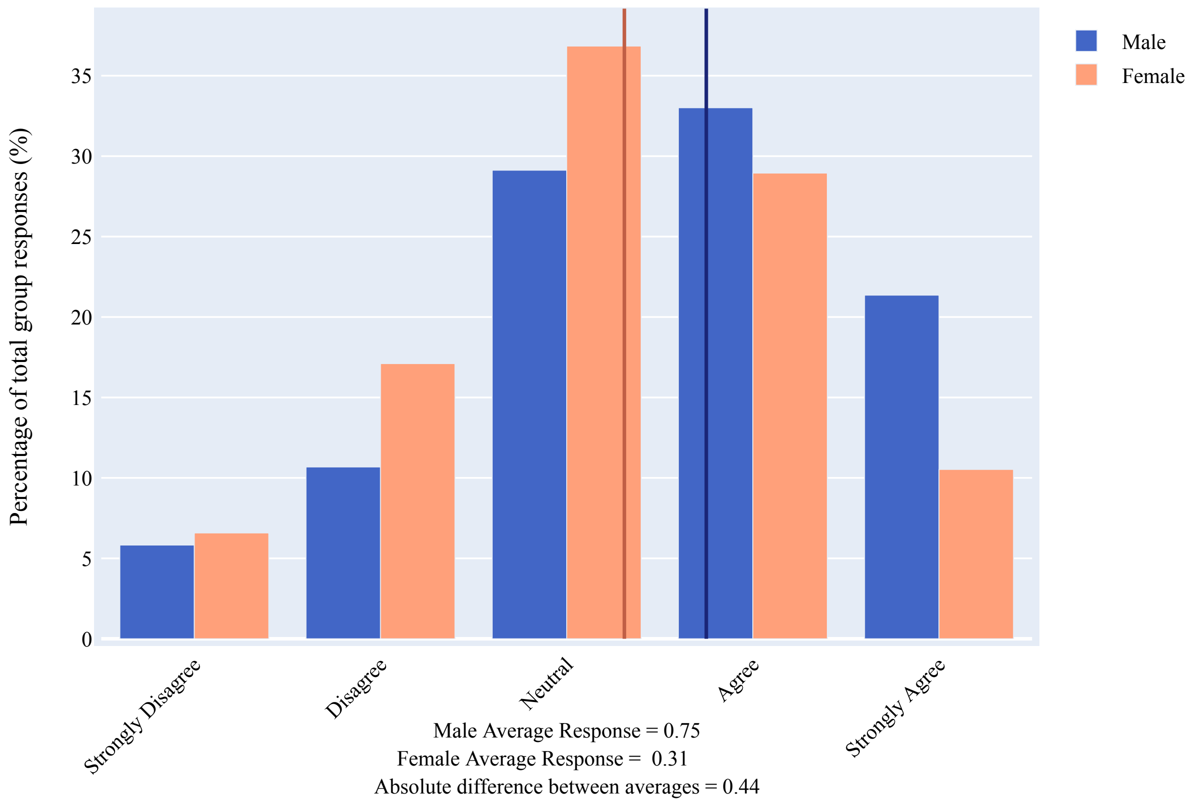
Any White background Average Response = 0.38
All other ethnicities Average Response = 0.65
Absolute difference between averages = 0.27

Learning Computer Science will be useful to me in the future.



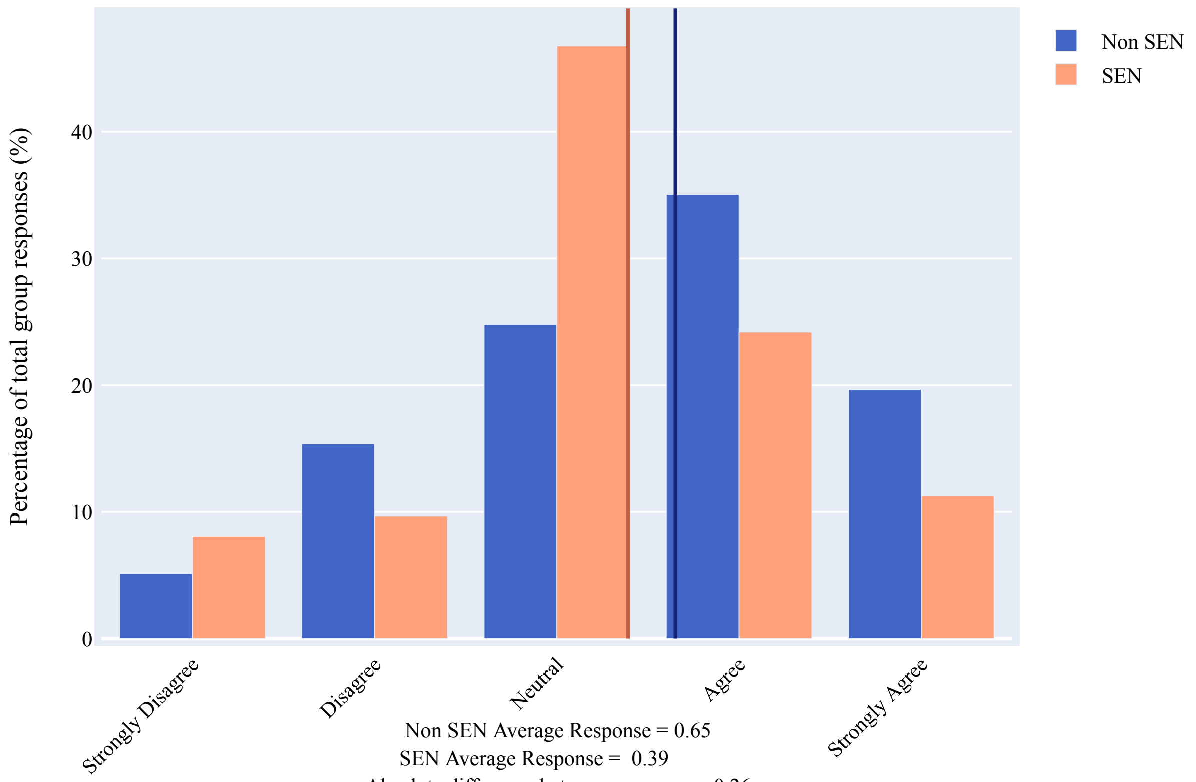
FSM Average Response = 0.83
Non FSM Average Response = 0.52
Absolute difference between averages = 0.31

Learning Computer Science will be useful to me in the future.

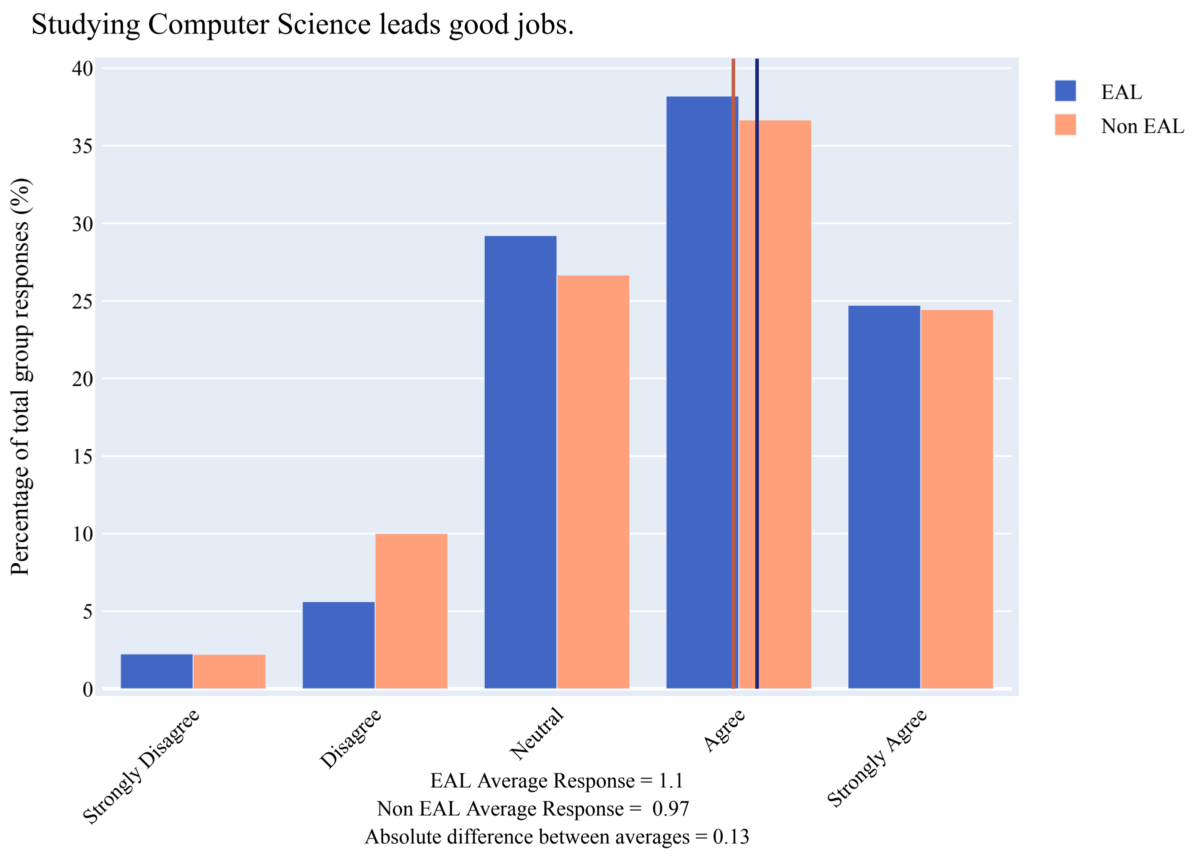


Male Average Response = 0.75
Female Average Response = 0.31
Absolute difference between averages = 0.44

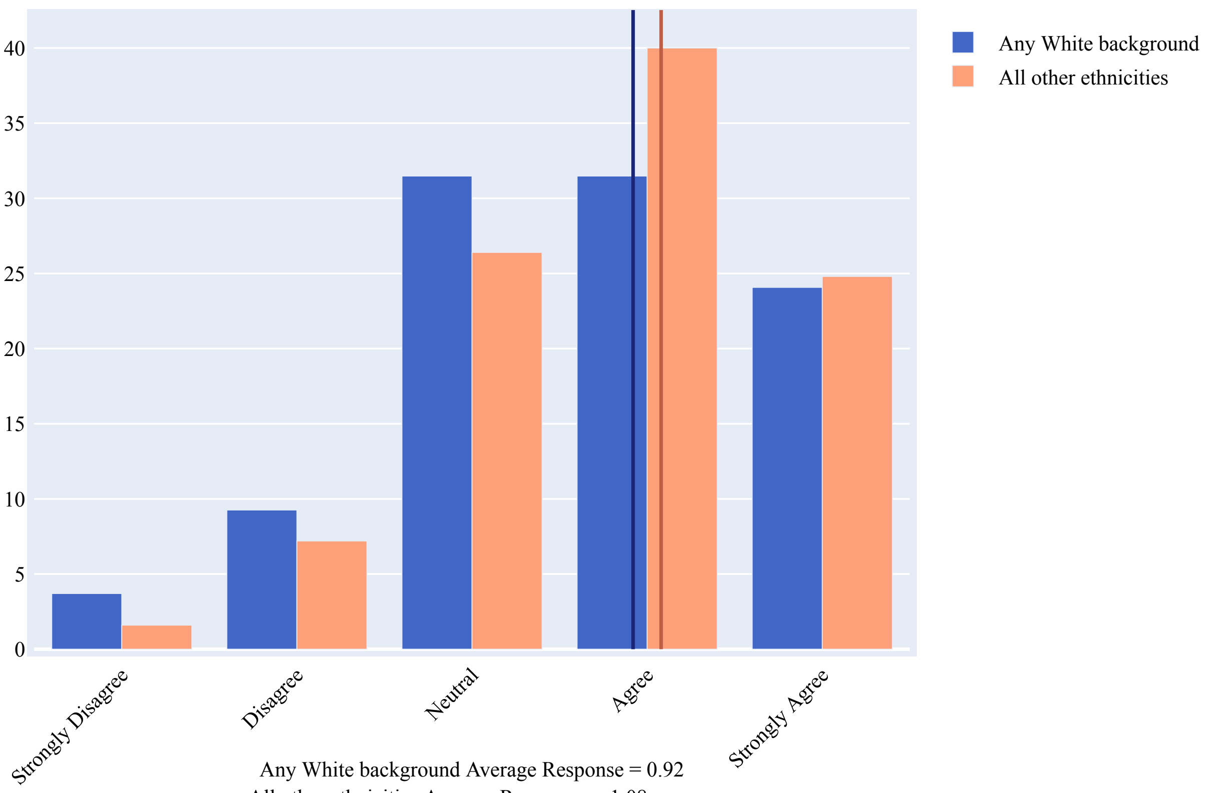
Learning Computer Science will be useful to me in the future.



Non SEN Average Response = 0.65
SEN Average Response = 0.39
Absolute difference between averages = 0.26

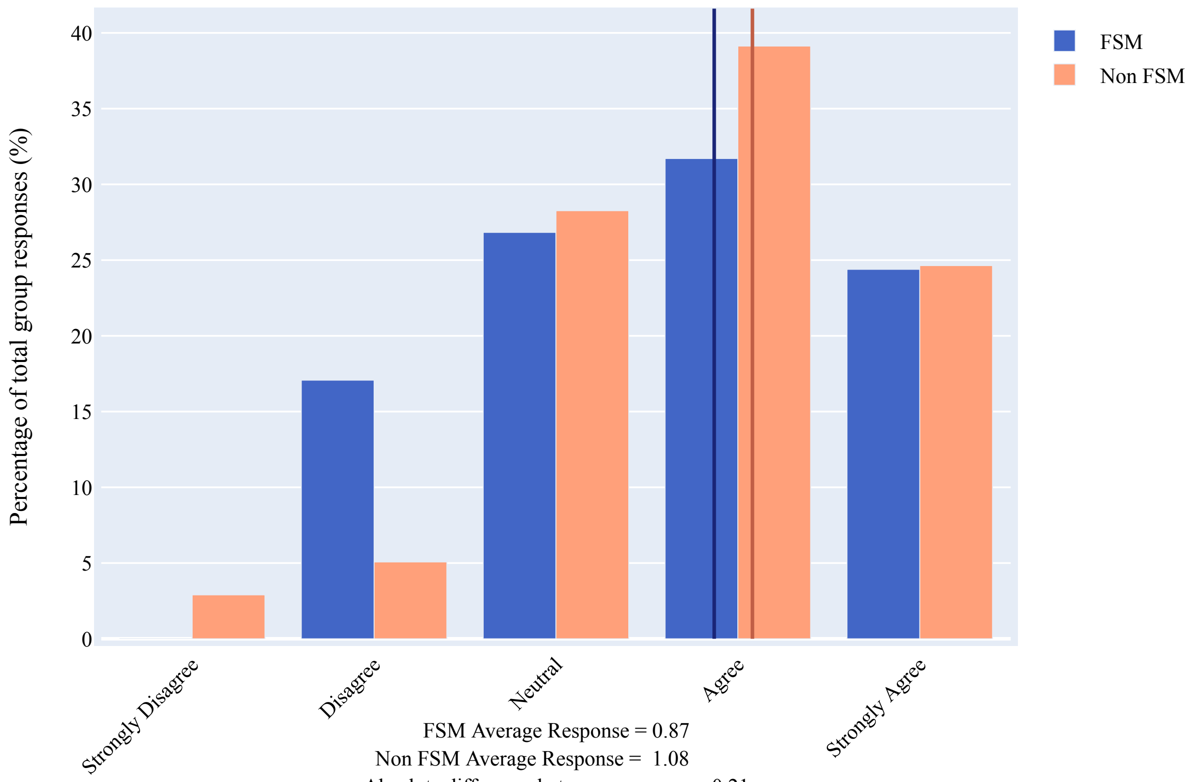

Studying Computer Science leads good jobs.
Percentage of total group responses (%)
40
35
30
25
20
15
10
5
0
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
EAL
Non EAL
EAL Average Response = 1.1
Non EAL Average Response = 0.97
Absolute difference between averages = 0.13

Studying Computer Science leads good jobs.



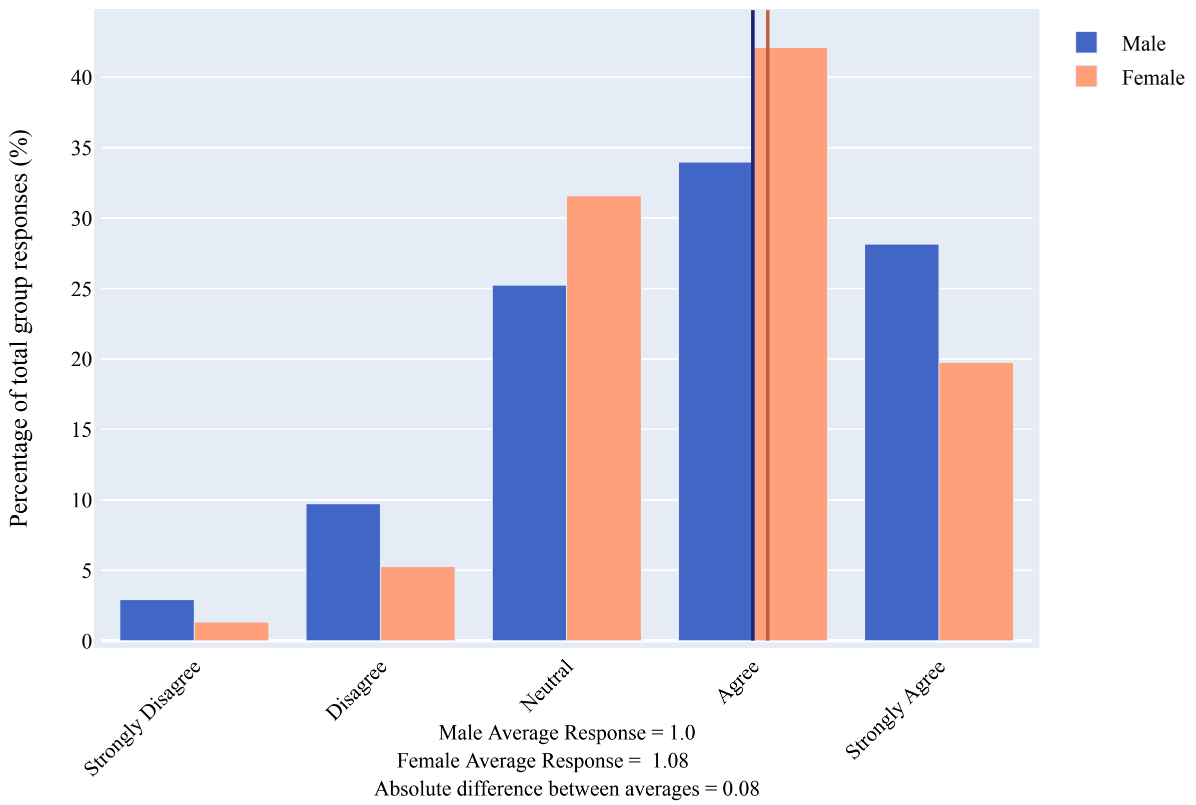
Any White background Average Response = 0.92
All other ethnicities Average Response = 1.08
Absolute difference between averages = 0.16

Studying Computer Science leads good jobs.



FSM Average Response = 0.87
Non FSM Average Response = 1.08
Absolute difference between averages = 0.21

Studying Computer Science leads good jobs.

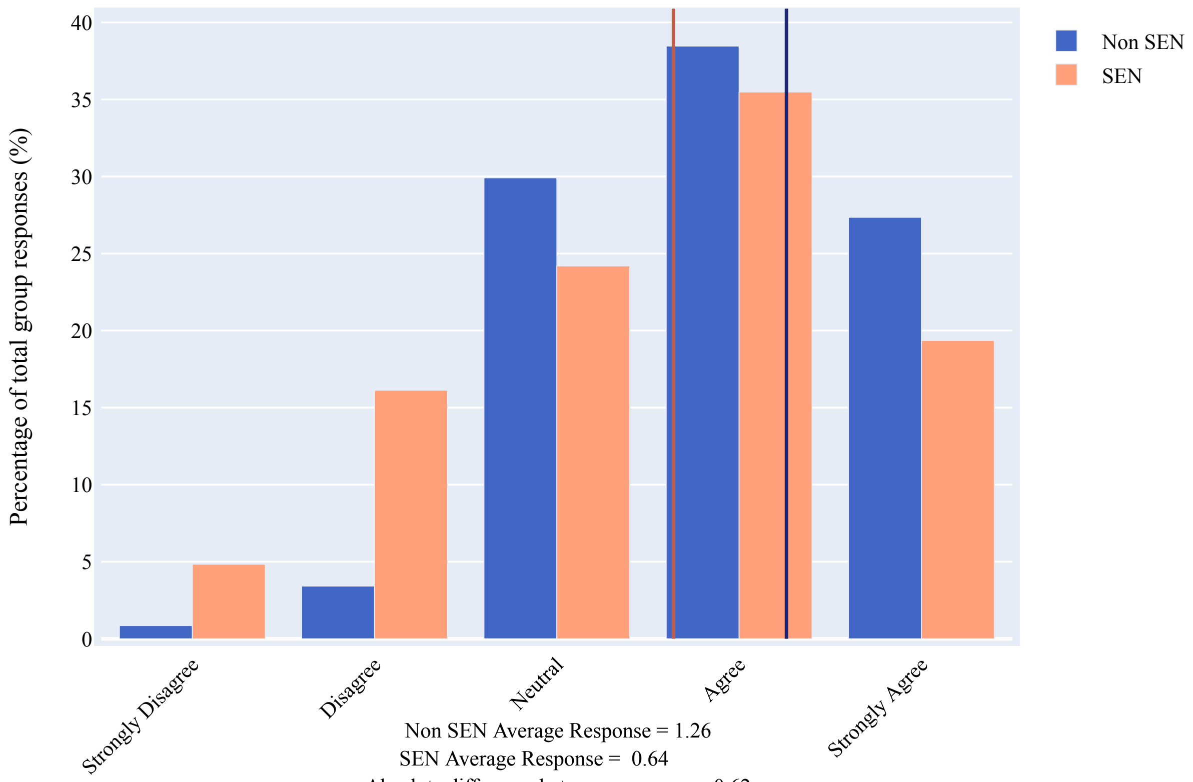


Male Average Response = 1.0

Female Average Response = 1.08

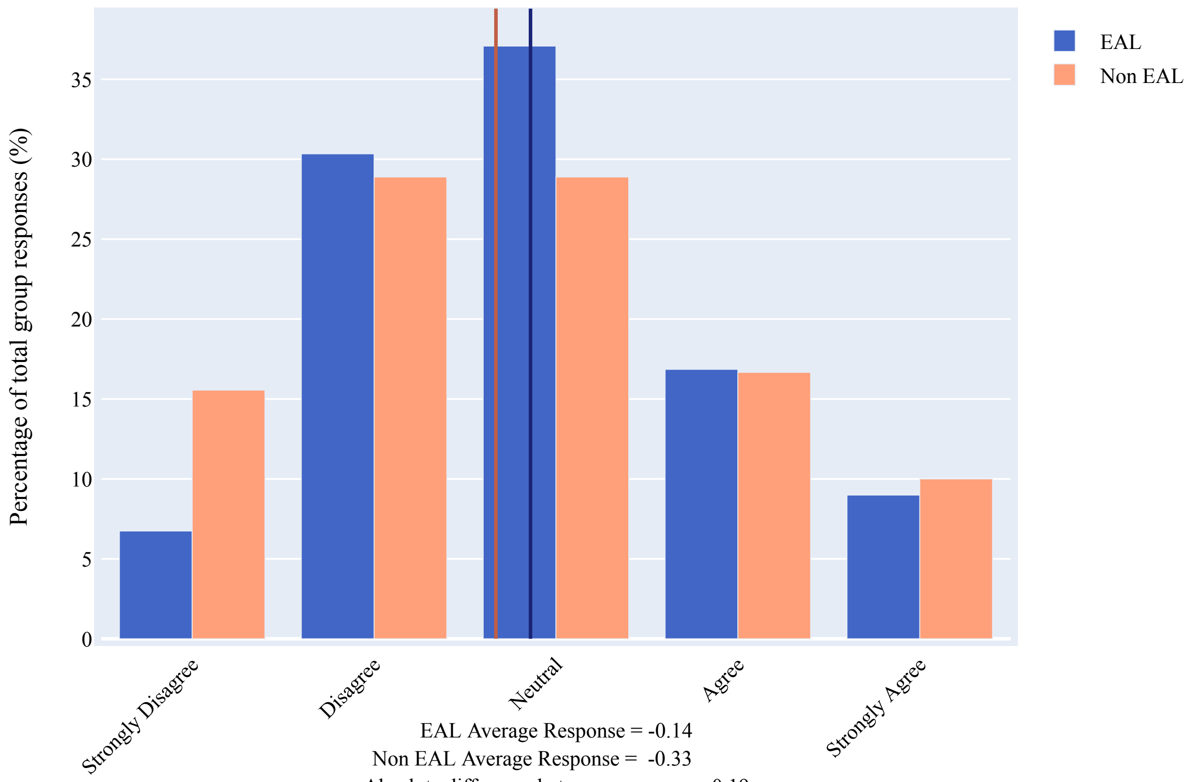
Absolute difference between averages = 0.08

Studying Computer Science leads good jobs.



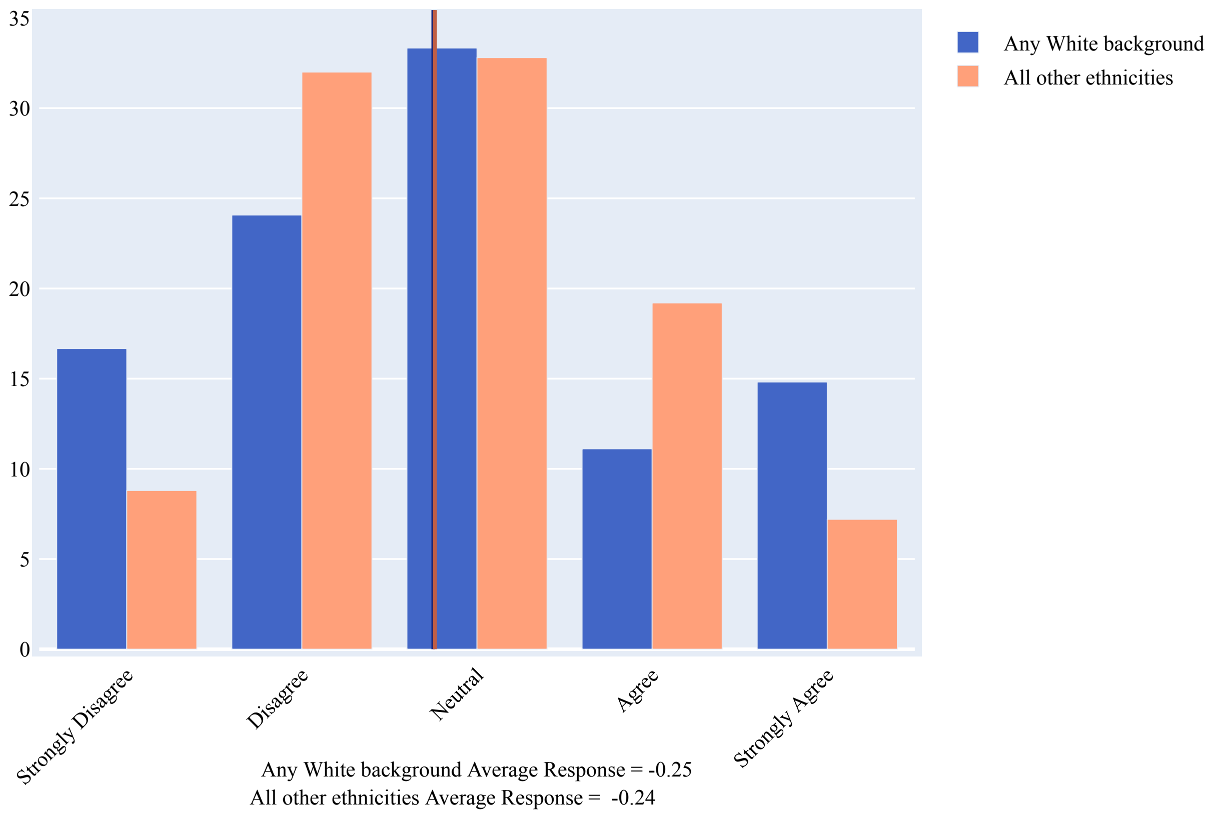
Non SEN Average Response = 1.26
SEN Average Response = 0.64
Absolute difference between averages = 0.62

I am interested in a career involving Computer Science.



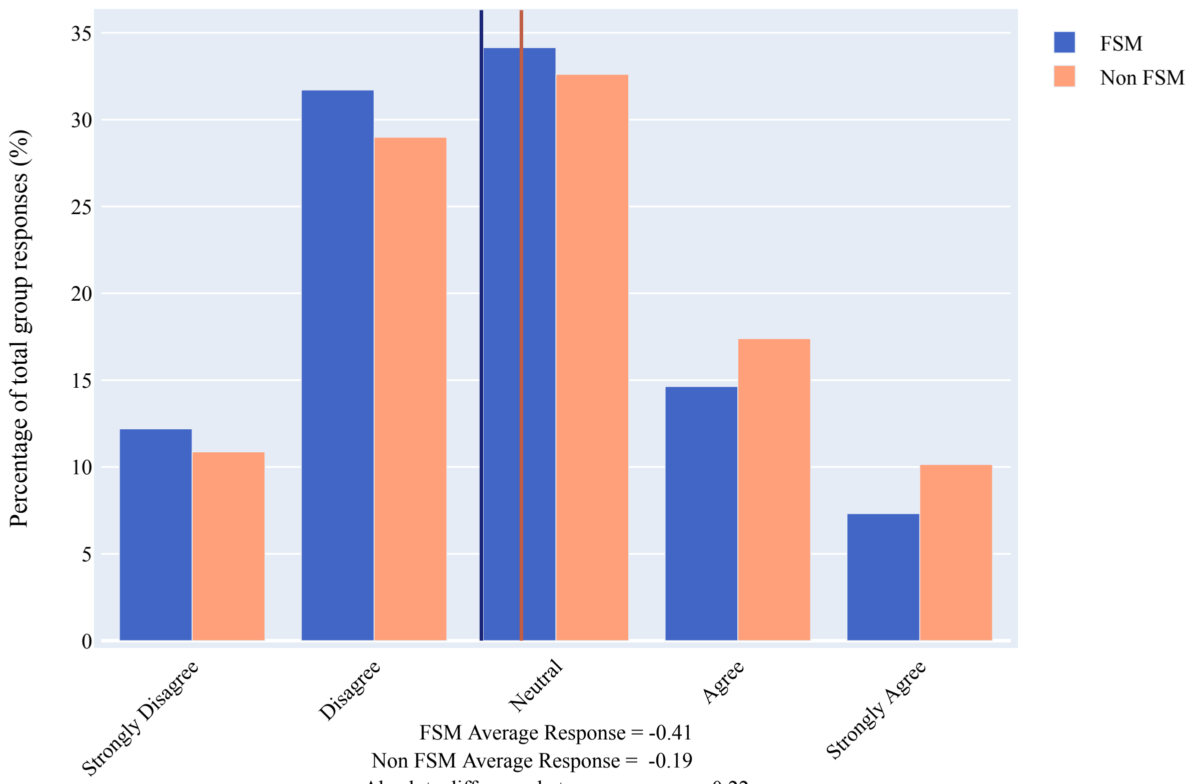
EAL Average Response = -0.14
Non EAL Average Response = -0.33
Absolute difference between averages = 0.19

I am interested in a career involving Computer Science.



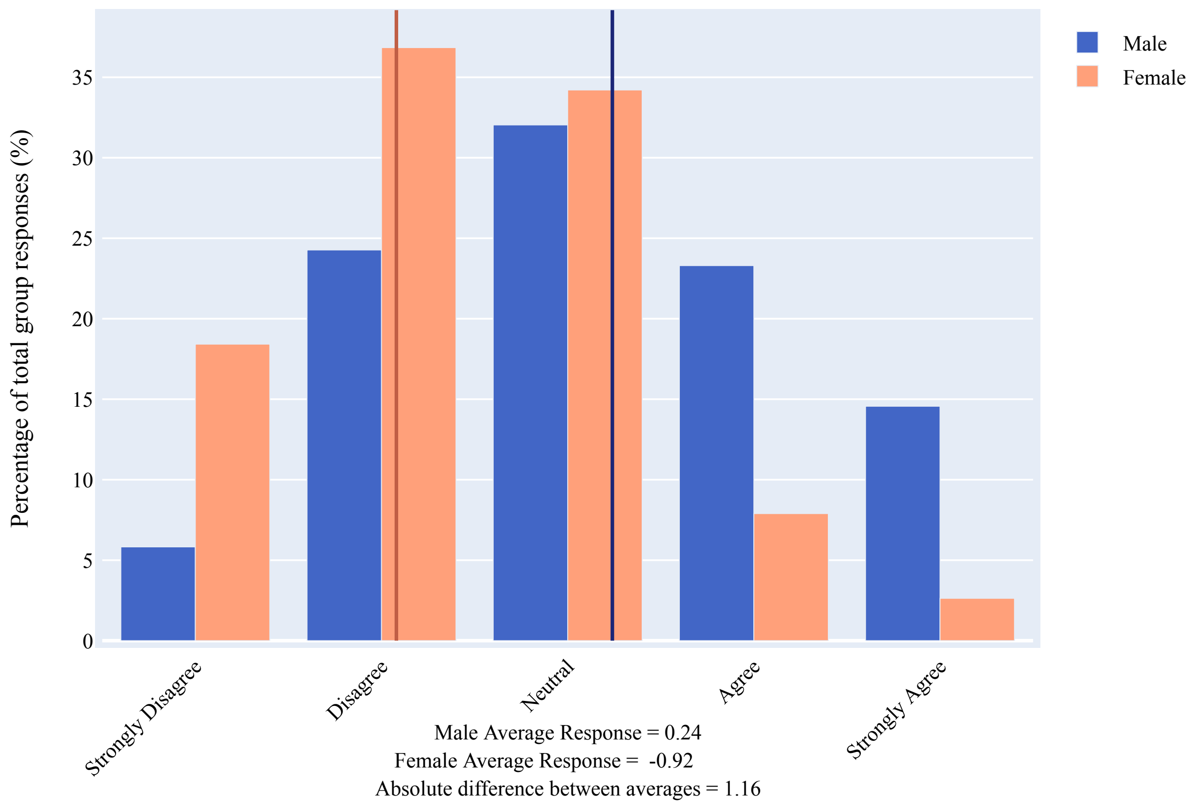
Any White background Average Response = -0.25
All other ethnicities Average Response = -0.24
Absolute difference between averages = 0.01

I am interested in a career involving Computer Science.



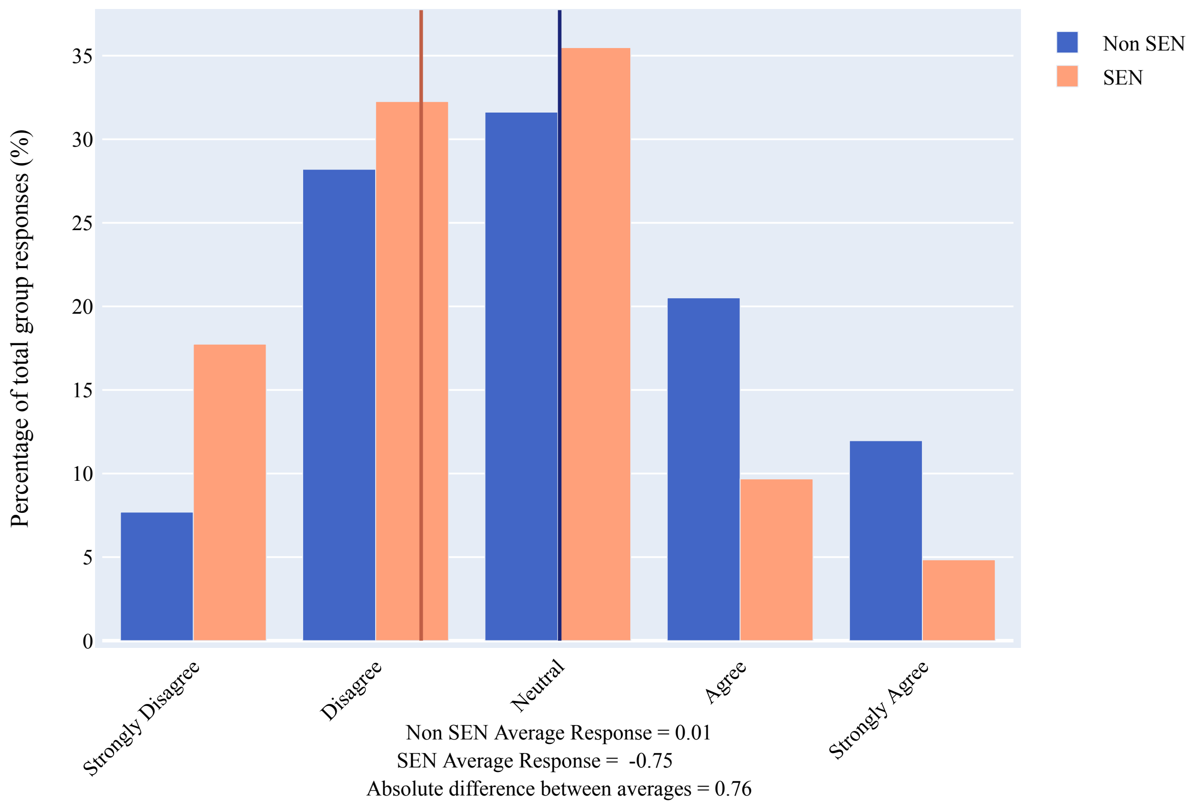
FSM Average Response = -0.41
Non FSM Average Response = -0.19
Absolute difference between averages = 0.22

I am interested in a career involving Computer Science.



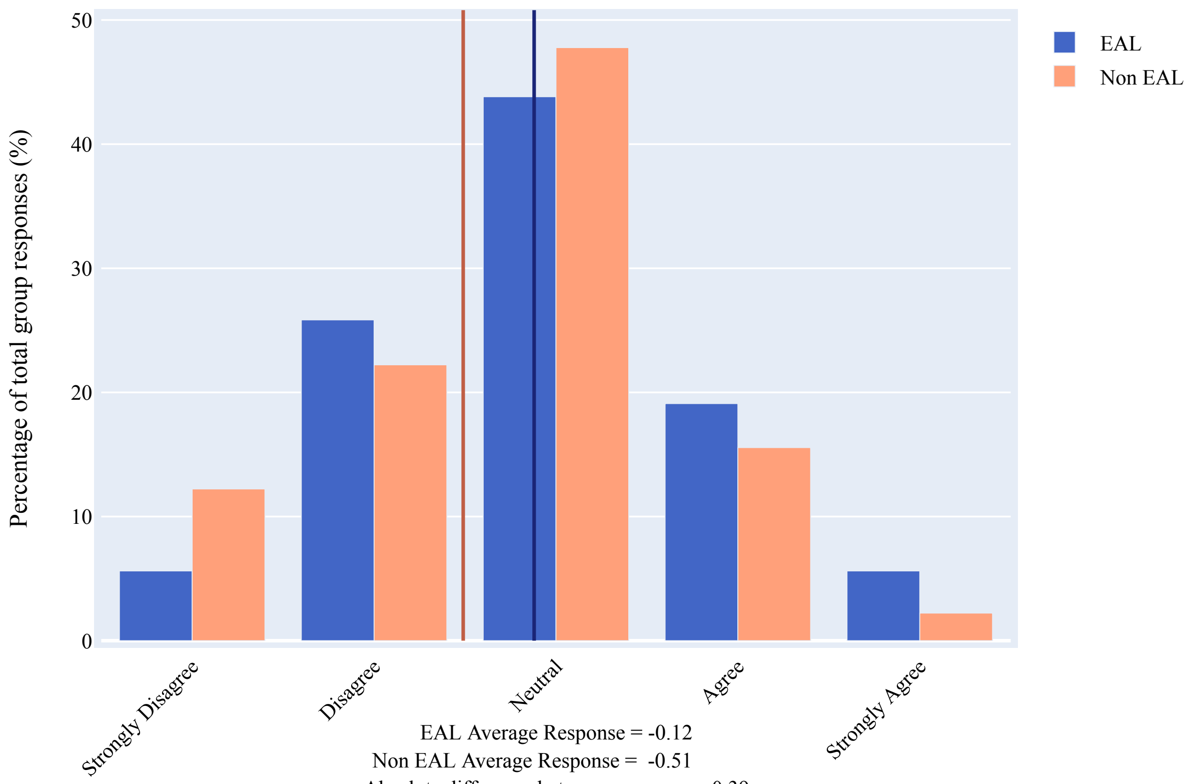
Male Average Response = 0.24
Female Average Response = -0.92
Absolute difference between averages = 1.16

I am interested in a career involving Computer Science.



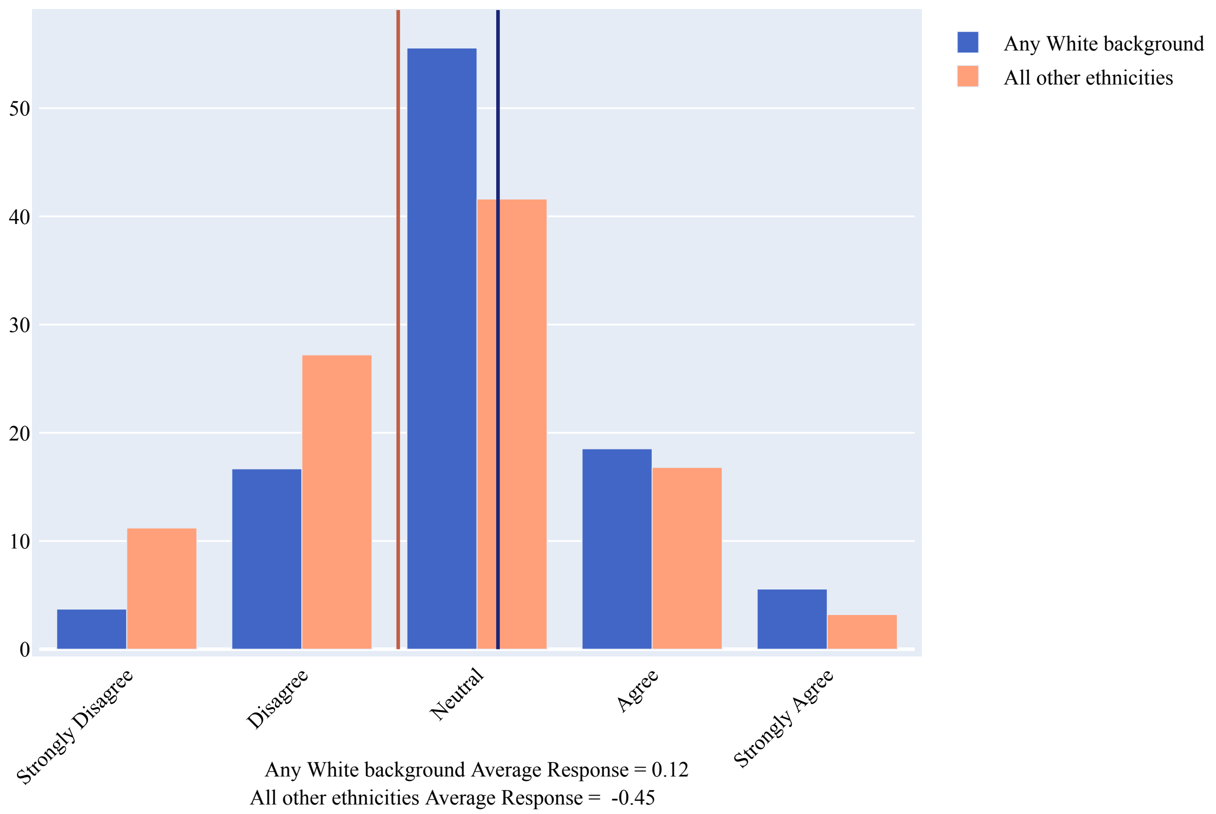
Non SEN Average Response = 0.01
SEN Average Response = -0.75
Absolute difference between averages = 0.76

Computer Science is easy.



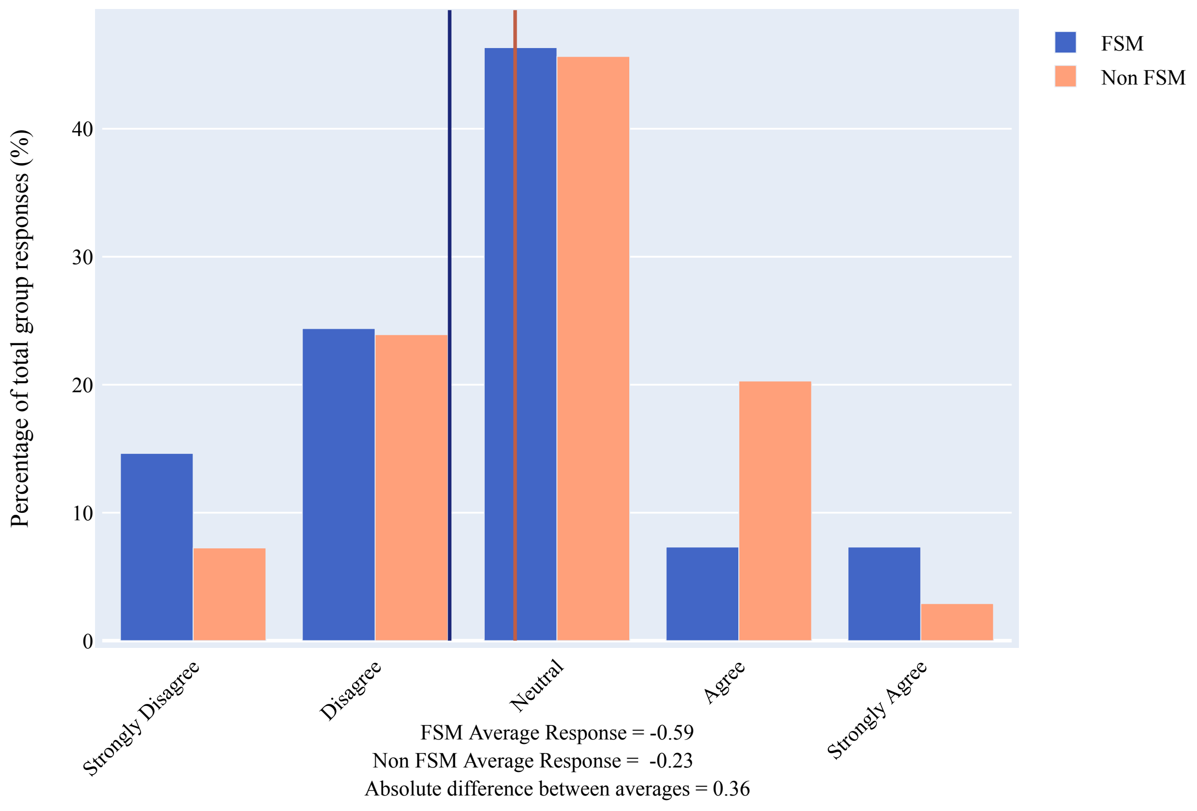
EAL Average Response = -0.12
Non EAL Average Response = -0.51
Absolute difference between averages = 0.39

Computer Science is easy.



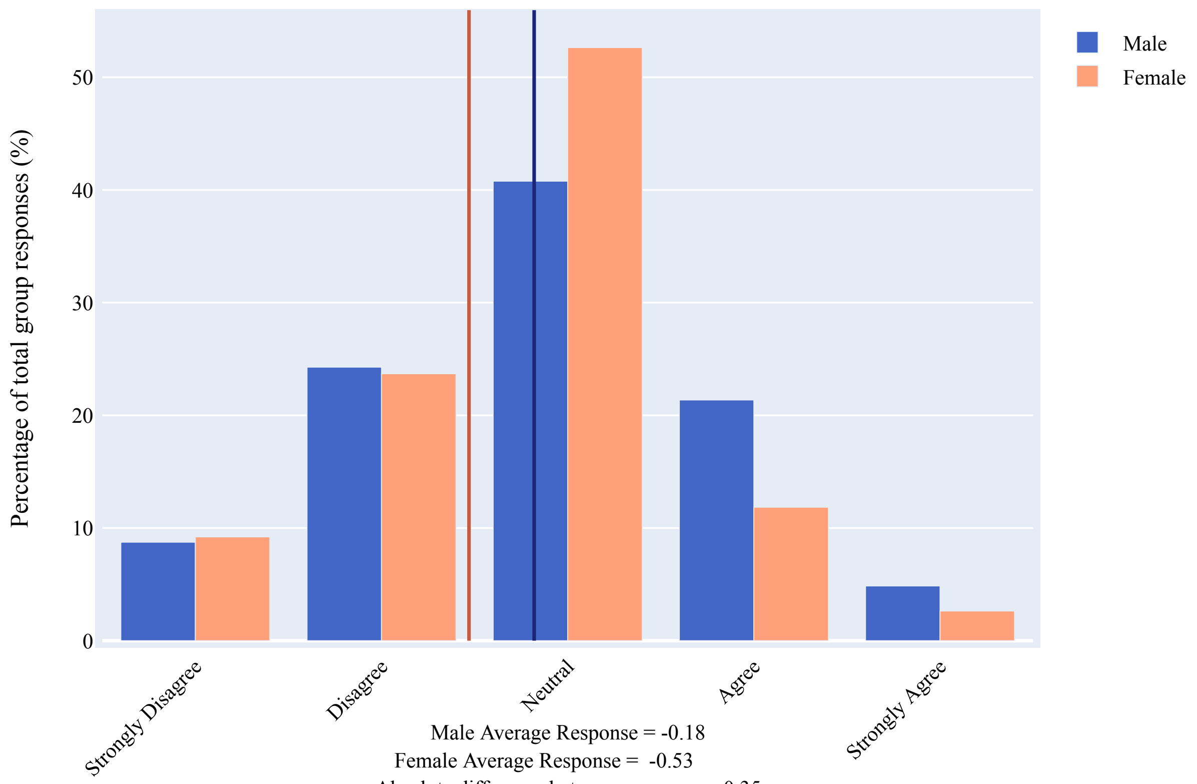
Any White background Average Response = 0.12
All other ethnicities Average Response = -0.45
Absolute difference between averages = 0.57

Computer Science is easy.



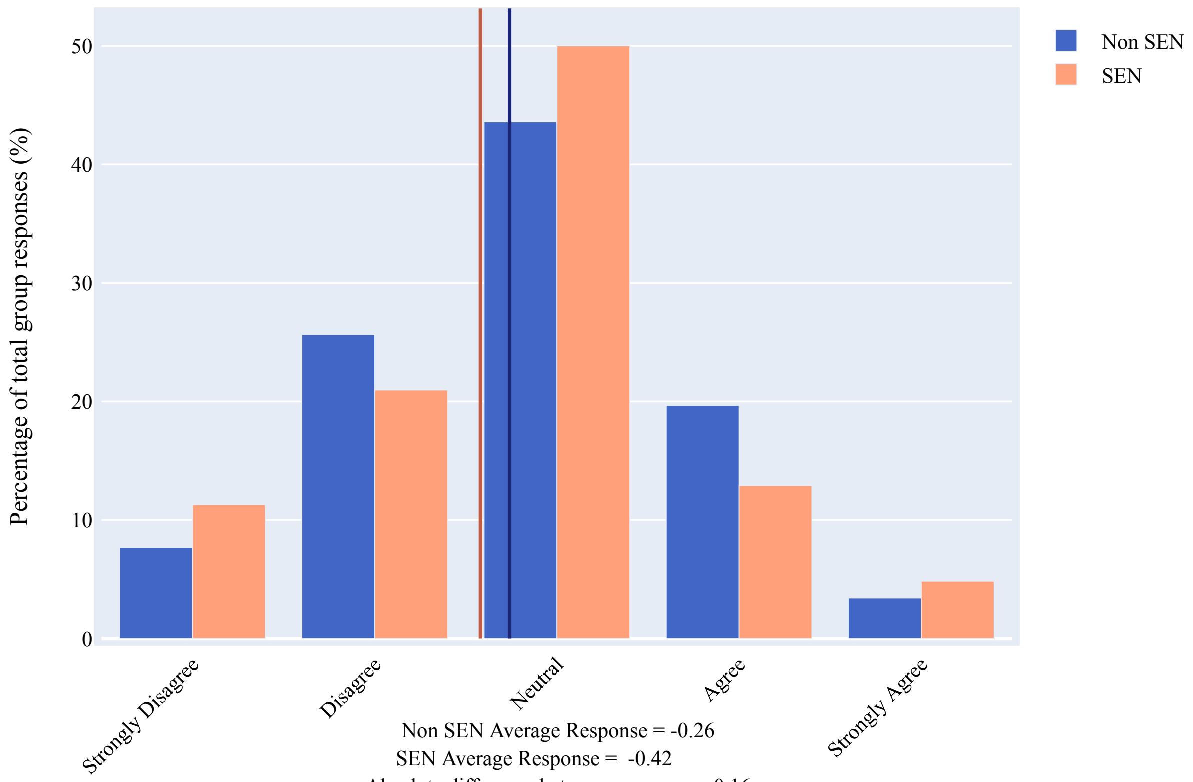
FSM Average Response = -0.59
Non FSM Average Response = -0.23
Absolute difference between averages = 0.36

Computer Science is easy.



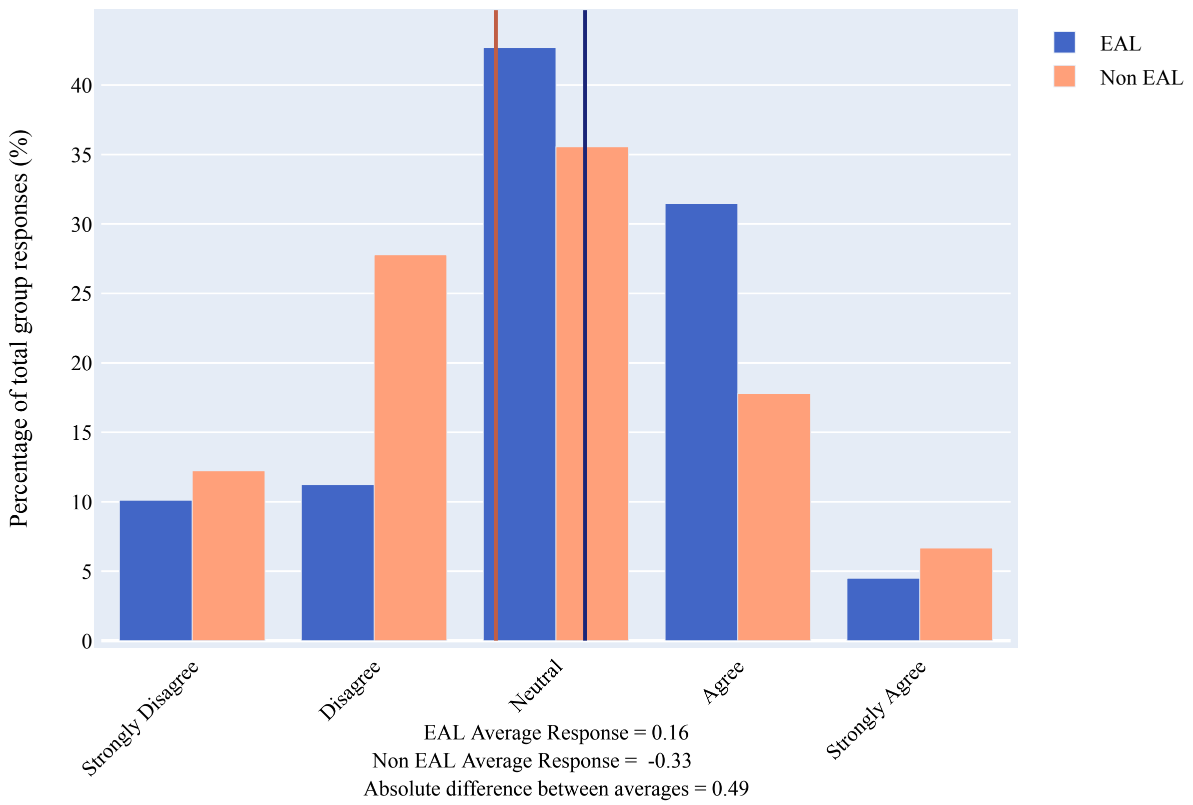
Male Average Response = -0.18
Female Average Response = -0.53
Absolute difference between averages = 0.35

Computer Science is easy.



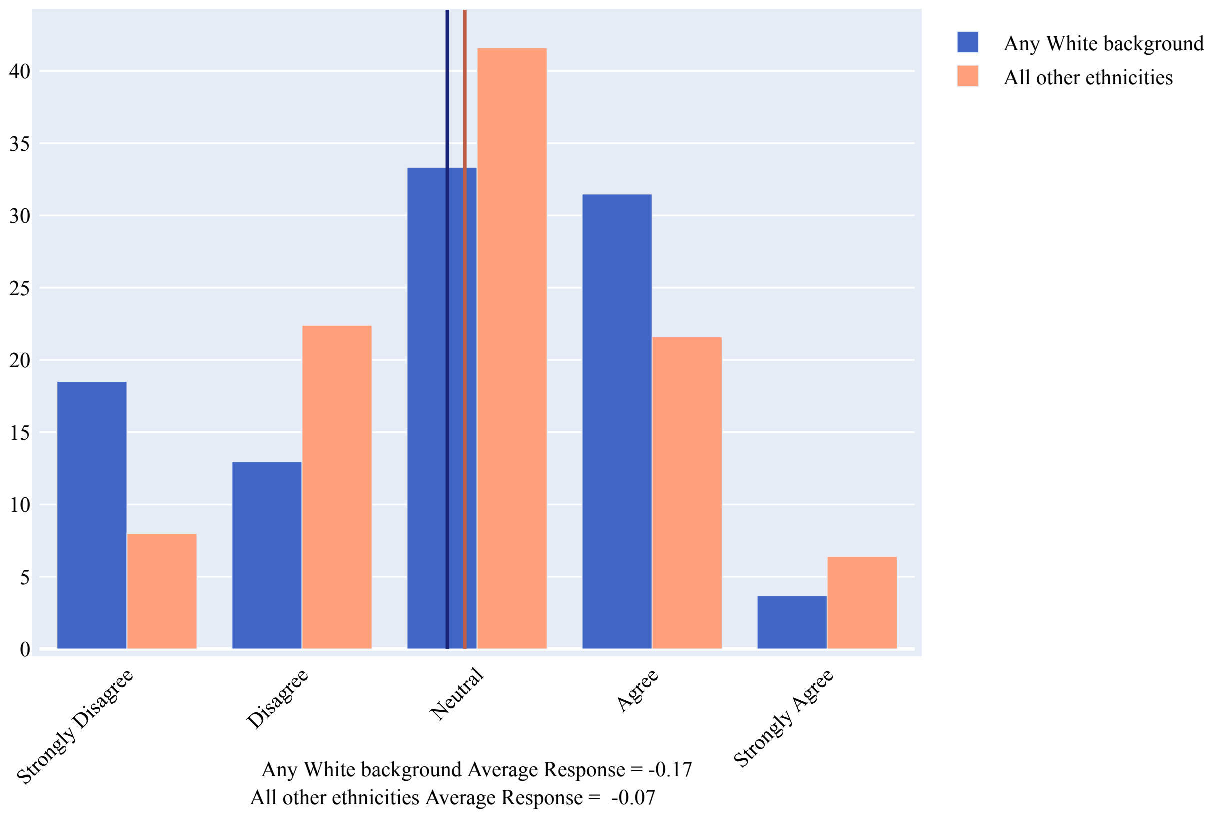
Non SEN Average Response = -0.26
SEN Average Response = -0.42
Absolute difference between averages = 0.16

If I wanted to I could get a job as a programmer.



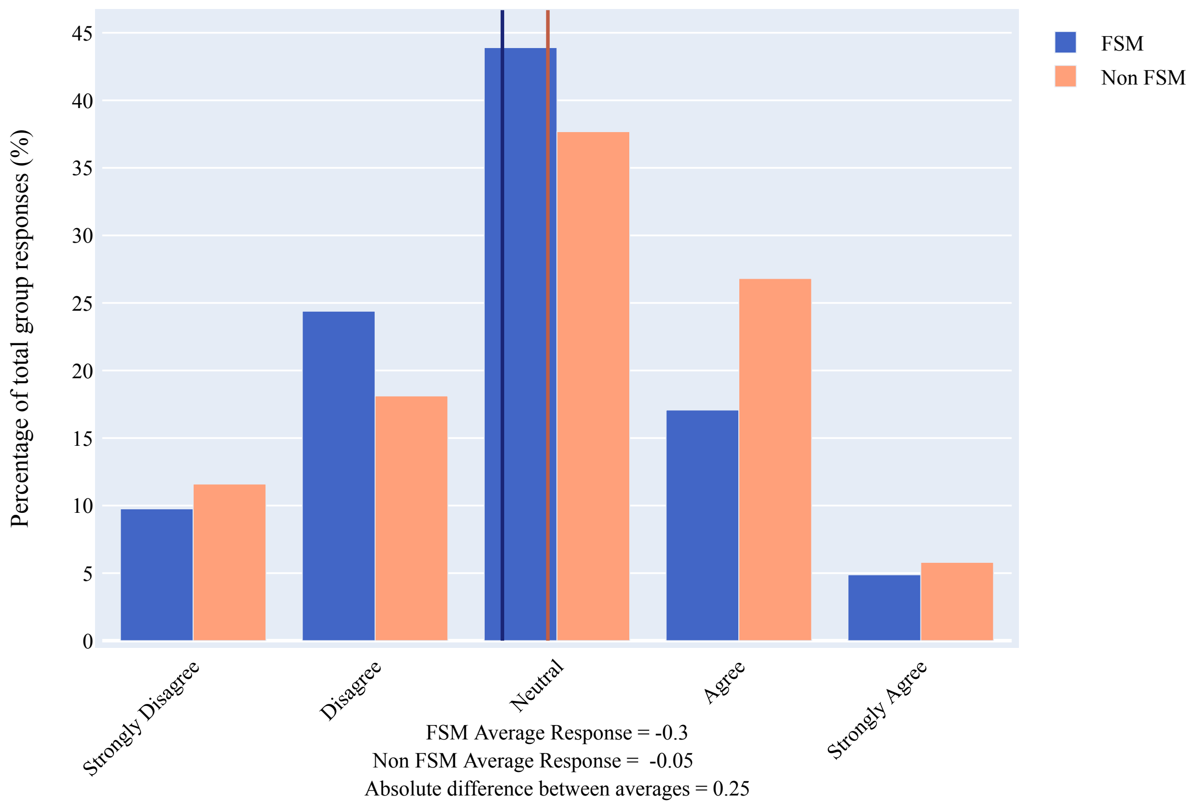
EAL Average Response = 0.16
Non EAL Average Response = -0.33
Absolute difference between averages = 0.49

If I wanted to I could get a job as a programmer.



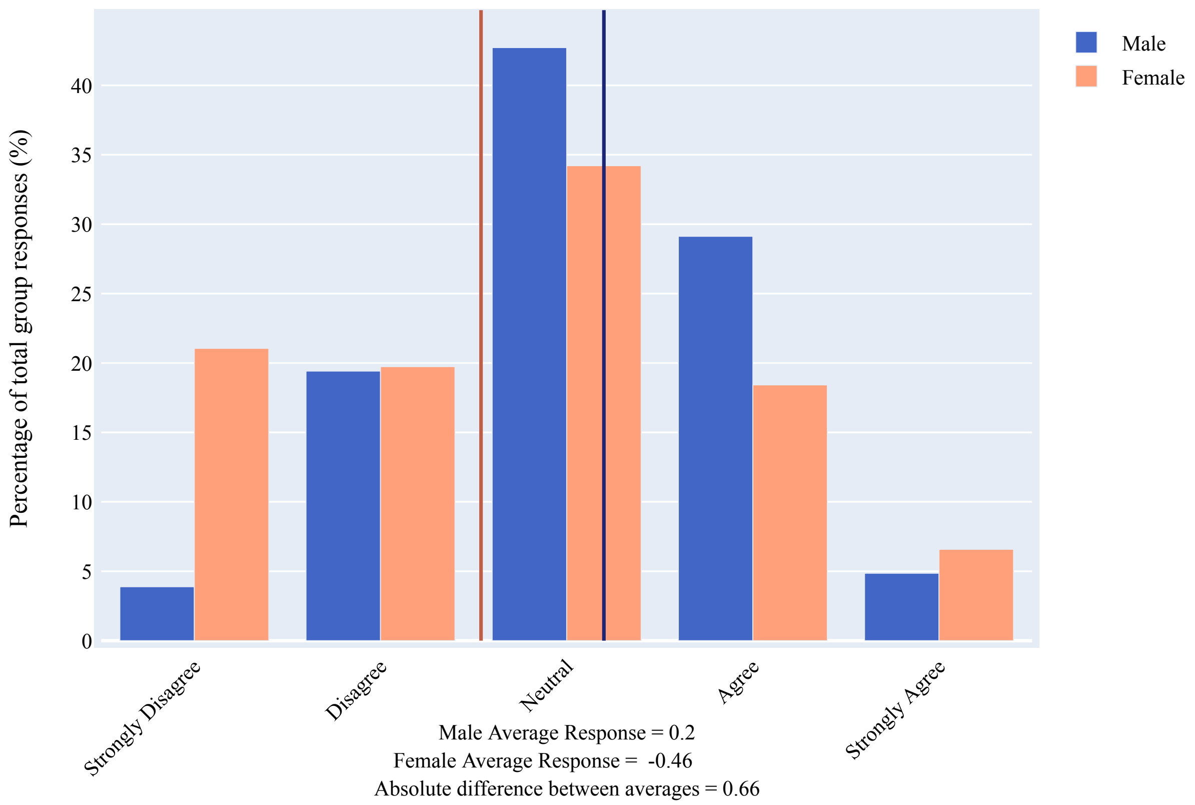
Any White background Average Response = -0.17
All other ethnicities Average Response = -0.07
Absolute difference between averages = 0.1

If I wanted to I could get a job as a programmer.



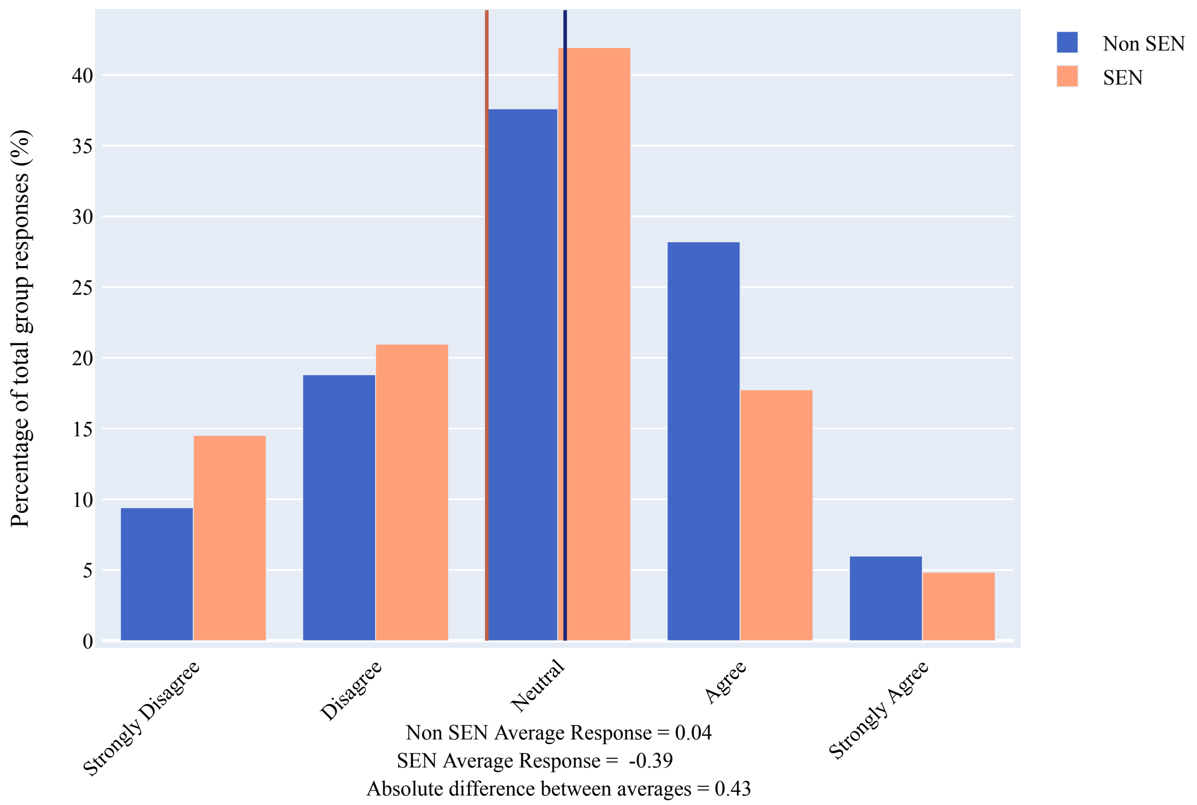
FSM Average Response = -0.3
Non FSM Average Response = -0.05
Absolute difference between averages = 0.25

If I wanted to I could get a job as a programmer.



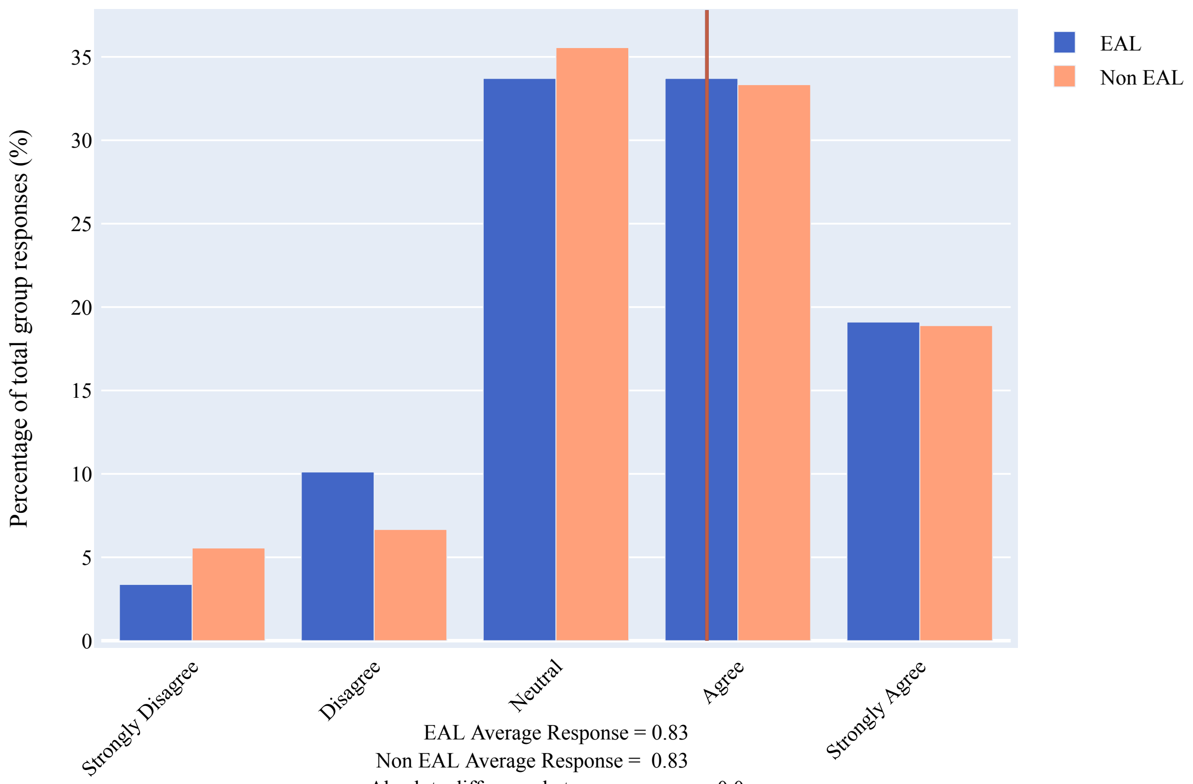
Male Average Response = 0.2
Female Average Response = -0.46
Absolute difference between averages = 0.66

If I wanted to I could get a job as a programmer.



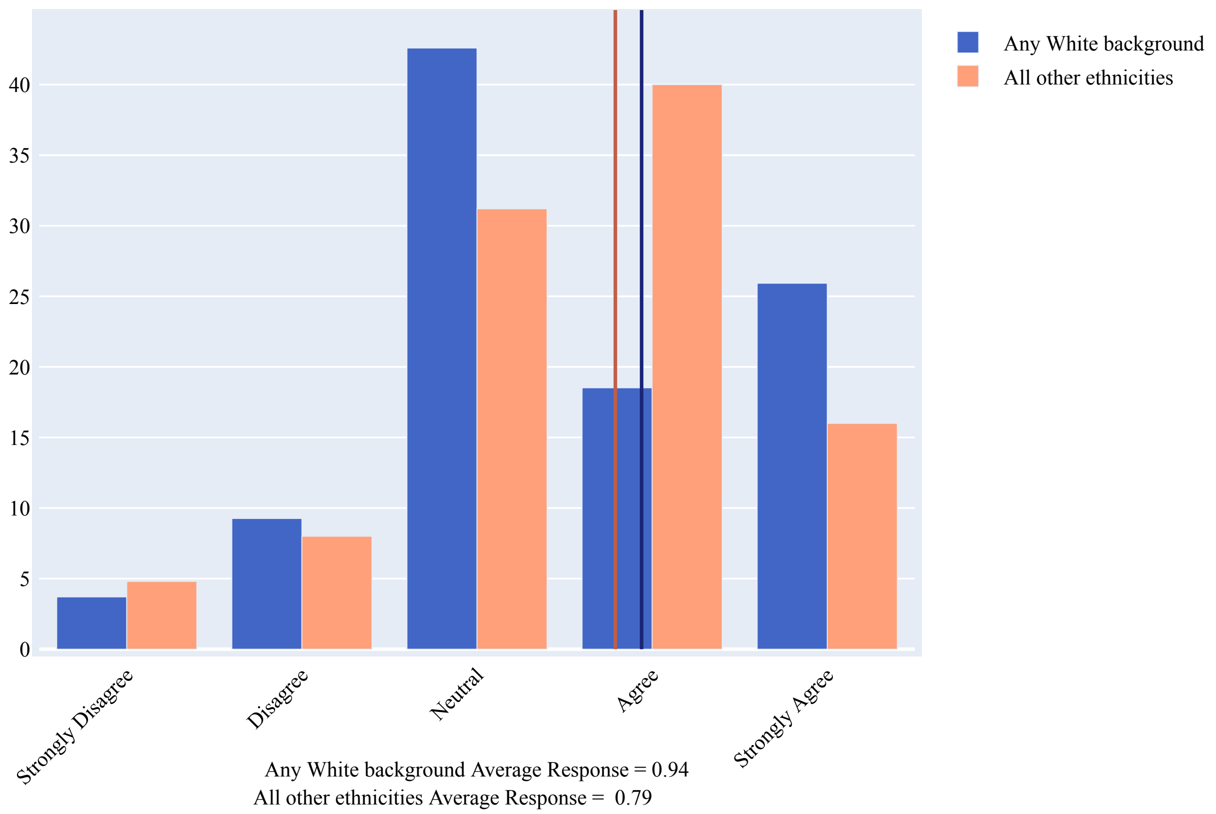
Non SEN Average Response = 0.04
SEN Average Response = -0.39
Absolute difference between averages = 0.43

There are lots of different types of jobs you can get by studying Computer Science.



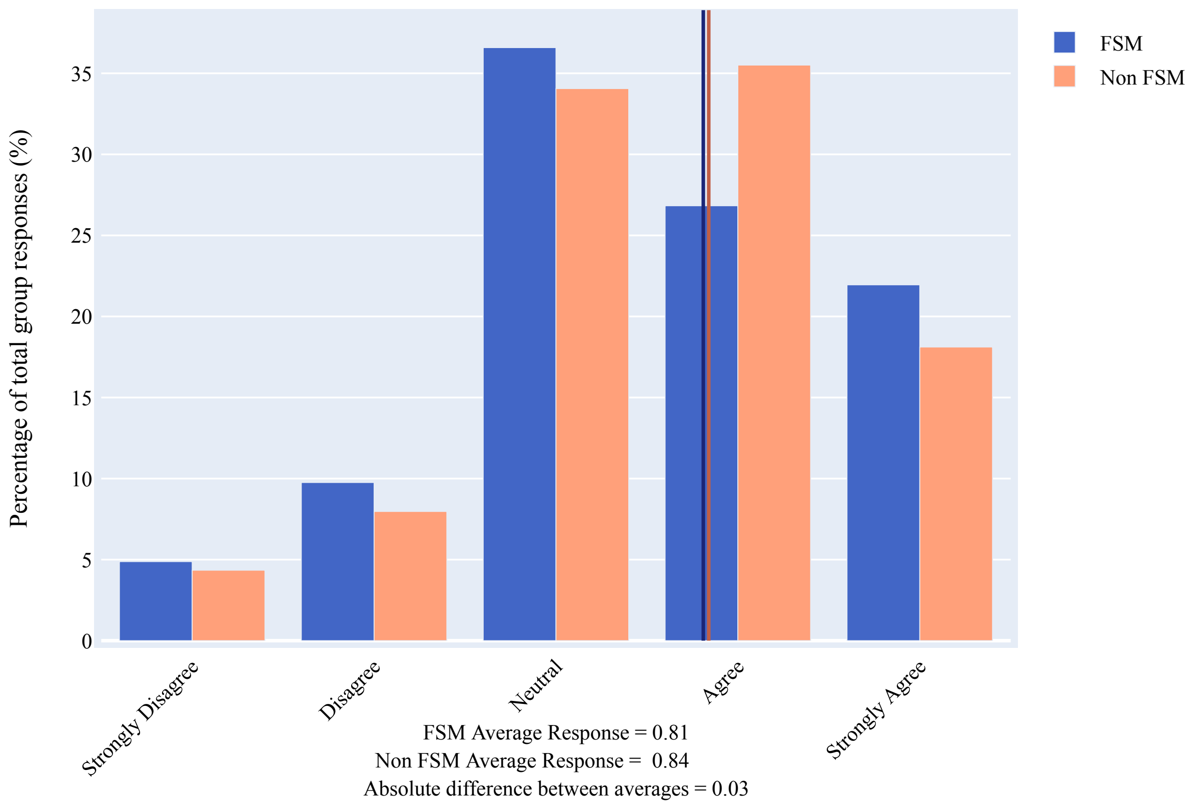
EAL Average Response = 0.83
Non EAL Average Response = 0.83
Absolute difference between averages = 0.0

There are lots of different types of jobs you can get by studying Computer Science.



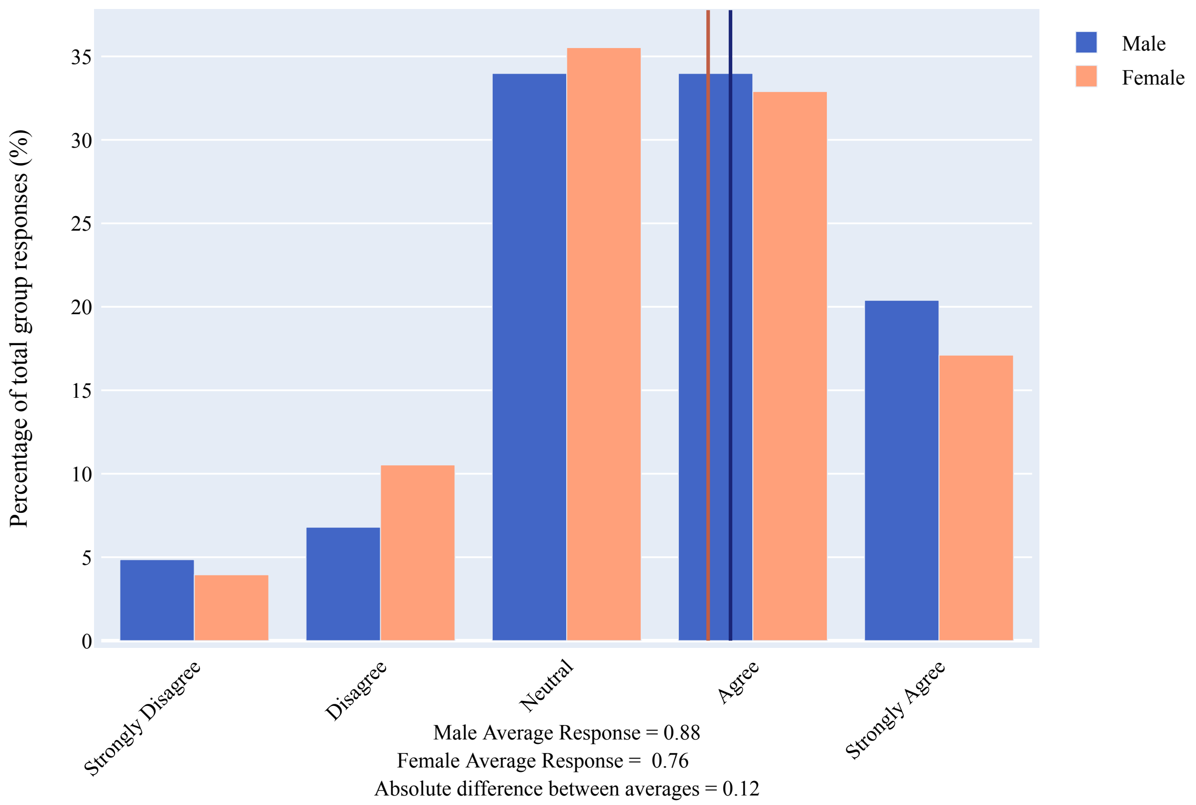
Any White background Average Response = 0.94
All other ethnicities Average Response = 0.79
Absolute difference between averages = 0.15

There are lots of different types of jobs you can get by studying Computer Science.



FSM Average Response = 0.81
Non FSM Average Response = 0.84
Absolute difference between averages = 0.03

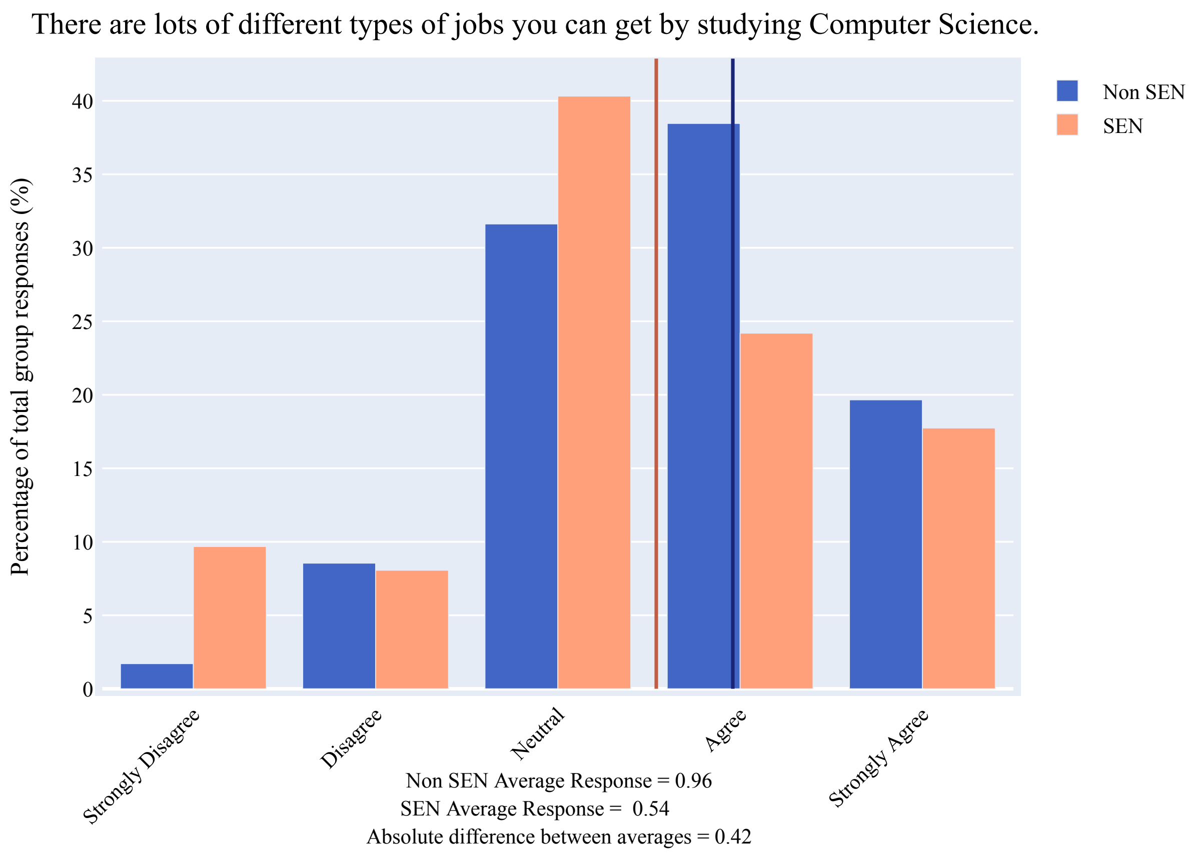
There are lots of different types of jobs you can get by studying Computer Science.



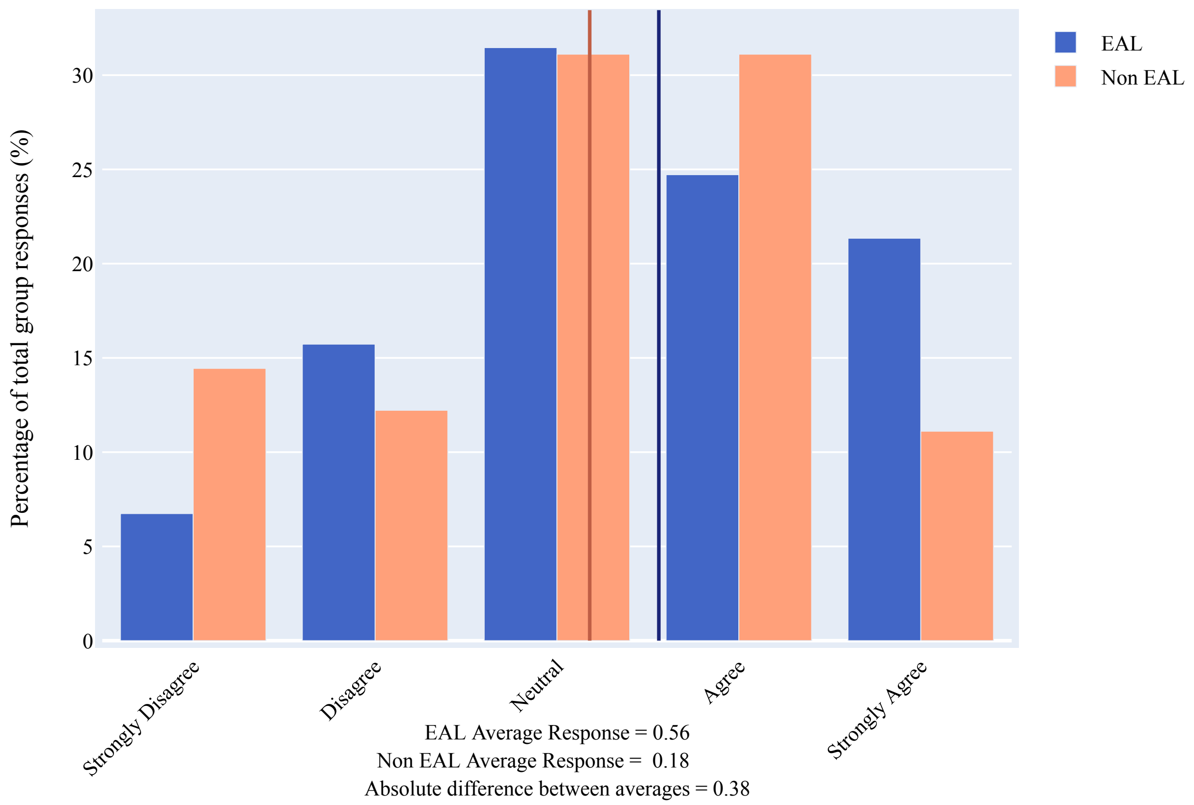
Male Average Response = 0.88

Female Average Response = 0.76

Absolute difference between averages = 0.12

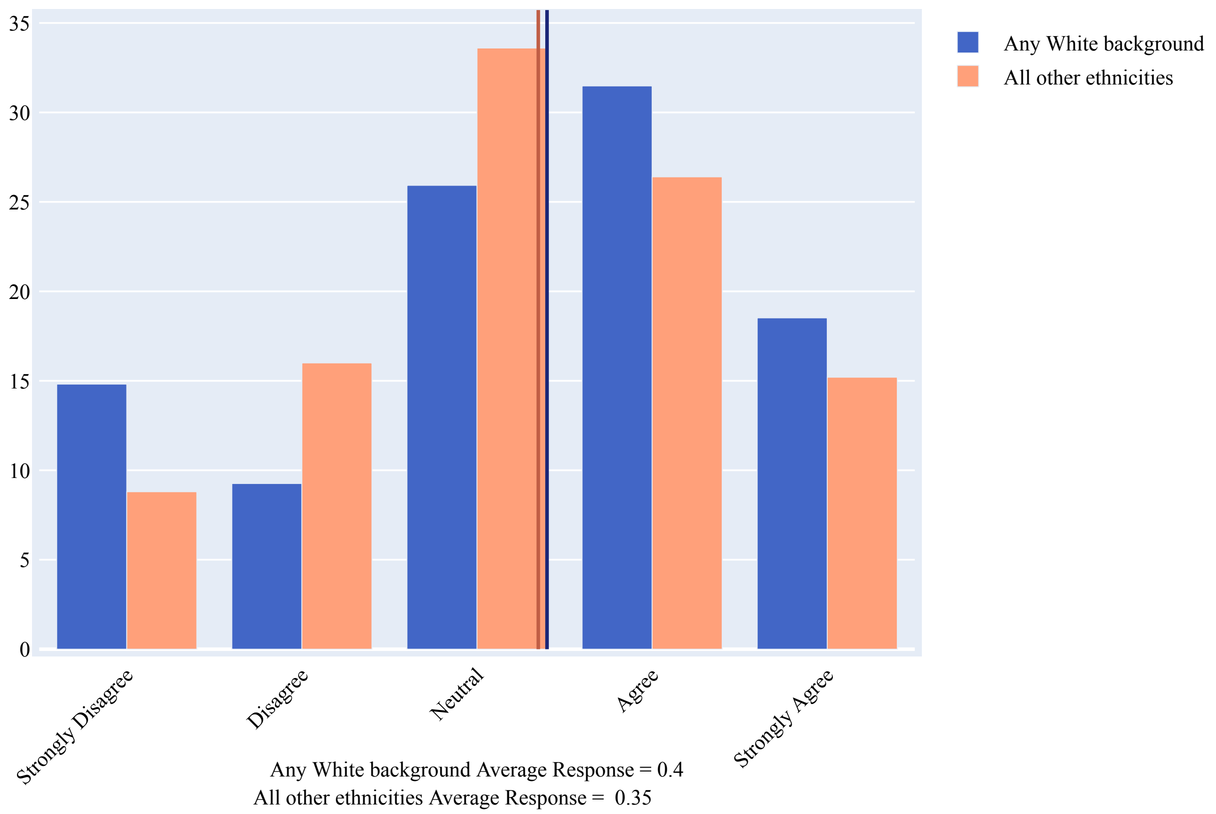

There are lots of different types of jobs you can get by studying Computer Science.
Percentage of total group responses (%)
0
5
10
15
20
25
30
35
40
Non SEN
SEN
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
Non SEN Average Response = 0.96
SEN Average Response = 0.54
Absolute difference between averages = 0.42

Computer Science is interesting.



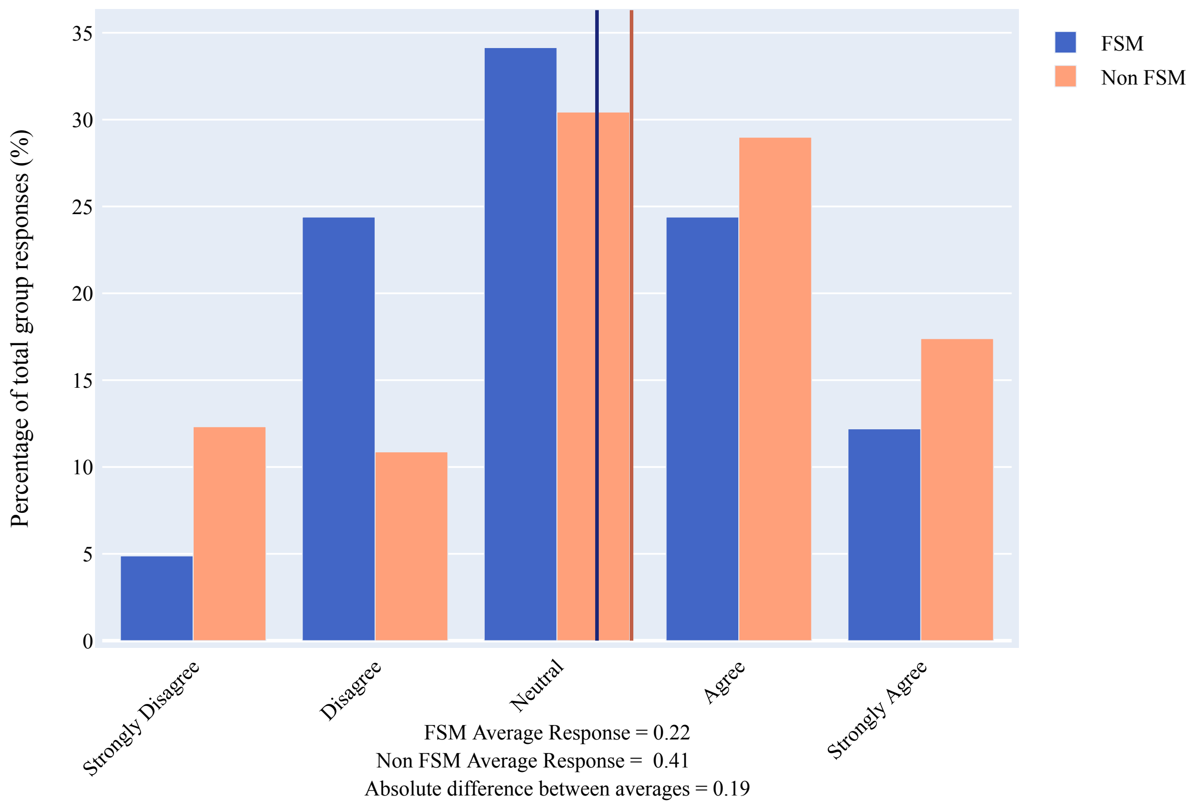
EAL Average Response = 0.56
Non EAL Average Response = 0.18
Absolute difference between averages = 0.38

Computer Science is interesting.



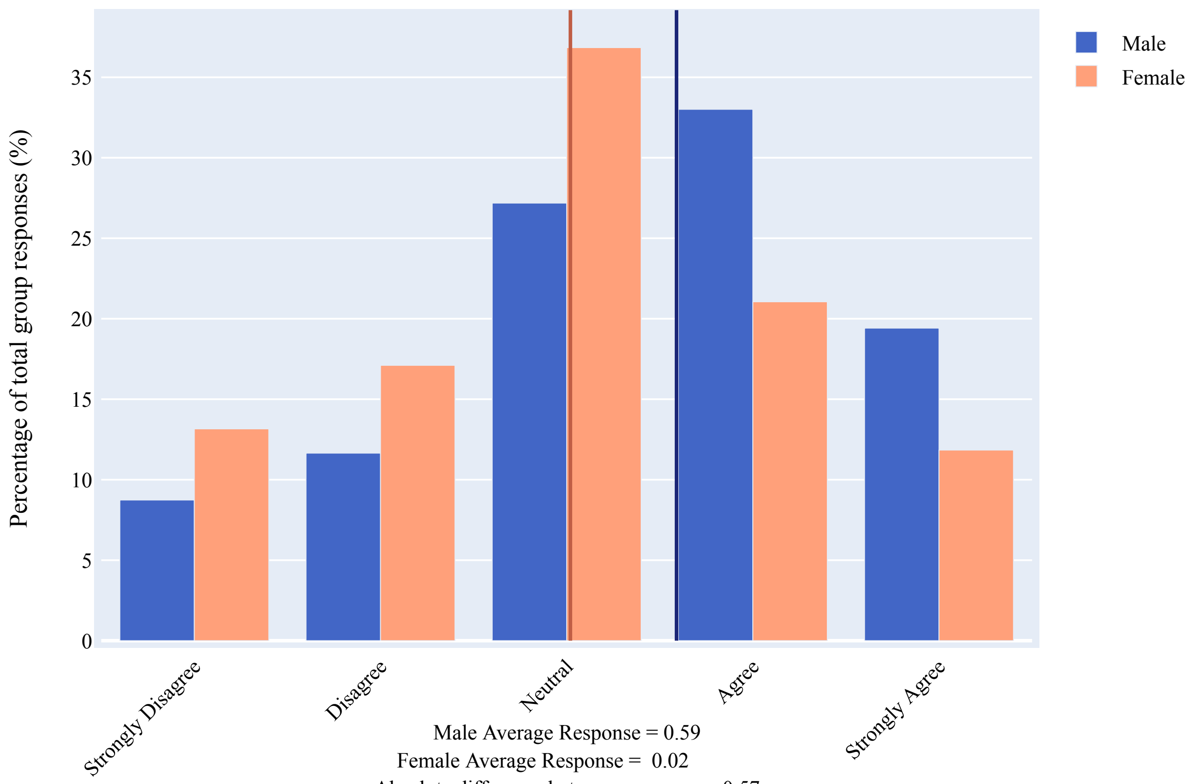
Any White background Average Response = 0.4
All other ethnicities Average Response = 0.35
Absolute difference between averages = 0.05

Computer Science is interesting.



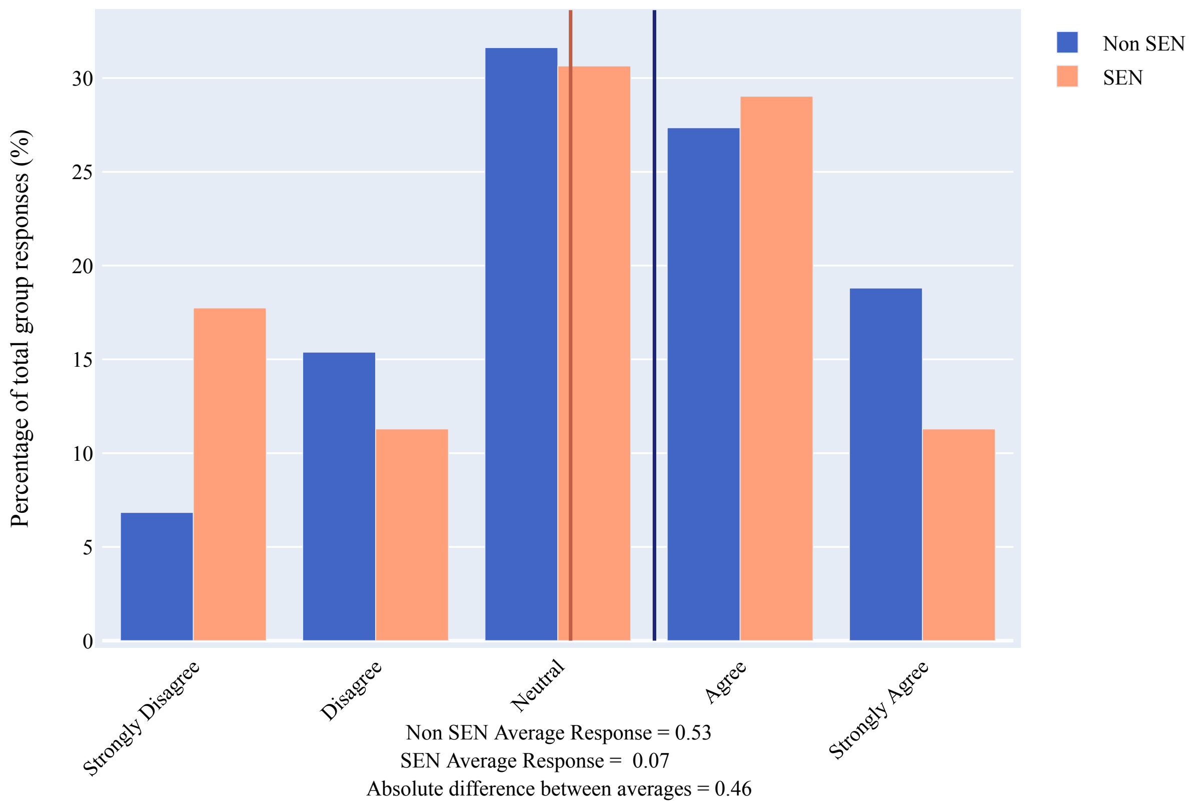
FSM Average Response = 0.22
Non FSM Average Response = 0.41
Absolute difference between averages = 0.19

Computer Science is interesting.



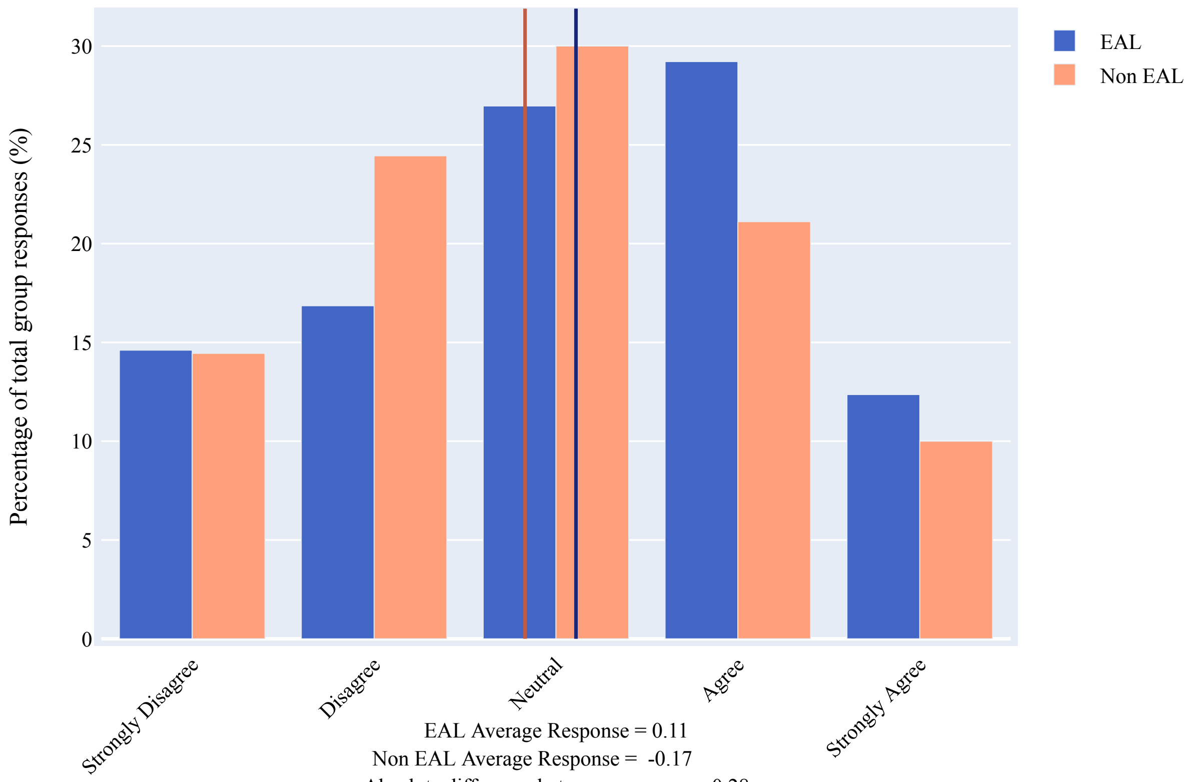
Male Average Response = 0.59
Female Average Response = 0.02
Absolute difference between averages = 0.57

Computer Science is interesting.

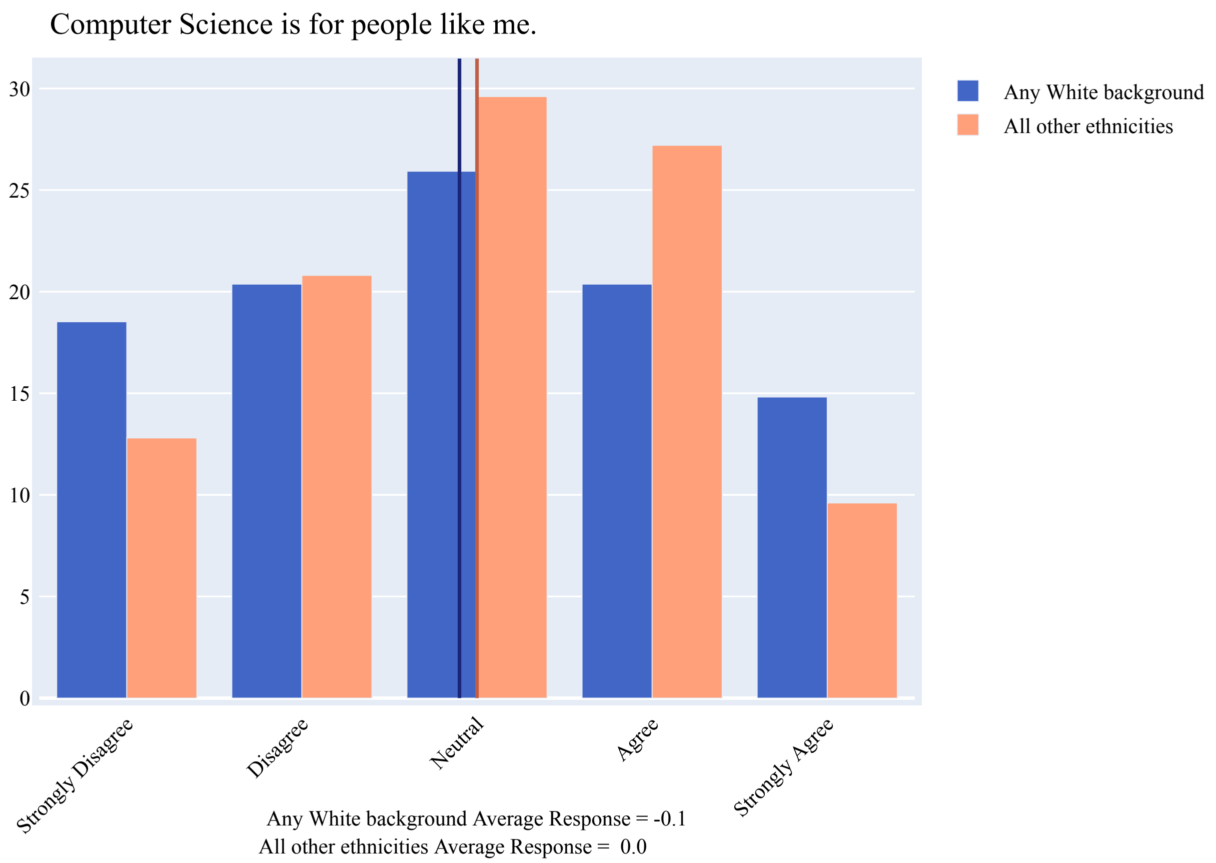


Non SEN Average Response = 0.53
SEN Average Response = 0.07
Absolute difference between averages = 0.46

Computer Science is for people like me.

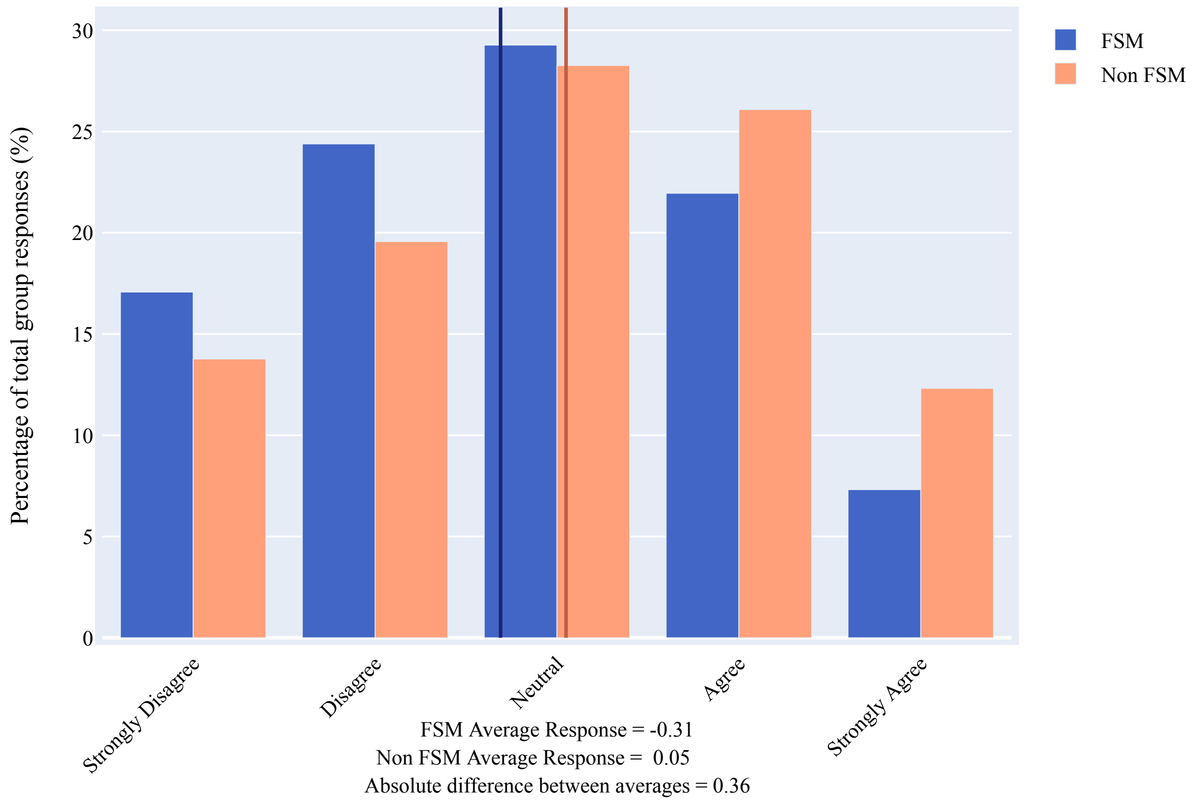


EAL Average Response = 0.11
Non EAL Average Response = -0.17
Absolute difference between averages = 0.28



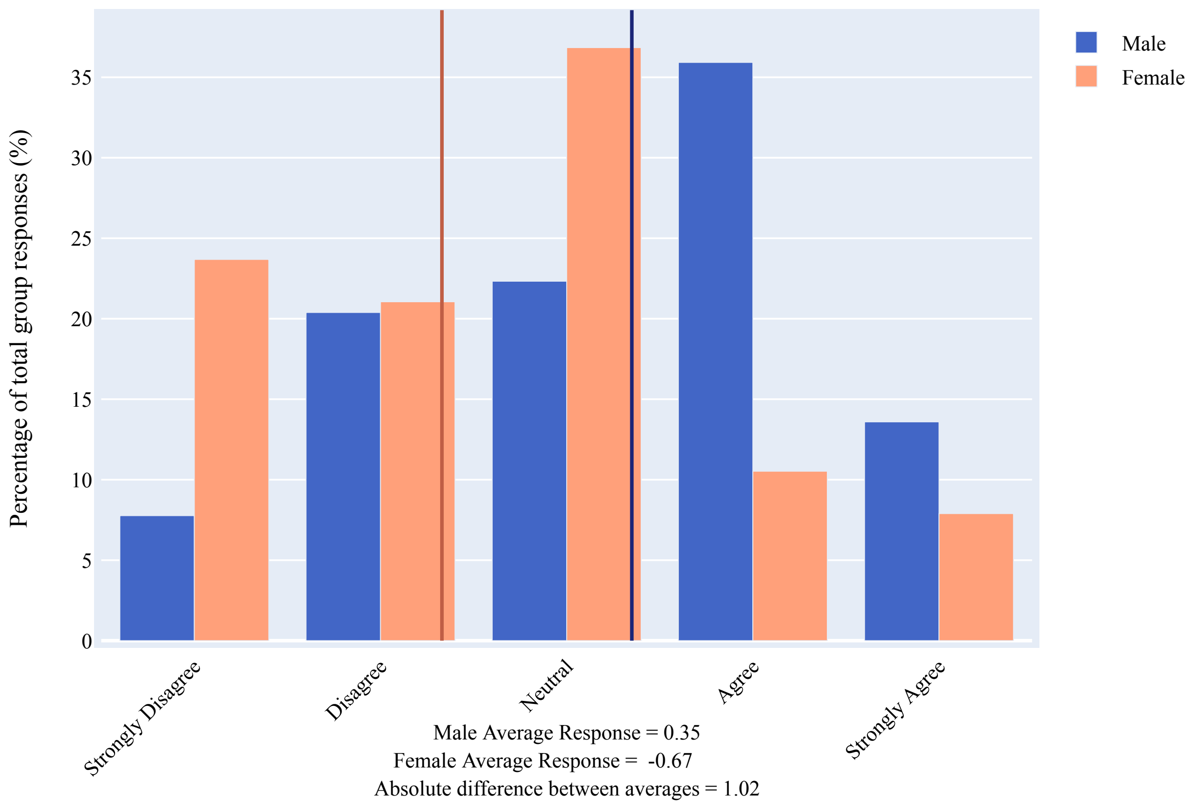
Any White background Average Response = -0.1
All other ethnicities Average Response = 0.0
Absolute difference between averages = 0.1

Computer Science is for people like me.



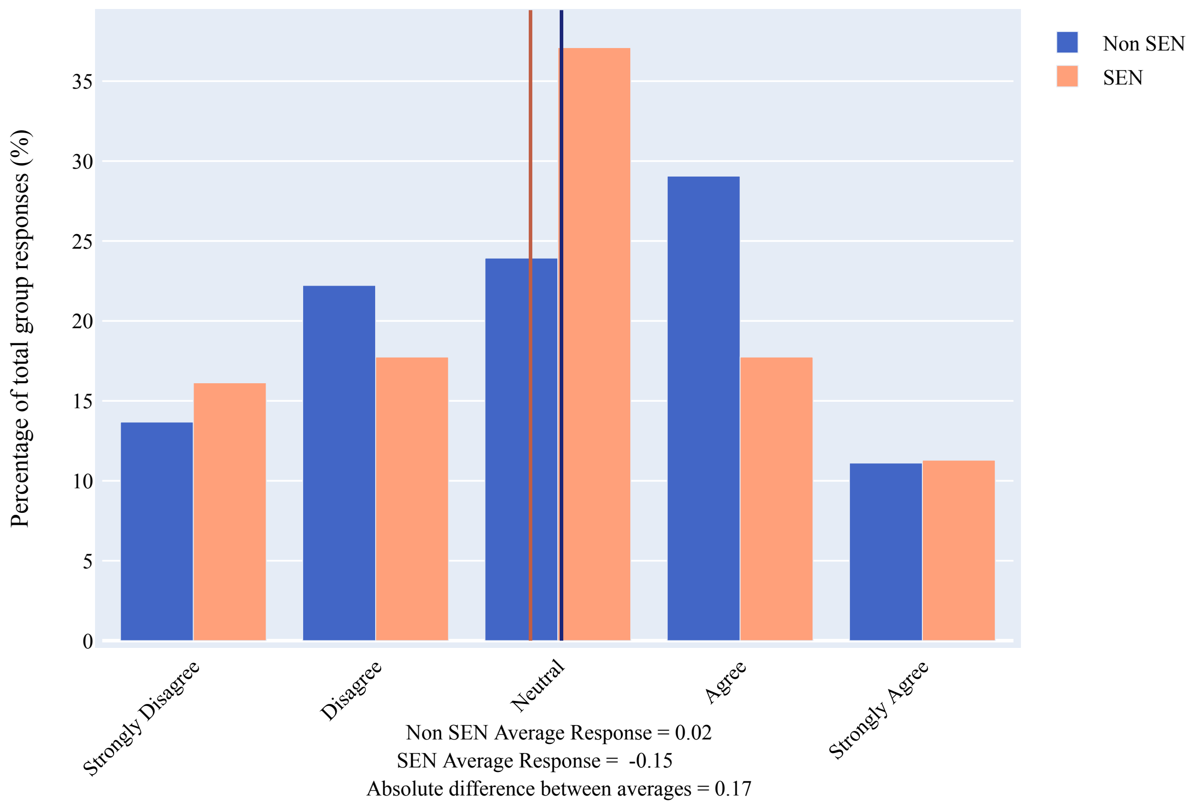
FSM Average Response = -0.31
Non FSM Average Response = 0.05
Absolute difference between averages = 0.36

Computer Science is for people like me.



Male Average Response = 0.35
Female Average Response = -0.67
Absolute difference between averages = 1.02

Computer Science is for people like me.


Percentage of total group responses (%)
Non SEN
SEN
Strongly Disagree
Disagree
Neutral
Agree
Strongly Agree
Non SEN Average Response = 0.02
SEN Average Response = -0.15
Absolute difference between averages = 0.17