STAC33

Assignment 1

Due Tuesday January 21 at 11:59pm

My assignments contain two kinds of questions: the ones referring to PASIAS are for your practice, and if you look in PASIAS you will see solutions, which you can check your solution against. The other questions, usually at the end, are an assignment: you need to do these questions yourself and hand them in (instructions below). These are due on the date shown above. An assignment handed in after the deadline is late, and may or may not be accepted (see course outline). I am usually OK with a few minutes late, but not more than that. Don't take that risk.

My solutions to the assignment questions will be available when everyone has handed in their assignment. These are *my* solutions; you can have something different and still be correct.

The grader will have 200-plus assignments to mark, so it is vitally important that you make your assignment *easy* for the grader to deal with. Your answers and explanations *must be easy* for the grader to find. A simple structure is, for each part of each question in order, to put these three things:

- Your *code*
- Your *output*
- Your *answers and explanation*

When you use an R Notebook with your comments below the output, and "preview" the result, the HTML (or Word) file that comes out is in this format. This is why I talk about R Notebooks in class. You can do it differently, but then you have extra work to organize things. *Hand in the HTML file, not the* `.Rmd` *file.* There are no marks for handing in an `.Rmd` file, because *you* need to run your code. You can not expect the grader to run your code on your behalf. That's *your* job. When you hand in your work on Quercus, download it again to check that you handed in what you thought you did.

You are reminded that work handed in with your name on it must be *entirely your own work*. It is as if you have signed your name under it. If it was done wholly or partly by someone else, *you have committed an academic offence*, and you can expect to be asked to explain yourself. The same applies if you allow someone else to copy your work. The graders will be watching out for assignments that look suspiciously similar to each other (or to my solutions). Besides which, if you do not do your own assignments, you *will* do badly on the exams, because the struggle to figure things out for yourself is an important part of the learning process.

Assignments are to be handed in on Quercus. See `https://www.utsc.utoronto.ca/~butler/c32/quercus1.nb.html` for instructions on handing in assignments in Quercus. *Allow yourself time to figure out what you need to do.*

As with any other course involving software, *there are no extensions due to failure to access software.* It is *your* responsibility to make sure that you allow yourself enough time to get connected to R, and to get any packages installed and working. (This is more of an issue if you are using R Studio Cloud, which might be busy and therefore slow; if you install R and R Studio on your own computer, you won't be fighting with other people for resources, but you will need to get everything set up.)

Once you are sitting in front of R Studio (either on `rstudio.cloud` or on your own computer), you would do well to make a new notebook, create a code chunk, in it put

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

and run it (like I just did). If you don't get output like I did, instead getting an error like "no such package", you need to install the Tidyverse first, like this:

```
install.packages("tidyverse")
```

and then you can try again. If at any point R Studio invites you to install any other packages, let it do so.

1. Work through chapter 4 of PASIAS, along with whatever of chapter 3 that you haven't worked through yet.

   Hand the next question in:

2. The Princeton Review conducts an annual survey of high school students who are applying to college, and also of the parents of students who are applying to college. (These are separate surveys; the number of students surveyed is bigger than the number of parents surveyed.) One of the questions was "how far from home would you like the college you attend to be", with students asked to choose one of four categories: less than 250 miles, 250–500 miles, 500–1000 miles, greater than 1000 miles. In the survey that parents completed, the question was "how far from home would you like the college your child attends to be?", with the same distance categories. The data are in `https://www.utsc.utoronto.ca/~butler/assgt_data/distance.csv`. Each row of the data file represents one respondent; the column `who` indicates whether the person responding was one of the students or one of the parents, and the column called `distance` indicates the person's preferred distance category.

   (a) (3 marks) Read in and display (some of) the data. How many people responded to the survey altogether?

   Solution:

   This is a `.csv` file, so:

```
my_url <- "https://www.utsc.utoronto.ca/~butler/assgt_data/distance.csv"
prefs <- read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##  distance = col_character(),
##  who = col_character()
## )
```

```
prefs
```

```
## # A tibble: 15,722 x 2
##    distance    who
##    <chr>       <chr>
##  1 250 or less students
##  2 250 or less students
##  3 250 or less students
##  4 250 or less students
##  5 250 or less students
##  6 250 or less students
##  7 250 or less students
##  8 250 or less students
##  9 250 or less students
## 10 250 or less students
## # ... with 15,712 more rows
```

Give the data frame whatever name you like. `distance` is a good name to avoid, since that's the name of one of the columns.

There are 15,722 rows, so that's how many people responded to the survey. (You ought to note also that you have the columns that were promised. These are also displayed by `read_csv`.)

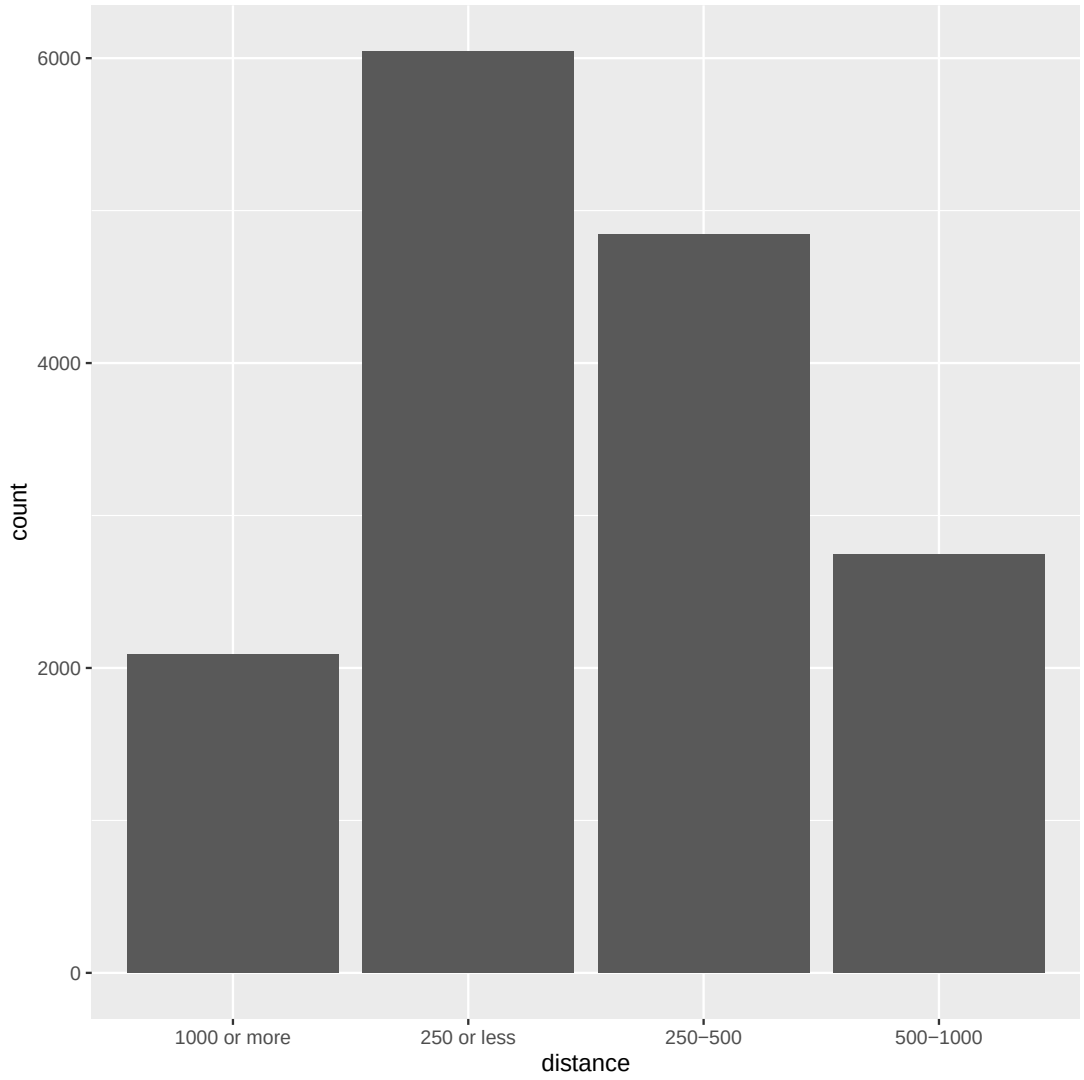
Extra: the preferred `distance` for a student to be from home is categorical, since in the survey everyone was offered a choice between those four categories, and so all we can do by way of summary (next part) is to count how many individuals fell in each category. This is true even though "preferred distance from home" is actually a number; we don't have access to the actual numbers, just to the categorized versions of them, so it is categorical rather than quantitative. This happens a lot on surveys; the actual number might be hard to name exactly, or the people collecting the data don't need to know the actual number, and so respondents are asked which one of a number of categories they fall into. If you have actual numbers, then turning them into categories loses information (as well as, sometimes, making things hard to analyze), but categories are what we have here.

(b) (2 marks) Make a suitable graph of the categorical column `distance` (ignoring `who` for the moment).

Solution:

One categorical variable implies a bar chart, counting up how many observations there were in each category:

```
ggplot(prefs, aes(x=distance)) + geom_bar()
```



This was meant to be a gentle start.

(c) (2 marks) What do you notice that is odd about your bar chart? Why do you think it came out that way? Explain briefly.

Solution: The distances are in the wrong order: the biggest one is at the beginning!

Categories of categorical variables are usually names, and what R does is to put them in alphabetical order before it makes the bars. Often that is sensible, but with numbers it is often not. Here, in alphabetical order, 1000 is before 250 (because 1 is before 2).

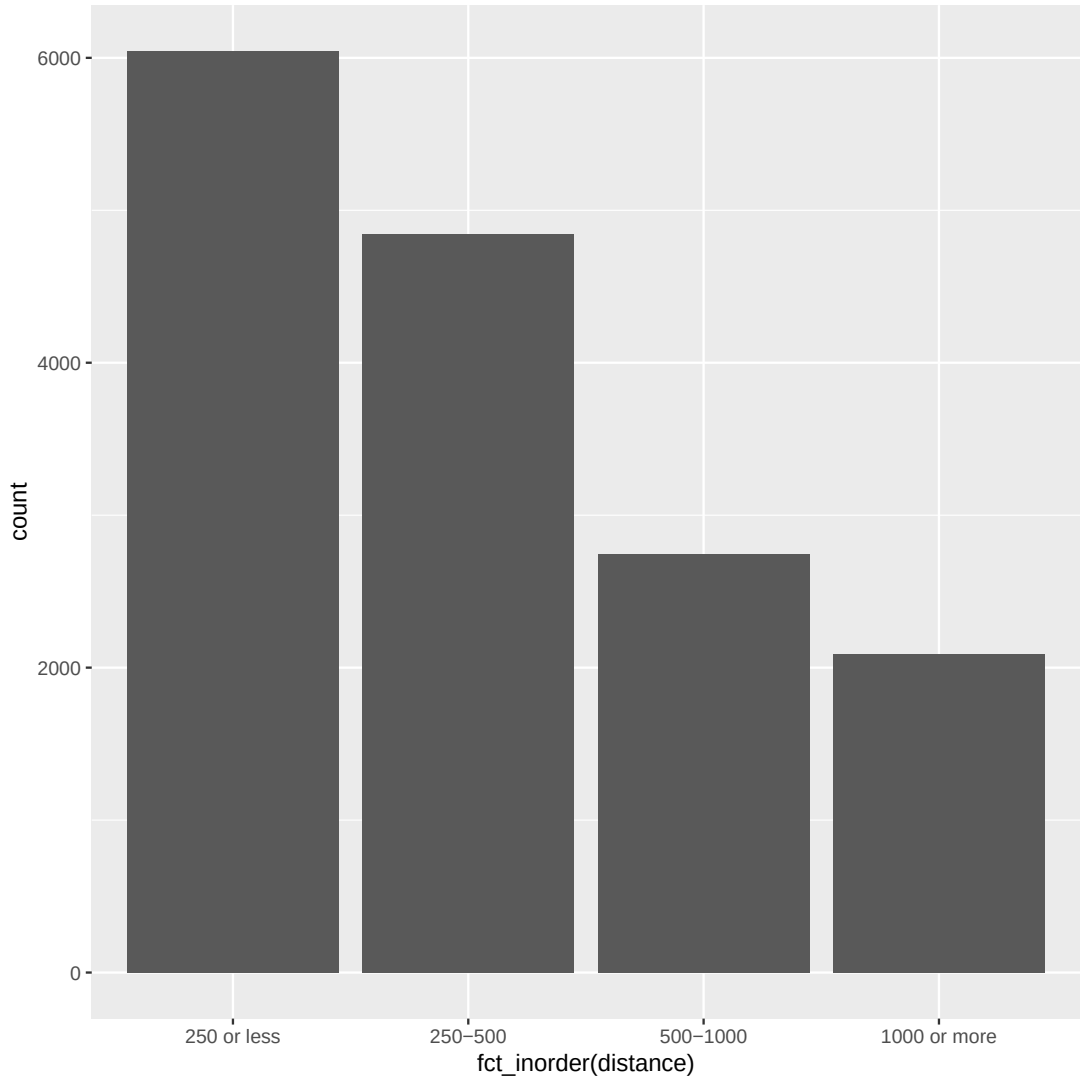
Two points, one for noting that the categories are out of order, and one for saying something about categories being in alphabetical order (and that this makes no sense here).

(d) (2 marks) In this case, it makes more sense to arrange the categories in the order they appear in the data. To do this, in place of `distance` use `fct_inorder(distance)` in the code for your graph. Does this put the categories in a sensible order now? (Use `fct_inorder` for the distances in the rest of this question.)

Solution:

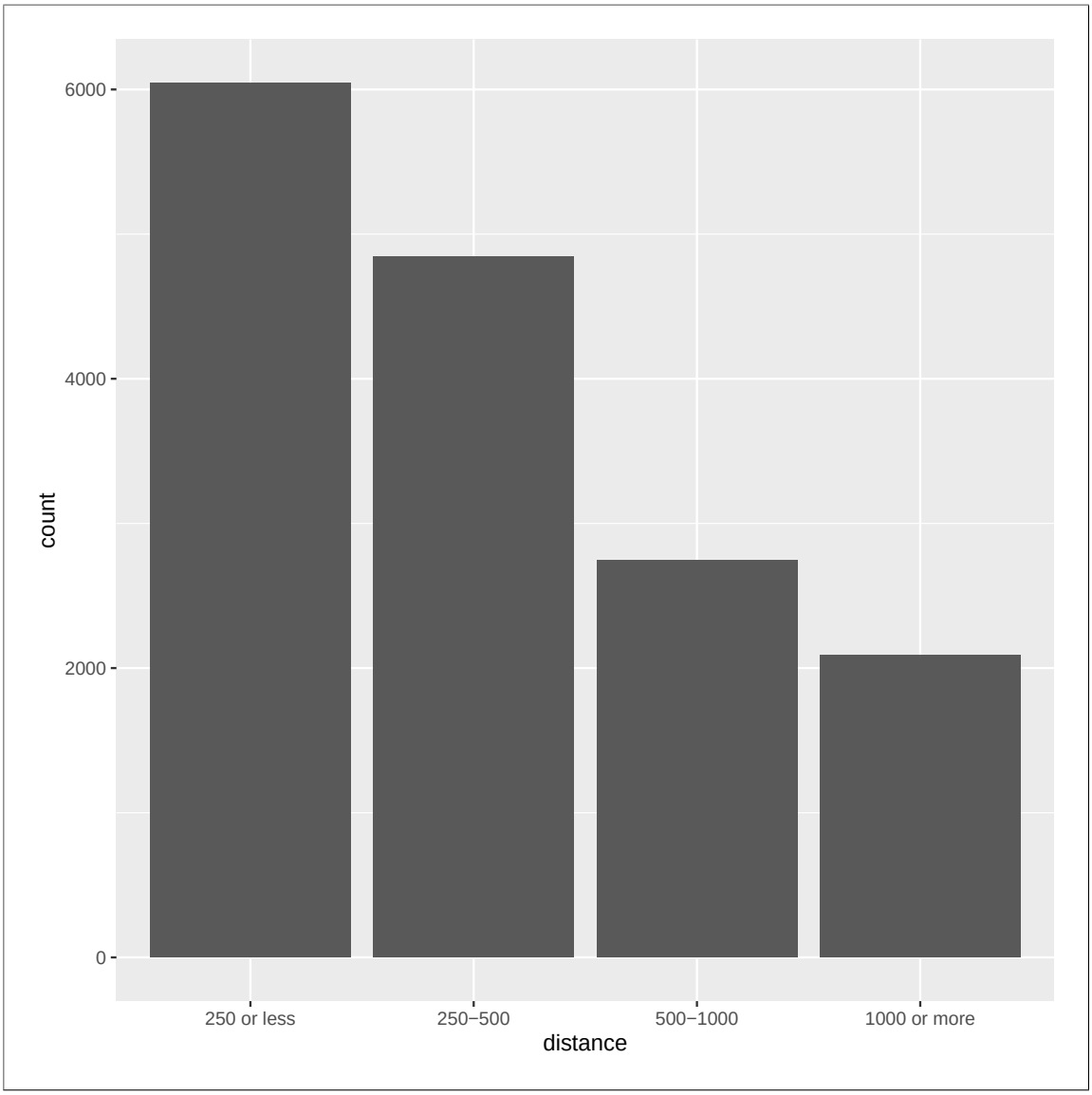
Try it and see (which ought to be a motto for this course):

```
ggplot(prefs, aes(x=fct_inorder(distance))) + geom_bar()
```



The distances are in the right order now, but the x-axis has a weird label, which you can explicitly reset like this, if you wish (not needed here, but for future reference):

```
ggplot(prefs, aes(x=fct_inorder(distance))) + geom_bar() +
  xlab("distance")
```



(e) (2 marks) What does your graph tell you, in the context of the data?

Solution: Several possible angles to take:

- Most people would prefer the student to be 250 miles or less from home
- Most people would prefer the student *not* to be more than 1000 miles from home
- More people would prefer the student to be closer to home rather than further away

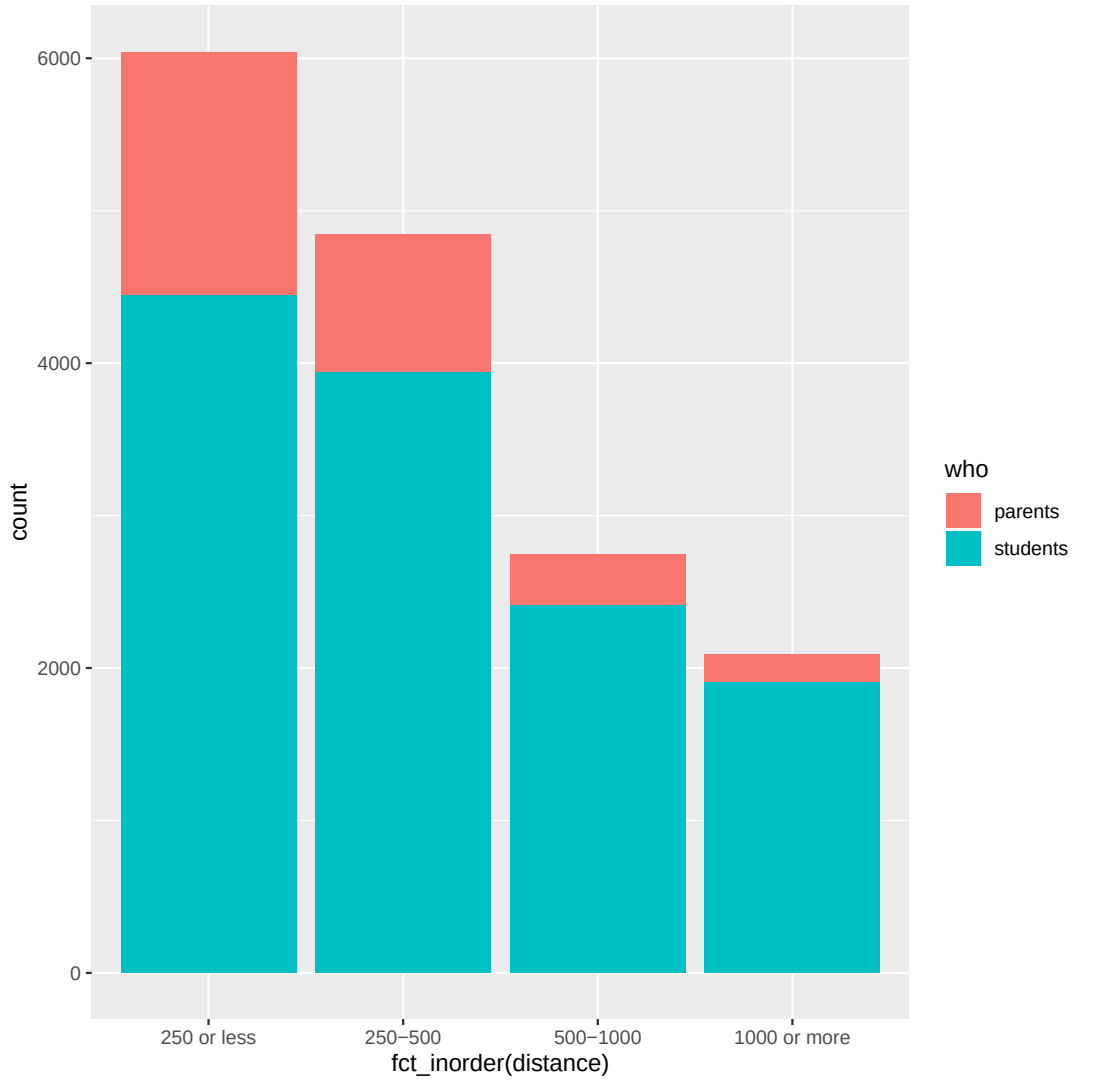
or something like one of those.

(f) (2 marks) Make a suitable graph that shows both distances and whether the person involved was a student or a parent. Hints: not stacked, and also you will have to make a decision about which variable is `x` and which is `fill`.

Solution:

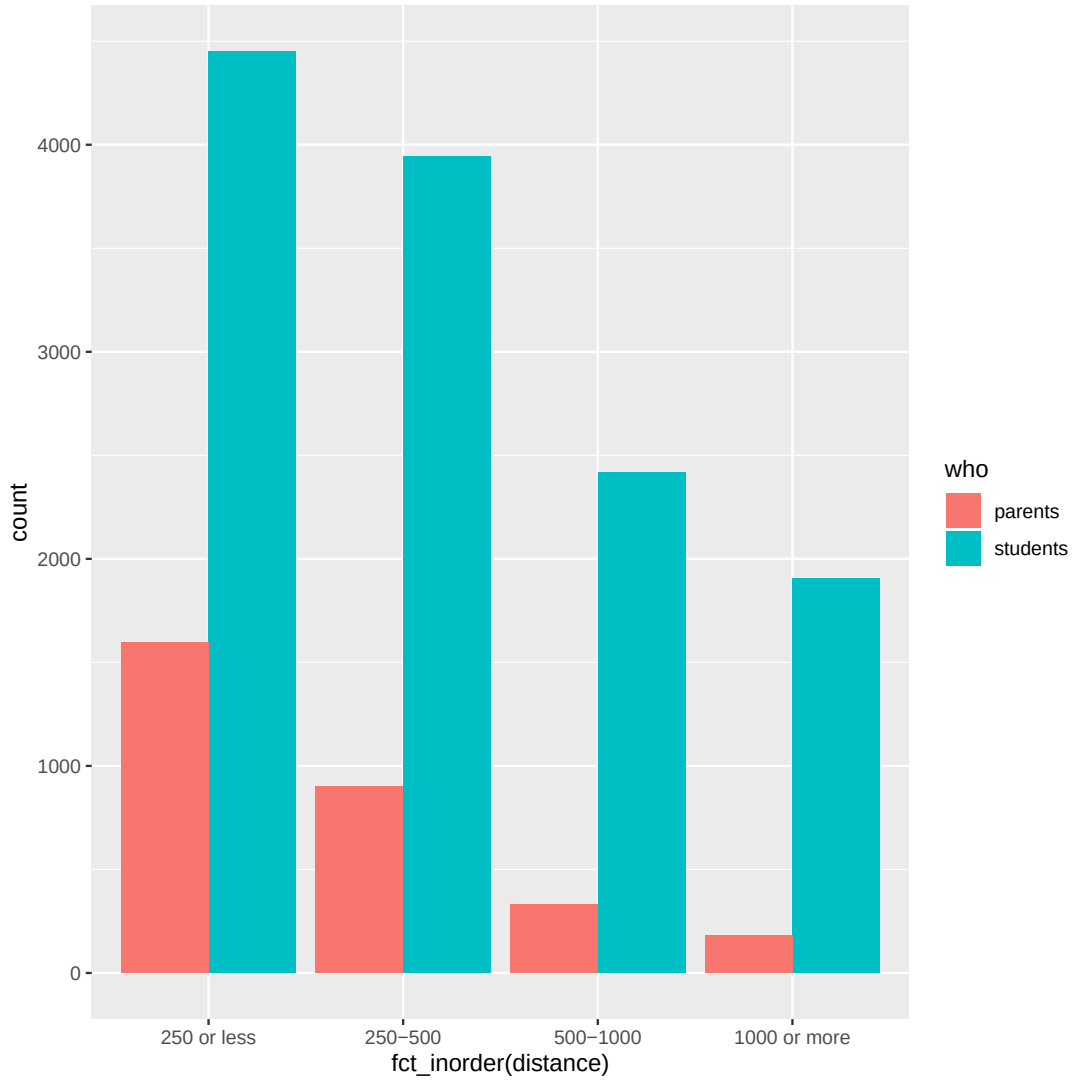
This is now of *two* categorical variables, so you need something like a grouped bar chart. The default way of doing this comes out with the bars stacked:

```
ggplot(prefs, aes(x=fct_inorder(distance), fill=who)) + geom_bar()
```



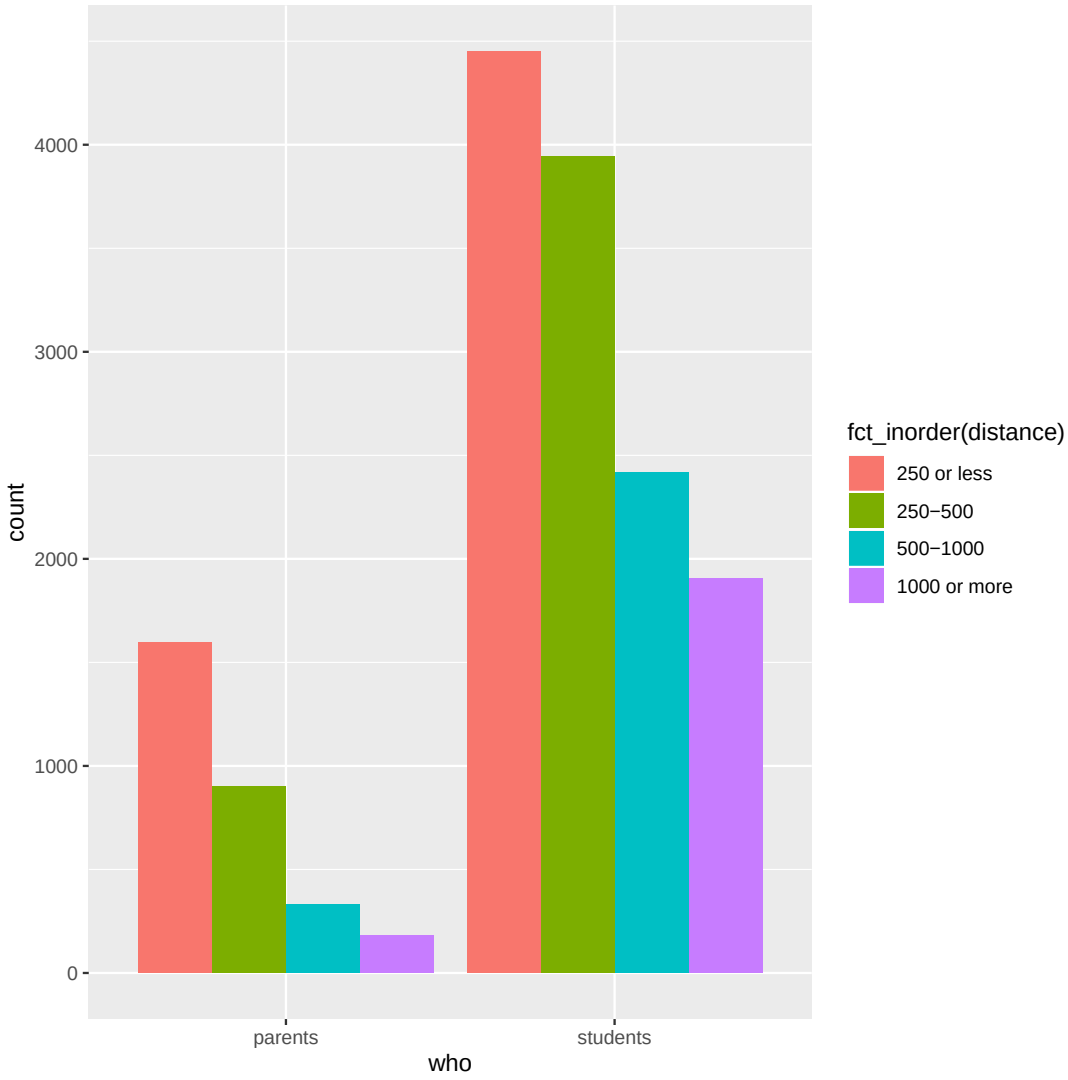
I don't like this, because the eye needs to compare the relative sizes of the bars that are on top of each other. The comparison is much easier if the bars are side by side, which is called `dodge` in `ggplot`:

```
ggplot(prefs, aes(x=fct_inorder(distance), fill=who)) + geom_bar(position="dodge")
```



I'll accept this, but you might wonder what happens if you switch the roles of `x` and `fill` around:

```
ggplot(prefs, aes(fill=fct_inorder(distance), x=who)) + geom_bar(position="dodge")
```



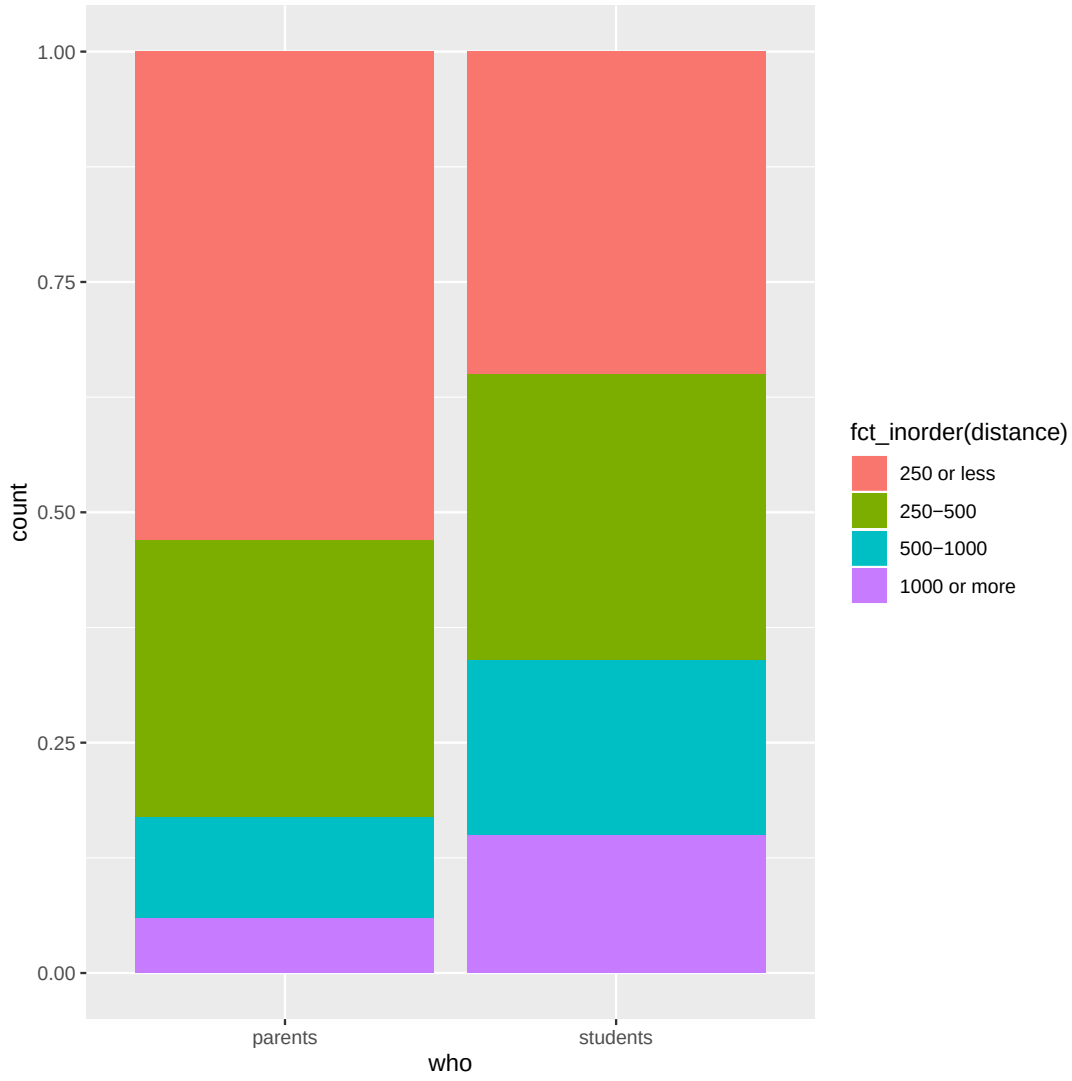
I like this the best, because it enables you to say "out of students, more would prefer to be 250 miles away or less, and this is also true for parents". The story is clear even though there are fewer parents than students in the survey. I guess you can get that from the other graph too, but it seems more work to come to that conclusion.

For these data, you could think of `who` as being explanatory (input) and `distance` as being a response (outcome). In the same way as when you make a frequency table you sum over the explanatory ("out of parents, how many preferred this distance?"), on a grouped bar chart you use the explanatory as `x` and let the response be `fill`.

(g) (3 marks) Another way to make a grouped bar chart is with `position="fill"`. Try that for these data (with `who` as `x`). What has happened? Explain briefly.

Solution: Did I say "try it and see" enough yet?

```
ggplot(prefs, aes(x=who, fill=fct_inorder(distance))) + geom_bar(position="fill")
```



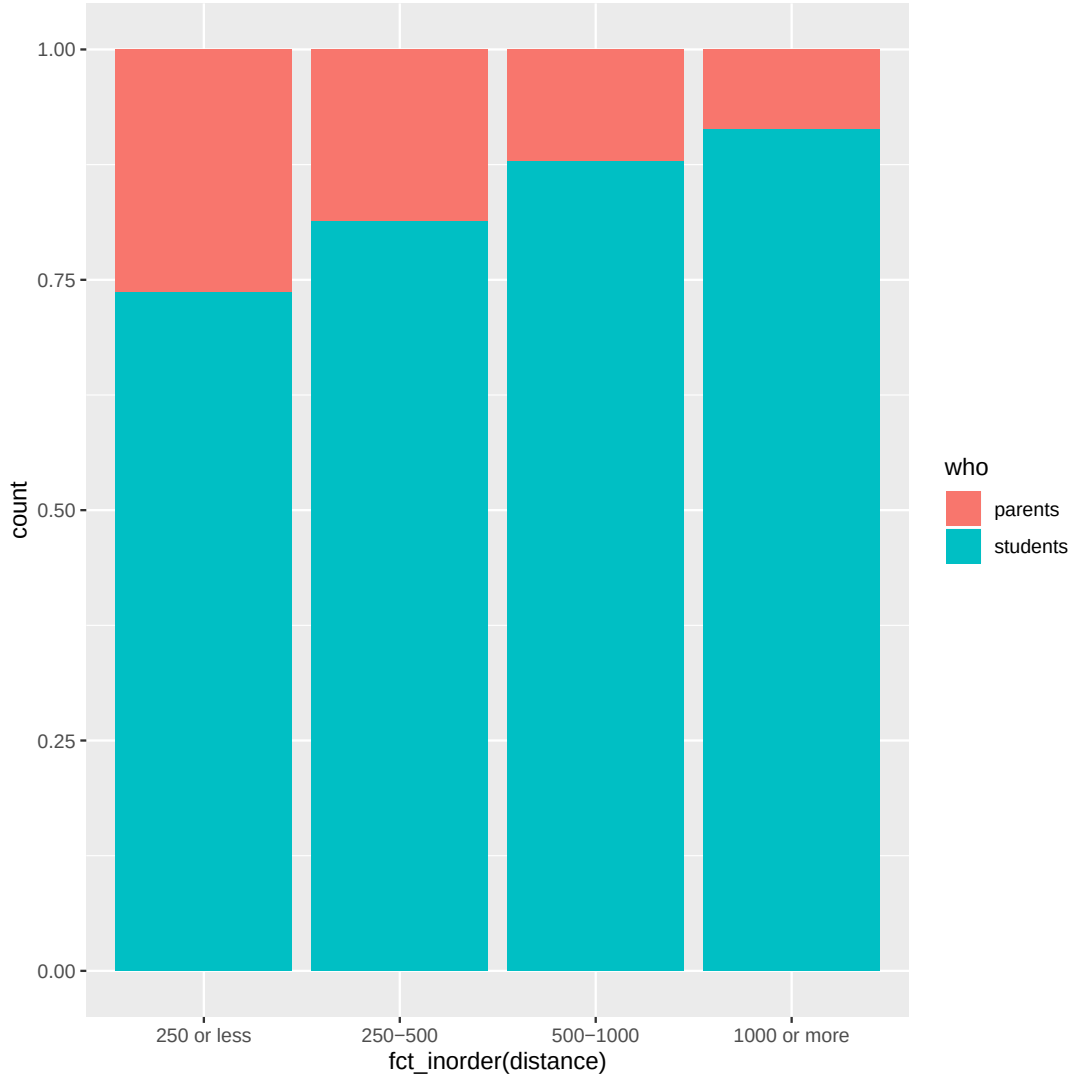
Each bar has been standardized to the same length (even though there were more students than parents in the survey), and the size of the coloured pieces reflects what fraction of the people in each group preferred each distance.

(That is going to be hard to put into words, but see what you can do.)

Extra: the value of this kind of plot is that you can compare the two red areas and see that a greater fraction of the parents would prefer their children to attend a college close to home, as compared to the fraction of *students* who would like to be close to home. Also you could compare the purple pieces at the bottom and say that even though not many of the students want to be over 1000 miles from home, *even fewer* of the parents would like their children to go to college so far away.

Having the `x` and `fill` the other way around makes less sense:

```
ggplot(prefs, aes(fill=who, x=fct_inorder(distance))) + geom_bar(position="fill")
```



Most of the people in all four distance categories are students (which is telling you only that there were more students than parents in the survey). I guess you can say there was less of a minority of parents for the shortest distance, but it seems less clear about what that means.