R Ladies Data Visualization

Limor Raviv

28 March 2019

Introduction

The following Rmarkdown document includes a detailed example of how to visualize data and plot regression models' output.

We'll use the "GSSvocab" dataset, which contains information from the General Social Survey (GSS) of the University of Chicago. It inlcudes vocabulary scores collected over the course of 20 years from over 28,000 people. We'll analyze the vocabulary scores by individuals' age, gender, education level and nativeness.

Feel free to reuse and edit any part of this document/code!

Reminder: What's Rmarkdown?

This is an R Markdown document. It basically combines text with R code (models, plots etc), and can be used to create beautiful HTMLs, PDFs, Word documents, slides and even websites.

When you click on the "Knit" button on top, it will generate a document that includes all the specified content: this text, as well as the output of any embedded R code chunks within the document (unless you decide not to include it in your final output).

For more details on using R Markdown see http://rmarkdown.rstudio.com.

I also recommend using this awesome cheat sheet: https://github.com/rstudio/cheatsheets/raw/master/rmarkdown-2.0.pdf

Text in Rmarkdown

Text appears like this, on a white background.

You can format the text to be **bold** or *italics*, and have it appear in different sizes by starting a line with hashtags:

for main headers

for subheaders

for subsubheaders

You can also use lists and bullet-points:

1. This
2. is
3. a list

- and
- these

- are
- bullets

And even write nice equations by using the dollar sign:

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Code in Rmarkdown

Chuncks of code appear in a grey box denoted by "‘ at the begining and end, and have a curly-brackets header (see below). The code always starts with some name, and then some technical instructions (e.g., do you want to include the actual code in the document or just the output? Do you want to see warnings?). For example "echo=TRUE" means I want the code itself to appear in the final file (not just an output, if any). Check out the cheat sheet for more details.

```
example <- 1.987
```

You can also include some R code inside the text by using your code in grave accents. For exmaple, 2 multiplied by 10 equals 20. This can be used to integrate values from your enviroment (like beta-coefficients) in the actual text without the need to copy them, like the value from the exmaple above is 1.987.

Let's get started!

For editing and running the code, please install and load the following packages first.

Note that "include=FALSE" here means that this chunk of code will not appear in the final document.

The dataset

Now, let's load the dataset and play with it a bit to see what's going on.

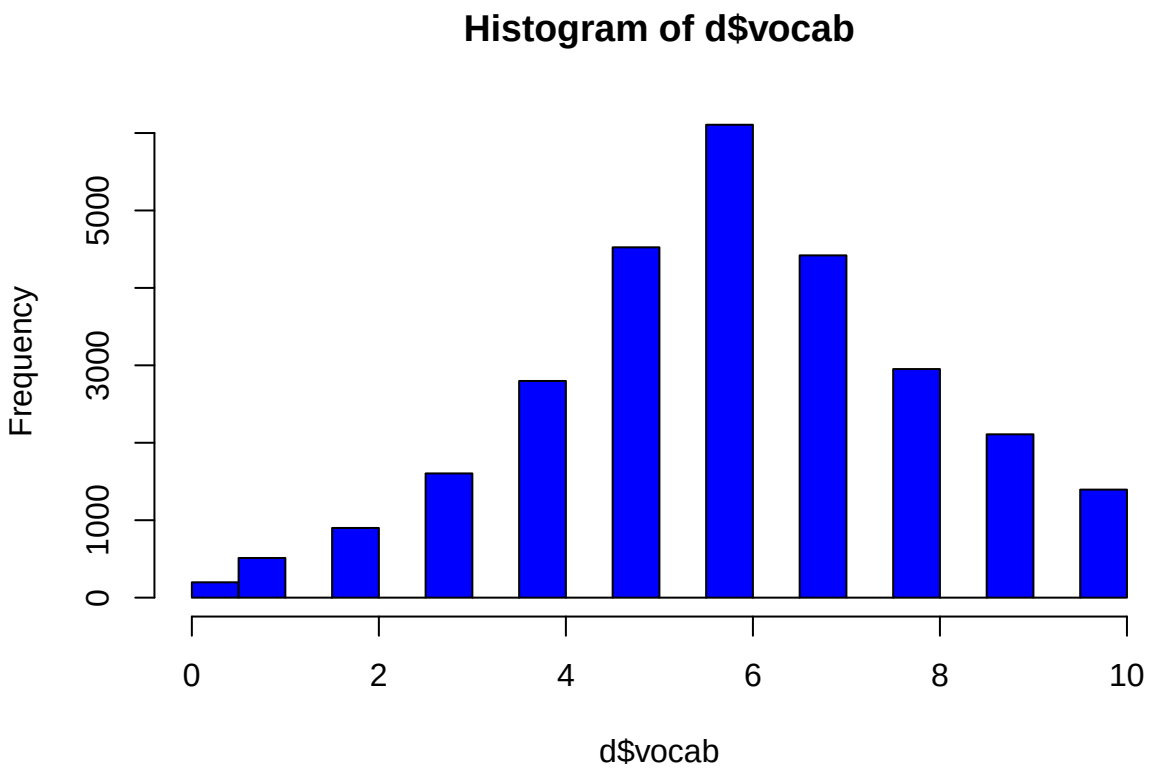
```
##       year          gender      nativeBorn    ageGroup        educGroup
##  1994   : 1977   female:16385   no  : 2556   18-29:5849   <12 yrs  :5924
##  1996   : 1960   male  :12482   yes :26224   30-39:6248   12 yrs   :8612
##  2016   : 1888                  NA's:   87   40-49:5246   13-15 yrs:7182
##  1982   : 1860                               50-59:4329   16 yrs   :3914
##  1987   : 1819                               60+  :7101   >16 yrs  :3154
##  2014   : 1675                               NA's :  94   NA's     :  81
##  (Other):17688
##      vocab             age             educ
##  Min.   : 0.000   Min.   :18.00   Min.   : 0.00
##  1st Qu.: 5.000   1st Qu.:32.00   1st Qu.:12.00
##  Median : 6.000   Median :44.00   Median :12.00
##  Mean   : 5.998   Mean   :46.18   Mean   :13.04
##  3rd Qu.: 7.000   3rd Qu.:59.00   3rd Qu.:15.00
##  Max.   :10.000   Max.   :89.00   Max.   :20.00
##  NA's   :1348     NA's   :94      NA's   :81
```

Let's recode the variable "nativeborn"" to make the output more informative!

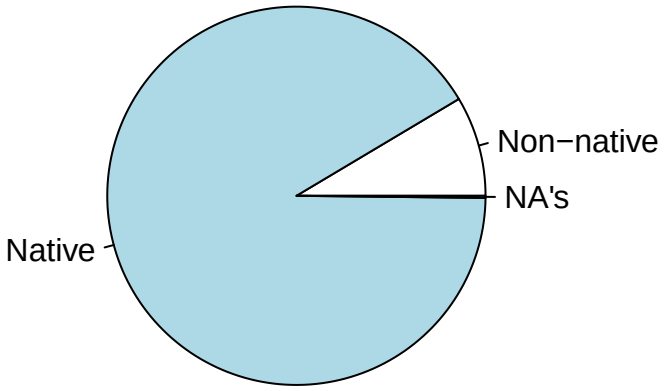
```
##       year          gender           nativeBorn     ageGroup
##  1994   : 1977   female:16385   Non-native: 2556   18-29:5849
##  1996   : 1960   male  :12482   Native    :26224   30-39:6248
```

```
##  2016   : 1888                  NA's      :   87   40-49:5246
##  1982   : 1860                                     50-59:4329
##  1987   : 1819                                     60+  :7101
##  2014   : 1675                                     NA's :  94
##  (Other):17688
##      educGroup        vocab             age             educ
##  <12 yrs  :5924   Min.   : 0.000   Min.   :18.00   Min.   : 0.00
##  12 yrs   :8612   1st Qu.: 5.000   1st Qu.:32.00   1st Qu.:12.00
##  13-15 yrs:7182   Median : 6.000   Median :44.00   Median :12.00
##  16 yrs   :3914   Mean   : 5.998   Mean   :46.18   Mean   :13.04
##  >16 yrs  :3154   3rd Qu.: 7.000   3rd Qu.:59.00   3rd Qu.:15.00
##  NA's     :  81   Max.   :10.000   Max.   :89.00   Max.   :20.00
##                   NA's   :1348     NA's   :94      NA's   :81
```

Plotting in Base R



Pie charts are something people often want to make– though, be careful when using them. They are only good for visualizing big differences among few levels of data. For example, for highlighting how this sample mostly tested native-born individuals...



Plotting with ggplot

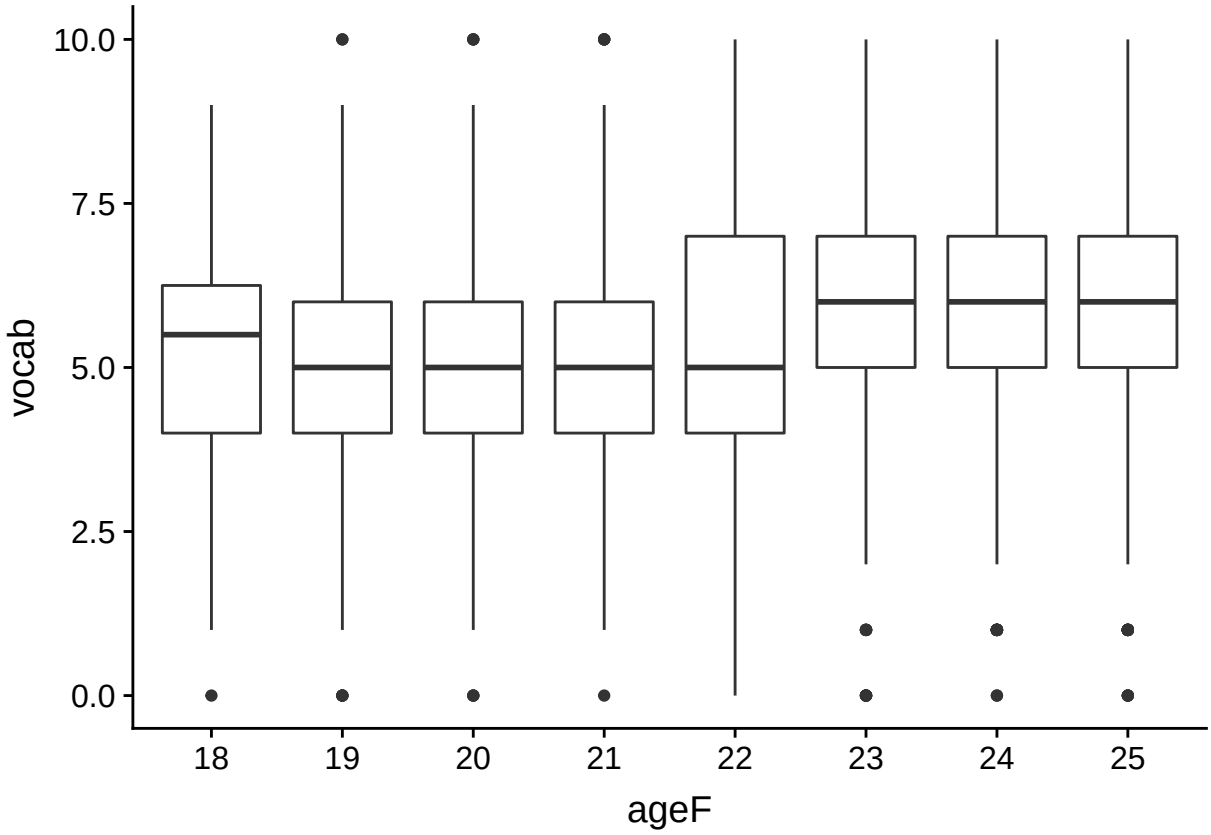
To get a better examination of our data, we can use some more sophisticated plotting code. We will now make plots with ggplot2().

Let's start by looking at how vocabulary changes as people age.

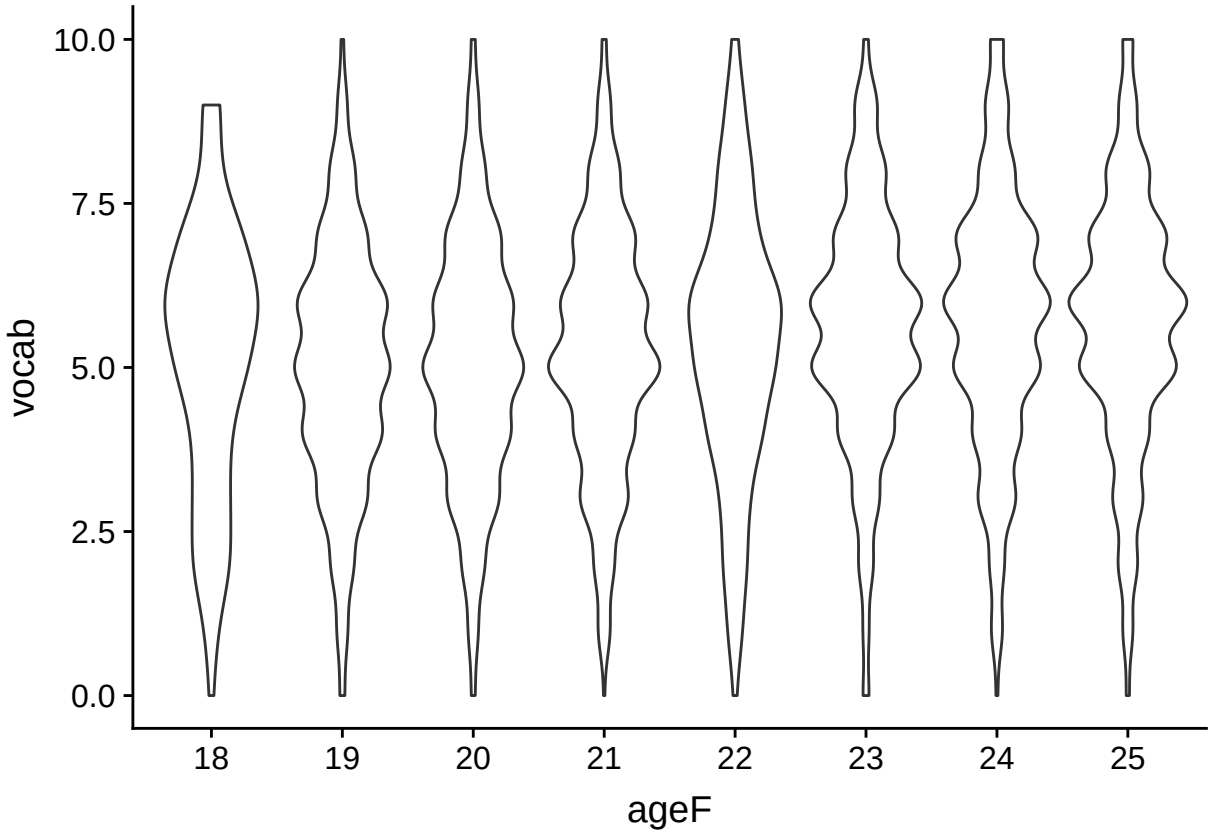
We will start by creating a boxplot using ggplot2() functions.

Basic ggplot syntax:
Start with the command ggplot(), specifying the data to be plotted, and the variables you want to pass to the plot inside the function aes(). These are 'aesthetics'– things that change in the plot, like x values, and y values. (Also: colors, groups of data, shapes, transparency. . .)

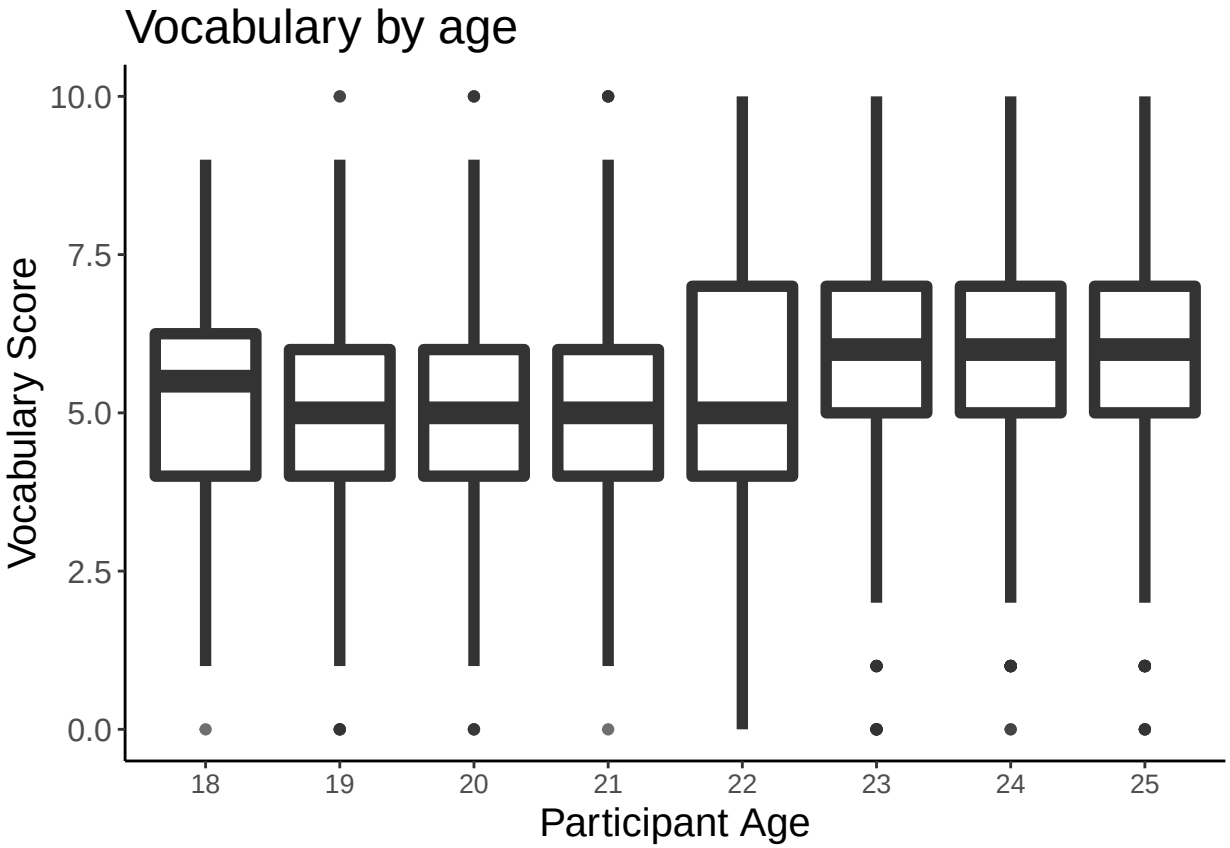
Then, you tell the plot what type of graph to use with a second command– a 'geom'. Here: geom_boxplot.



Here's another way to examine the same data, with violins rather than boxplot. Note that all we changed was the second line in the plot!

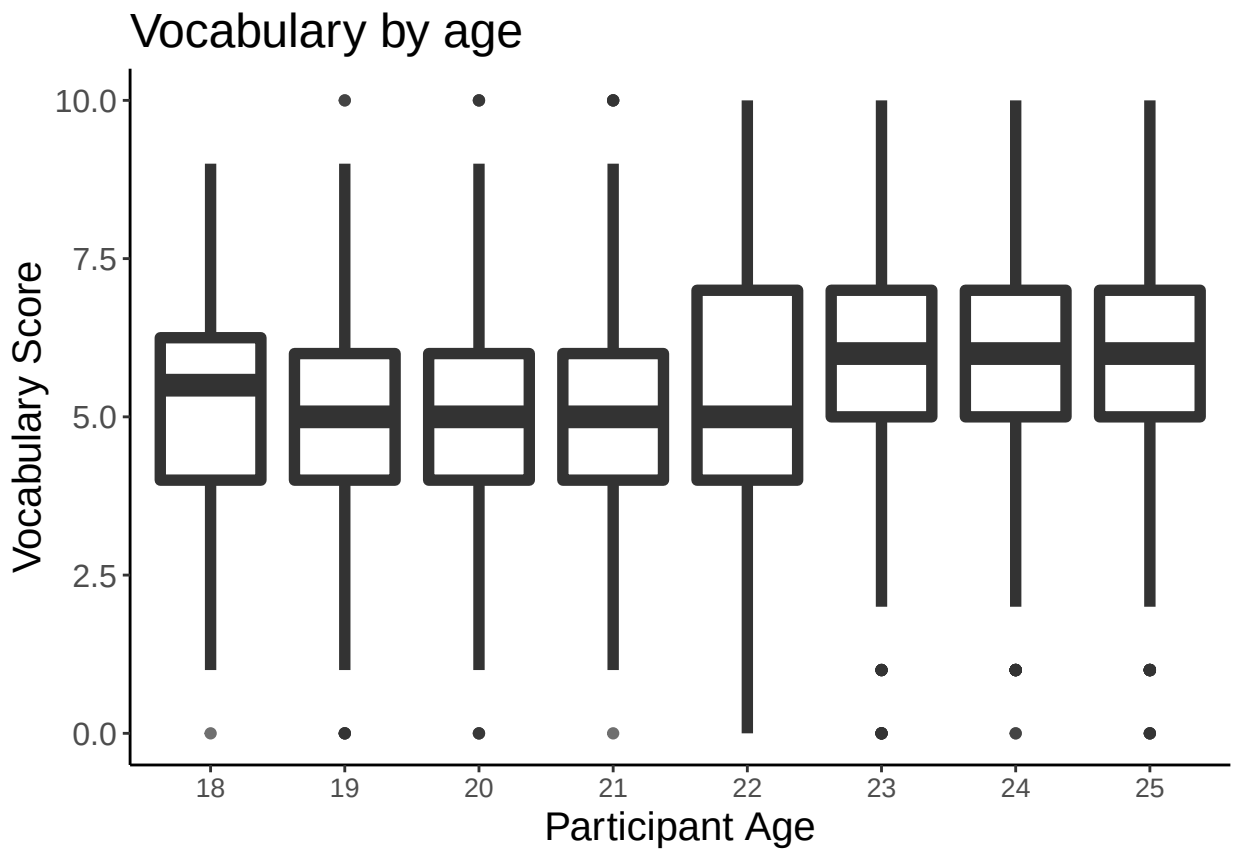


That worked nicely for these data because they are unimodal (= there's only one main bulge in the center of each distribution). Let's go back to boxes. We can pretty them up with some added code:

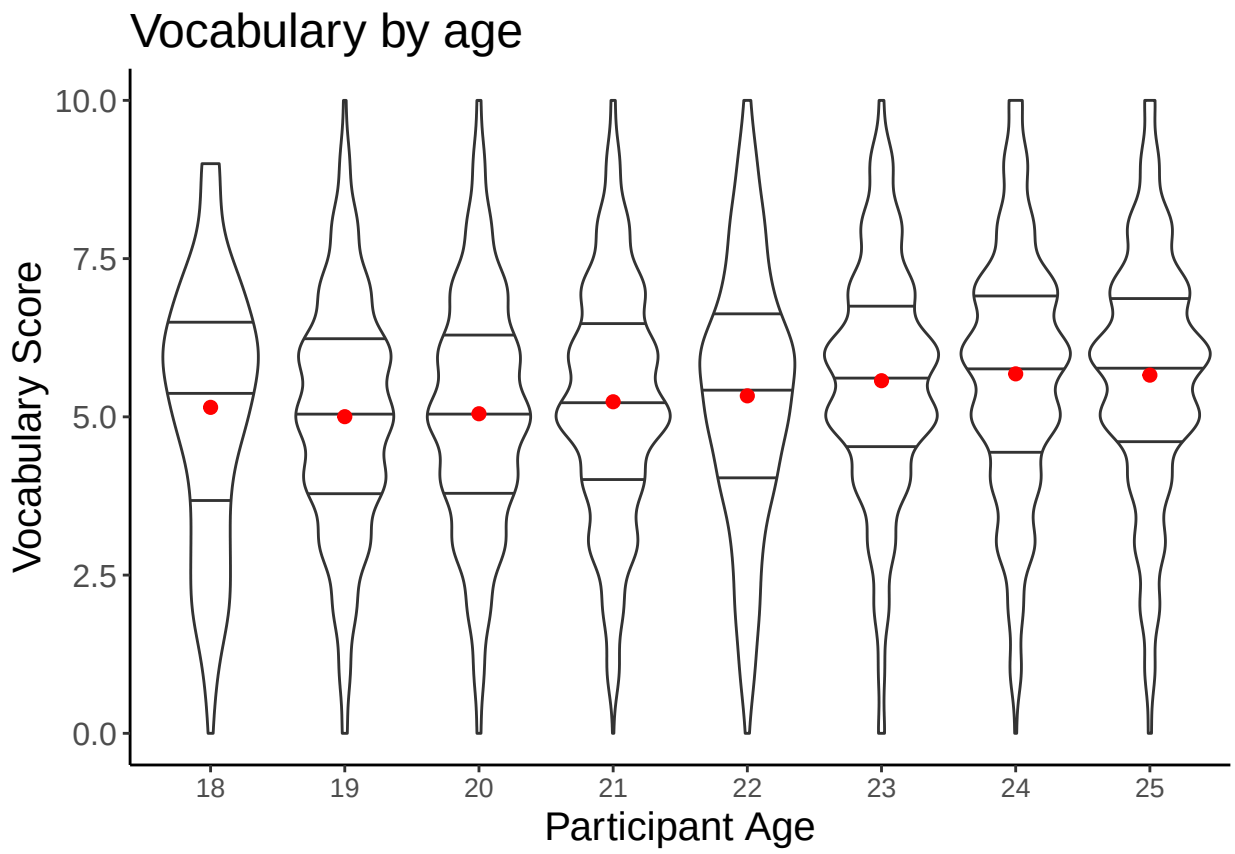


We can also run a code chunk to save parts of a plot, and add to it. I'm saving everything but the geom command to object p.

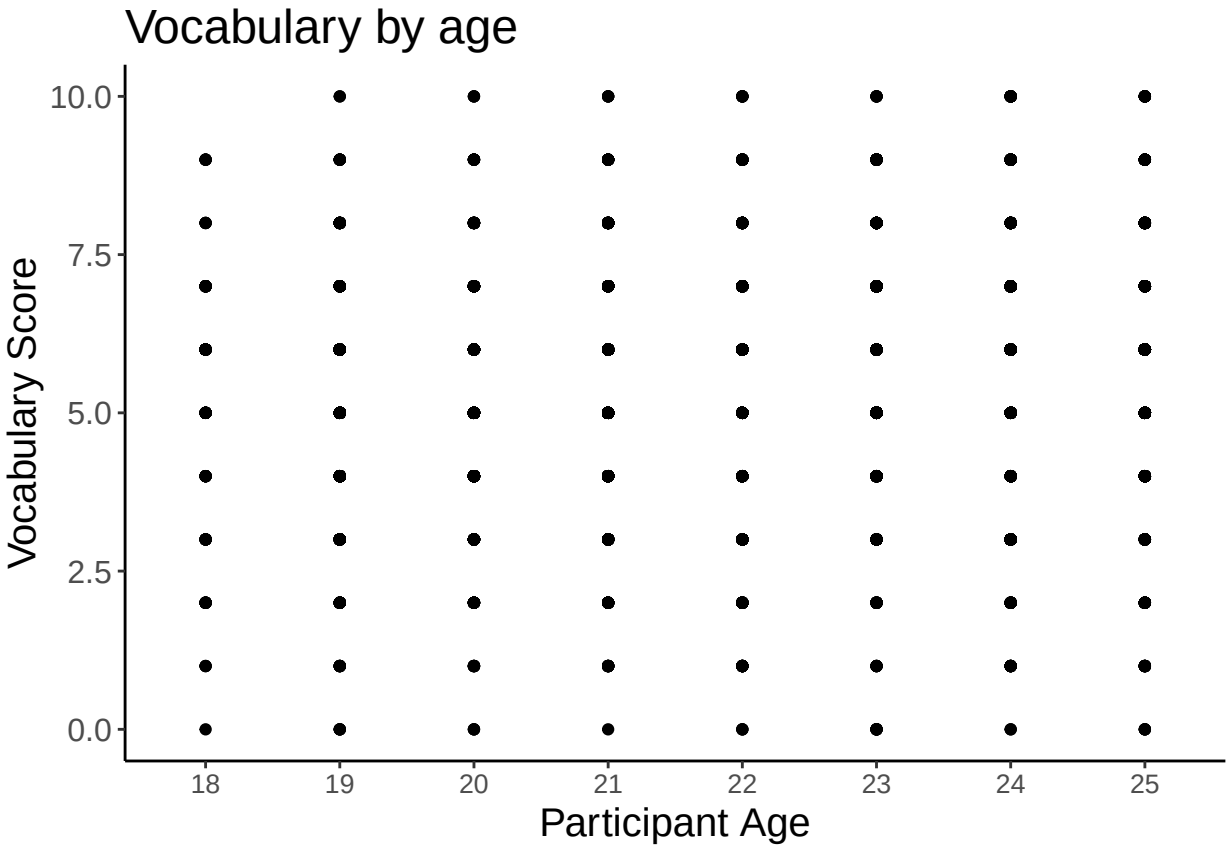
This will now give the same output as we got from plot6!



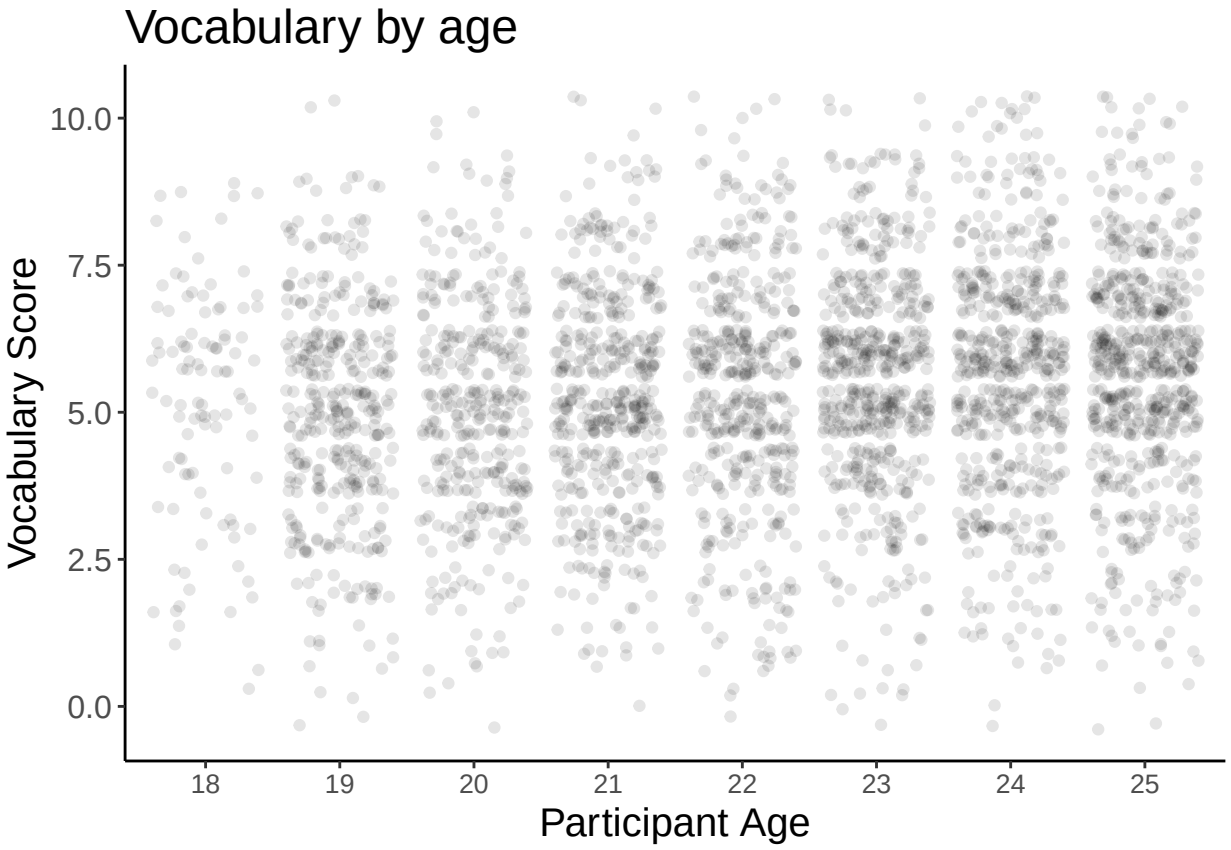
Try with some other geoms:



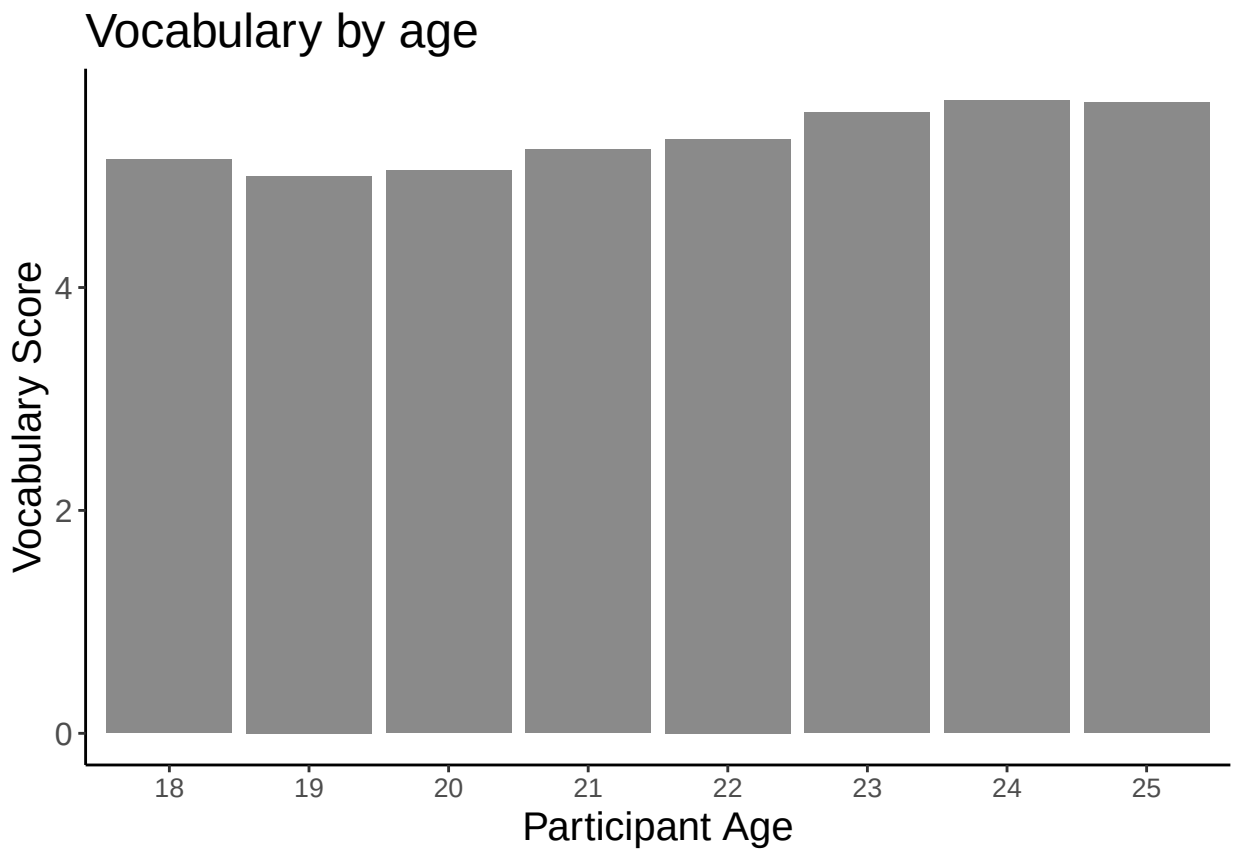
What about looking at the actual individual data points?



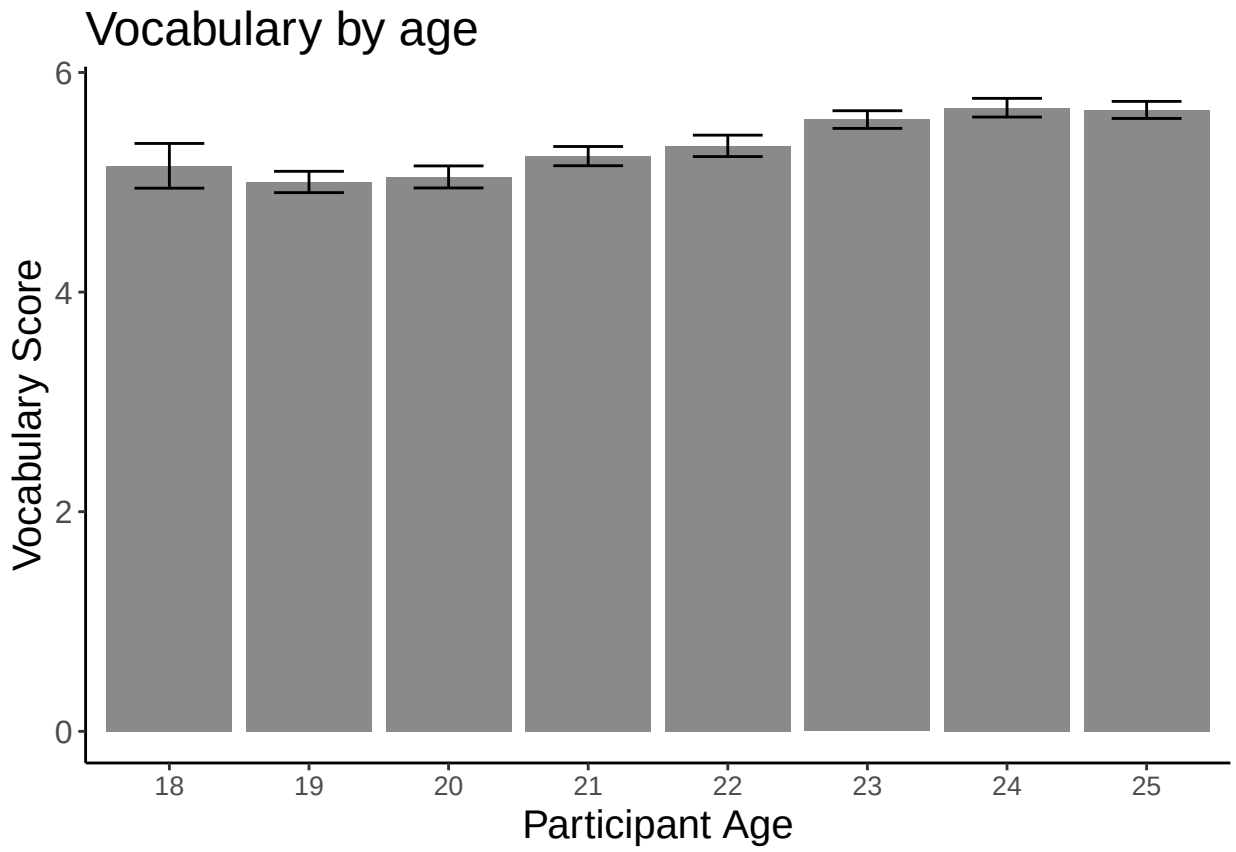
Those points were stacked on top of each other, which is why it looks like there are only a few data points by condition... we can instead 'jitter' them so we can get a sense of where there's more data. We can also make the points slightly transparent using alpha.



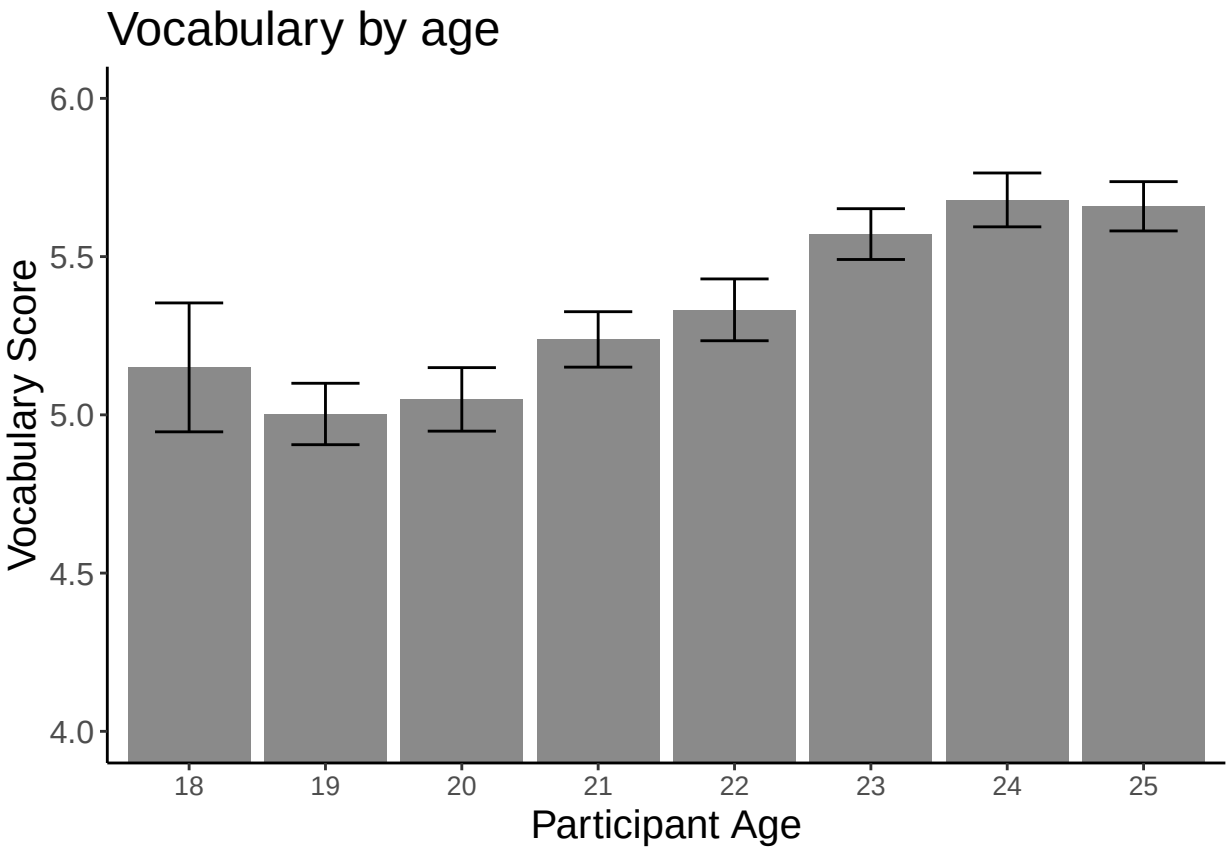
For data where there are lots of points, we might want to plot an aggregate measure. we can use a bar plot for this.



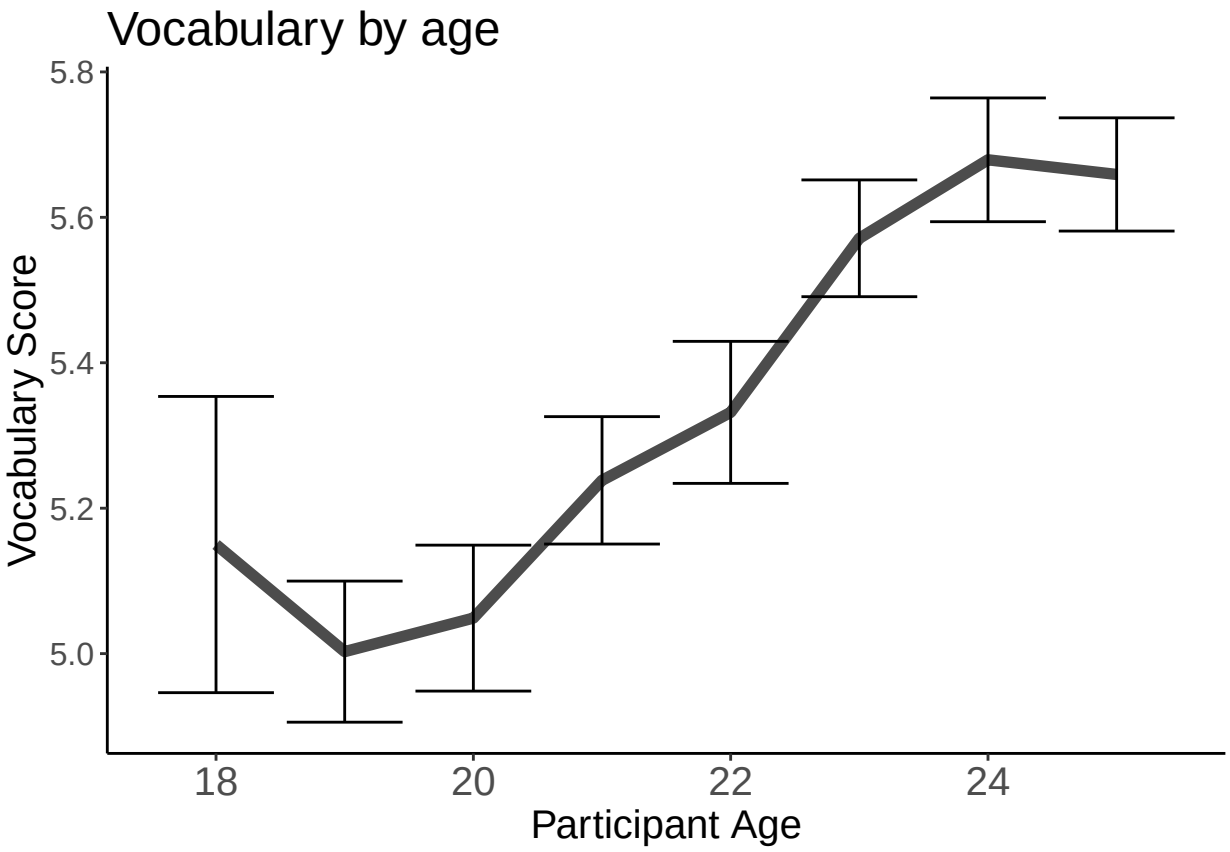
As a rule of thumb, bar plots should *always* show the variability inherent to the data. Otherwise, you have no idea whether differences between bars are reliable.

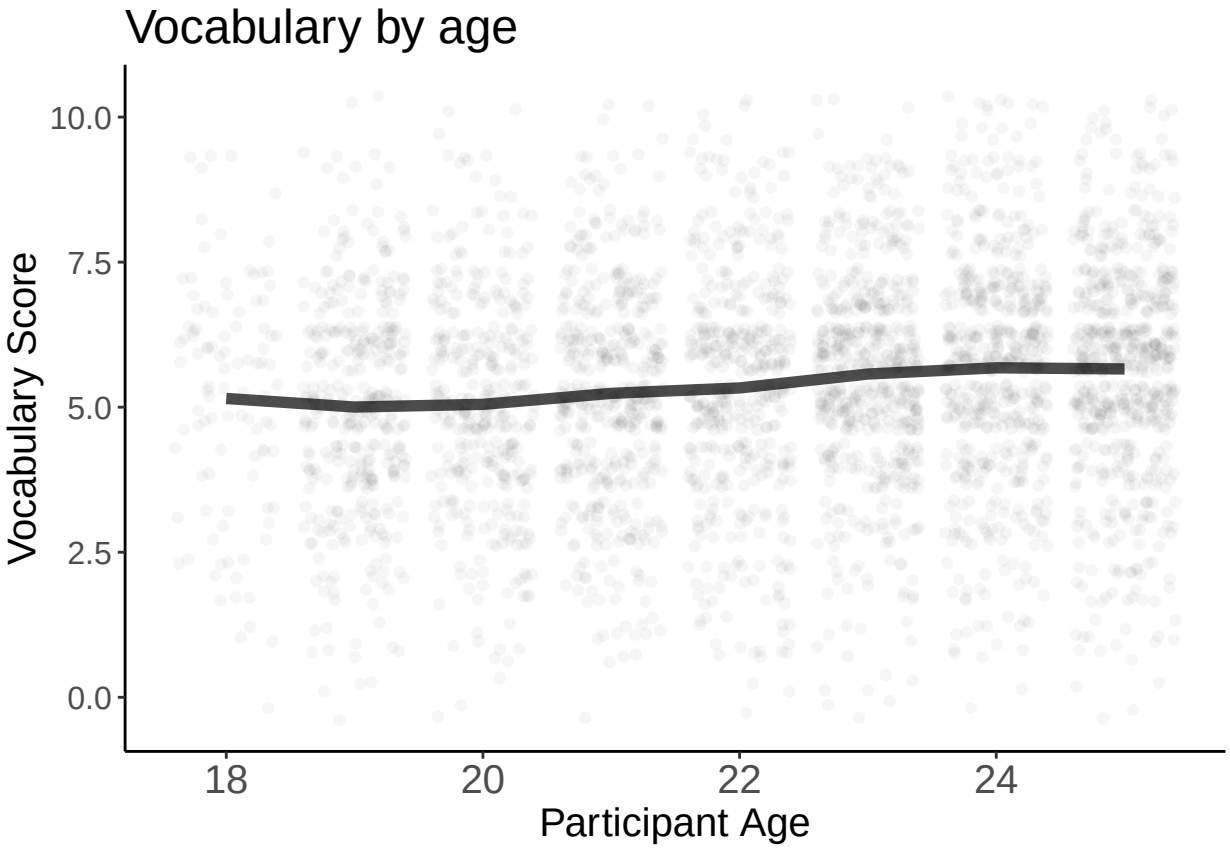


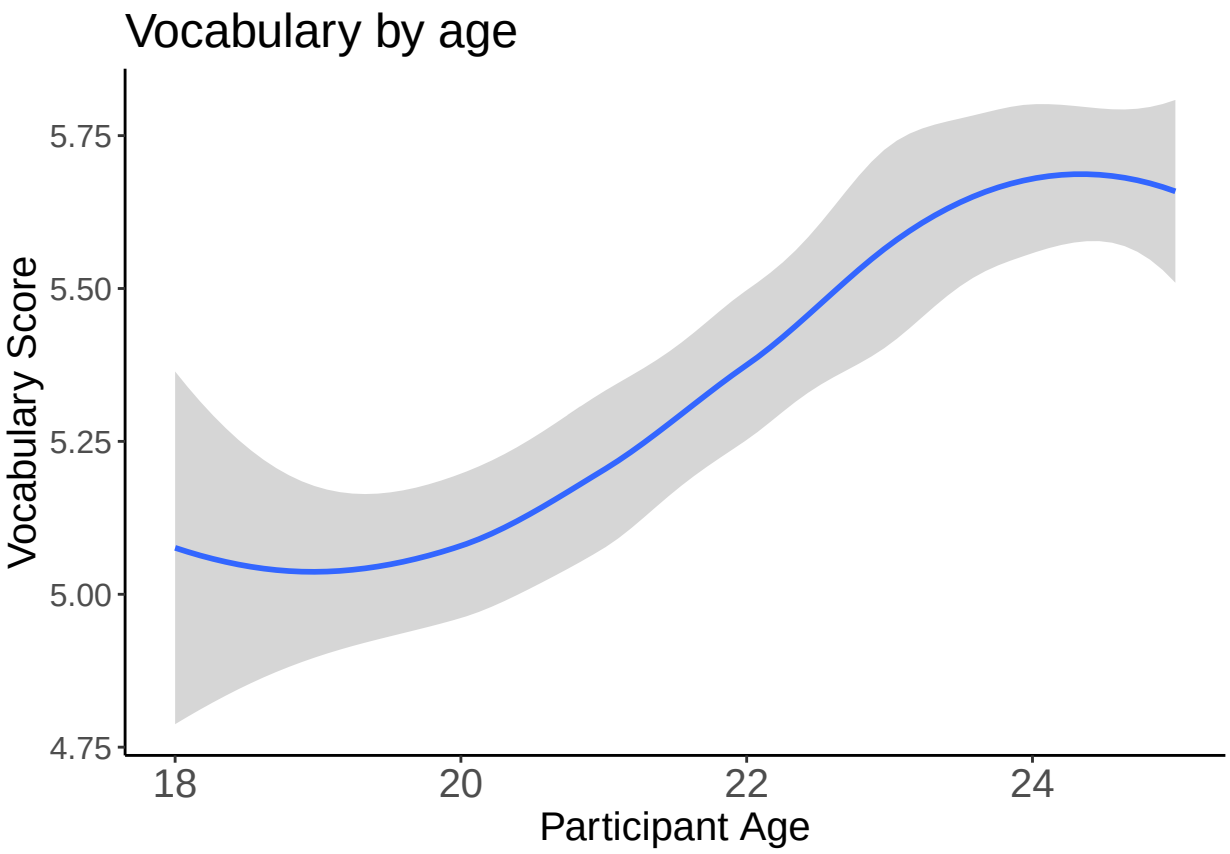
You might also want to 'zoom in' on the differences in the bar plot, to better see them. Mean scores vary between 4 and 6, but it is hard to see because the bar plot starts at 0. Instead, we can zoom in so that the y-axis plots only from 4 to 6.



Or we could make a similar a line plot. This is a little better for describing continuous patterns– like change over time. Line plots should also, as a rule of thumb, provide a sense of variability. You can do with error bars, with smooths, or by just plotting the points in the data.

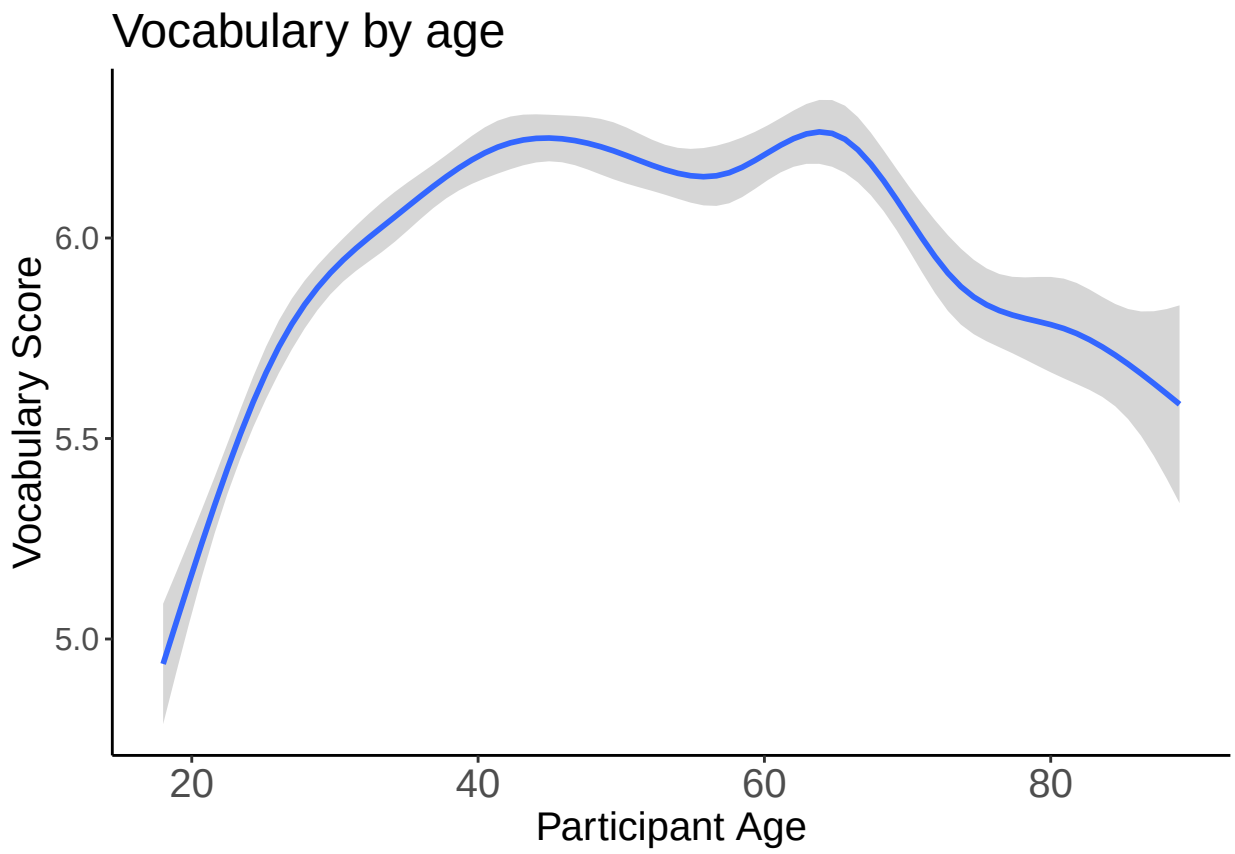

Vocabulary by age
Vocabulary Score
5.8
5.6
5.4
5.2
5.0
18
20
22
24
Participant Age


Vocabulary by age
Vocabulary Score
10.0
7.5
5.0
2.5
0.0
18
20
22
24
Participant Age

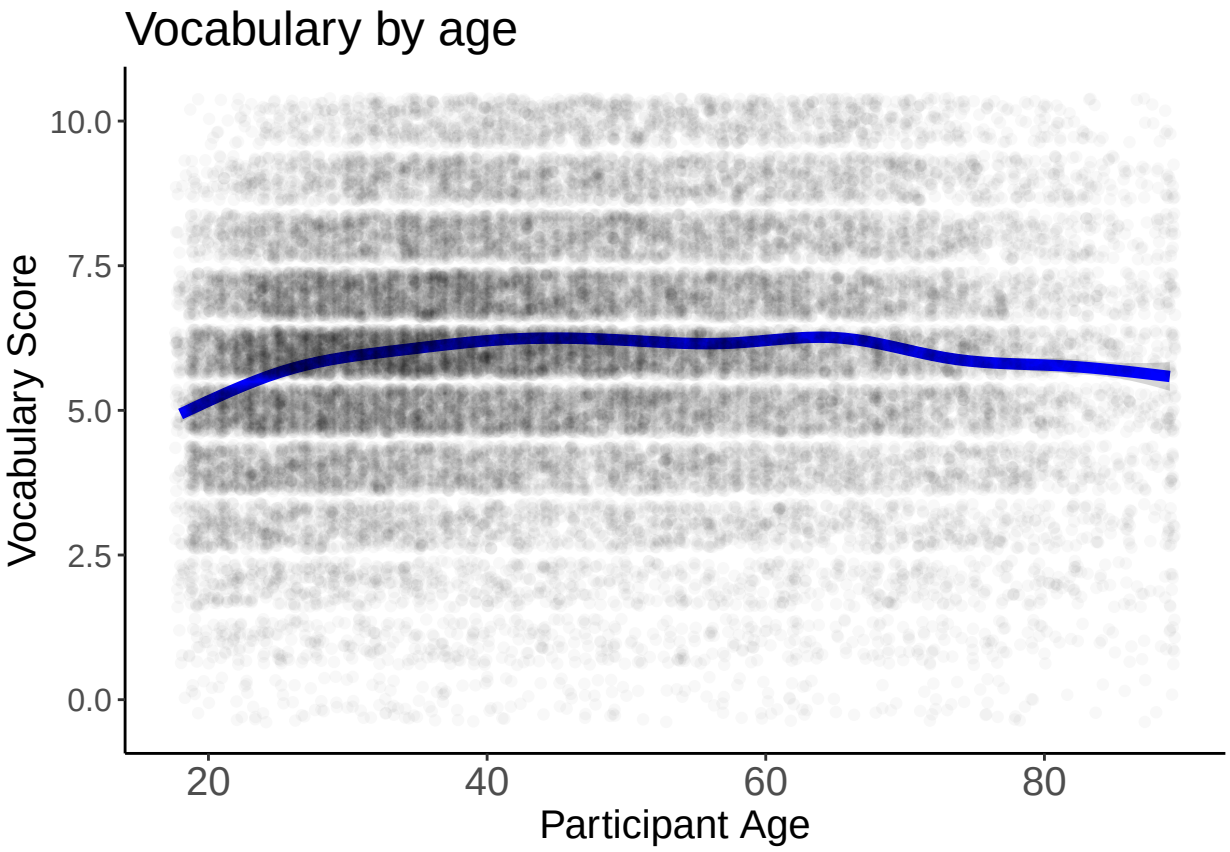


Now, let's look at the trend for the full age range (going back to the original, yet cleaned, dateset d). We can also save the full plot in workspace to call up later.

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

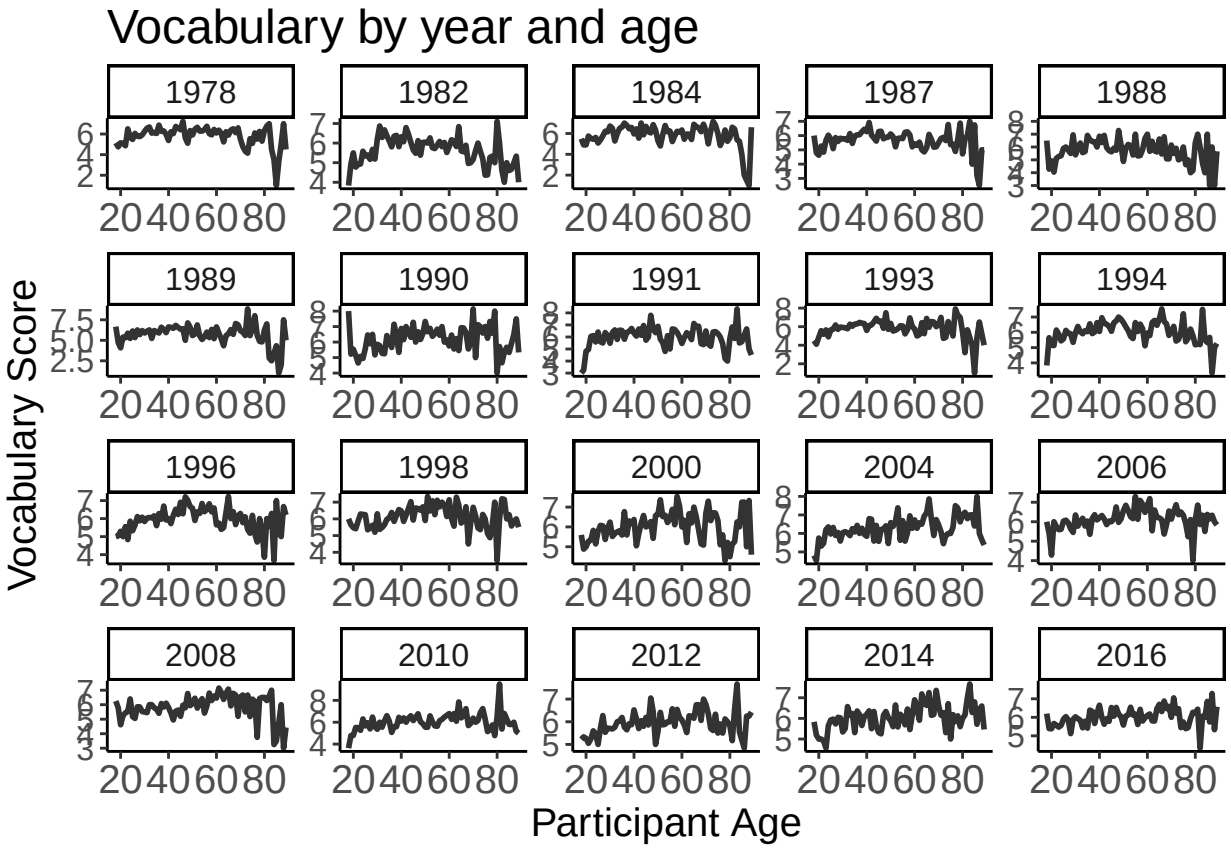



```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



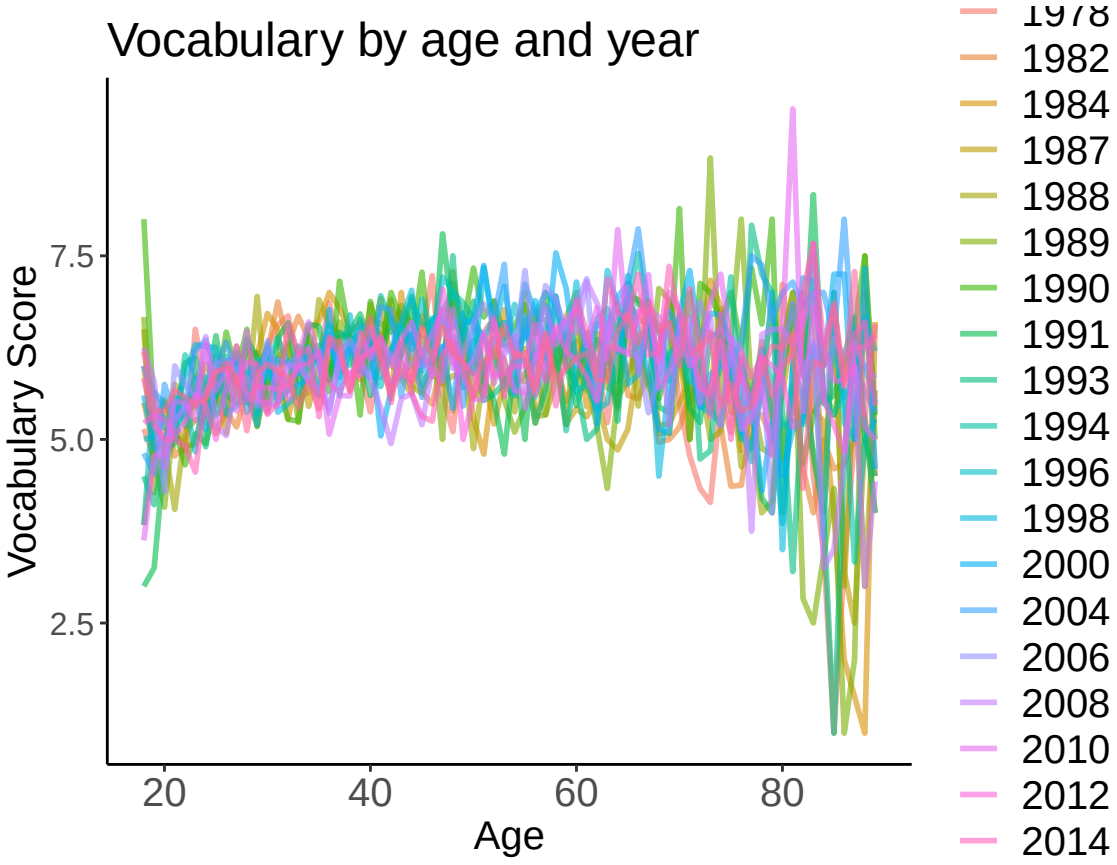

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

We can even examine the relationship between age and vocabulary scores over the years to make sure it's consistent. To make these plots, facet by year. We'll skip the error bars/ smooths this time beacuse the panels also provide some sense of variability, and the plot will get too complex fast with more stuff in it.

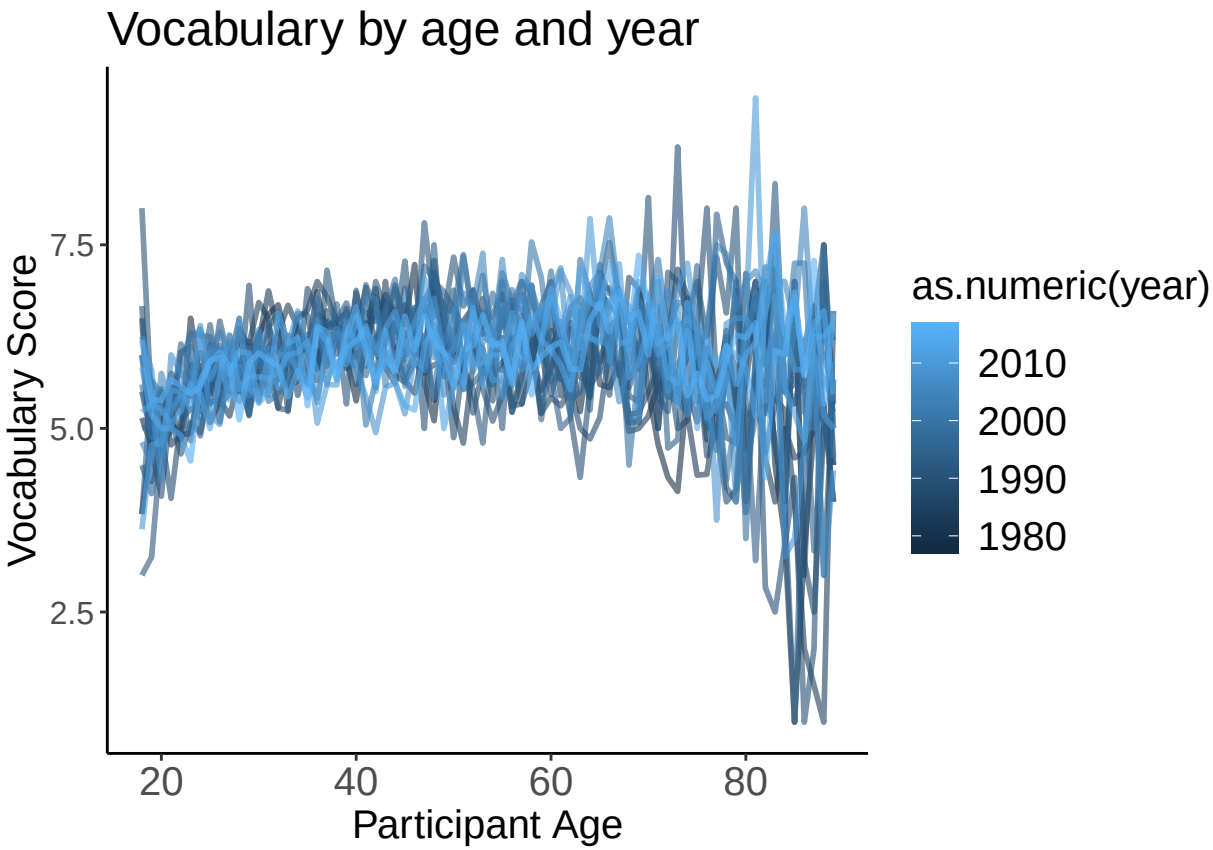


But this is too hard to evaluate properly. It would be better to stack the lines on top of each other and color-code them by year, so we can actually see if there are any meaningful differences. We do this by adding "color=year" to the aes().

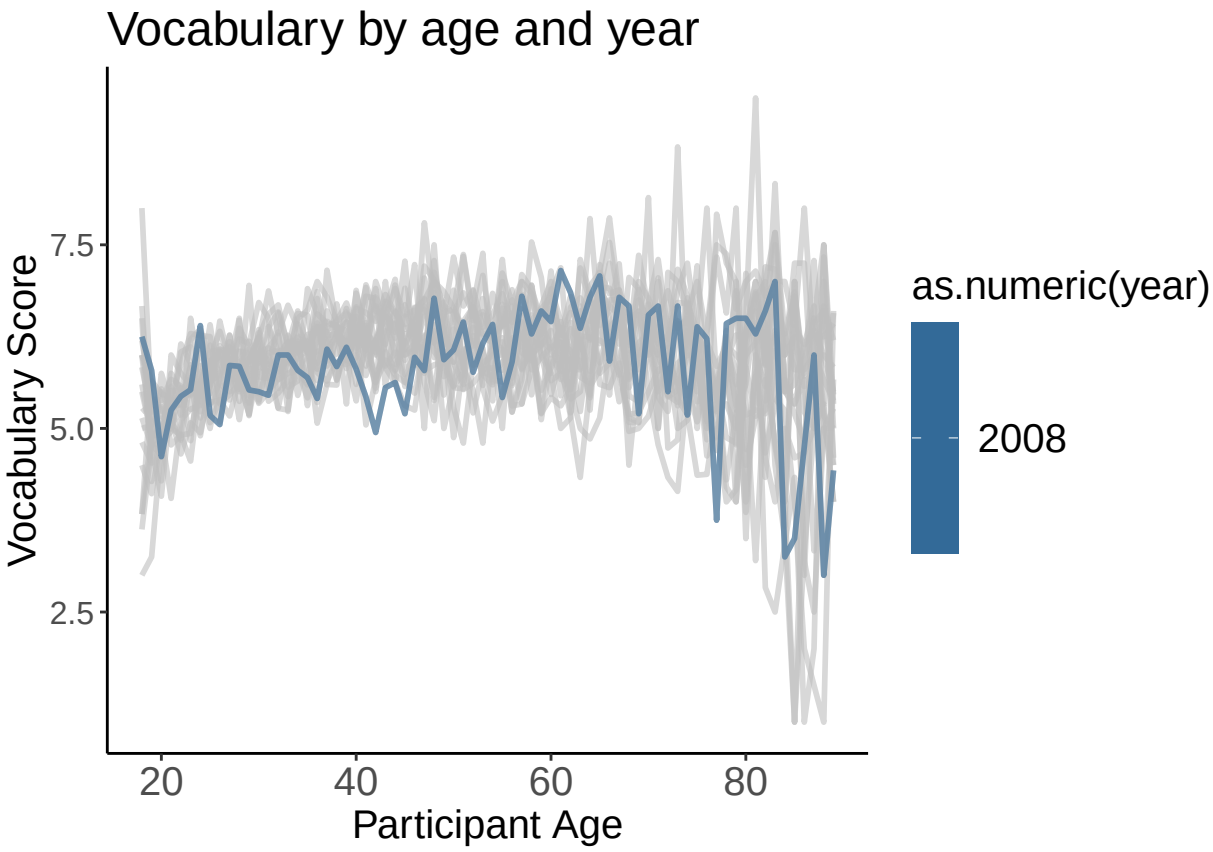
Here, we also add the "group" variable to tell ggplot to give one line per year. You can check to see what happens if you remove this grouping!



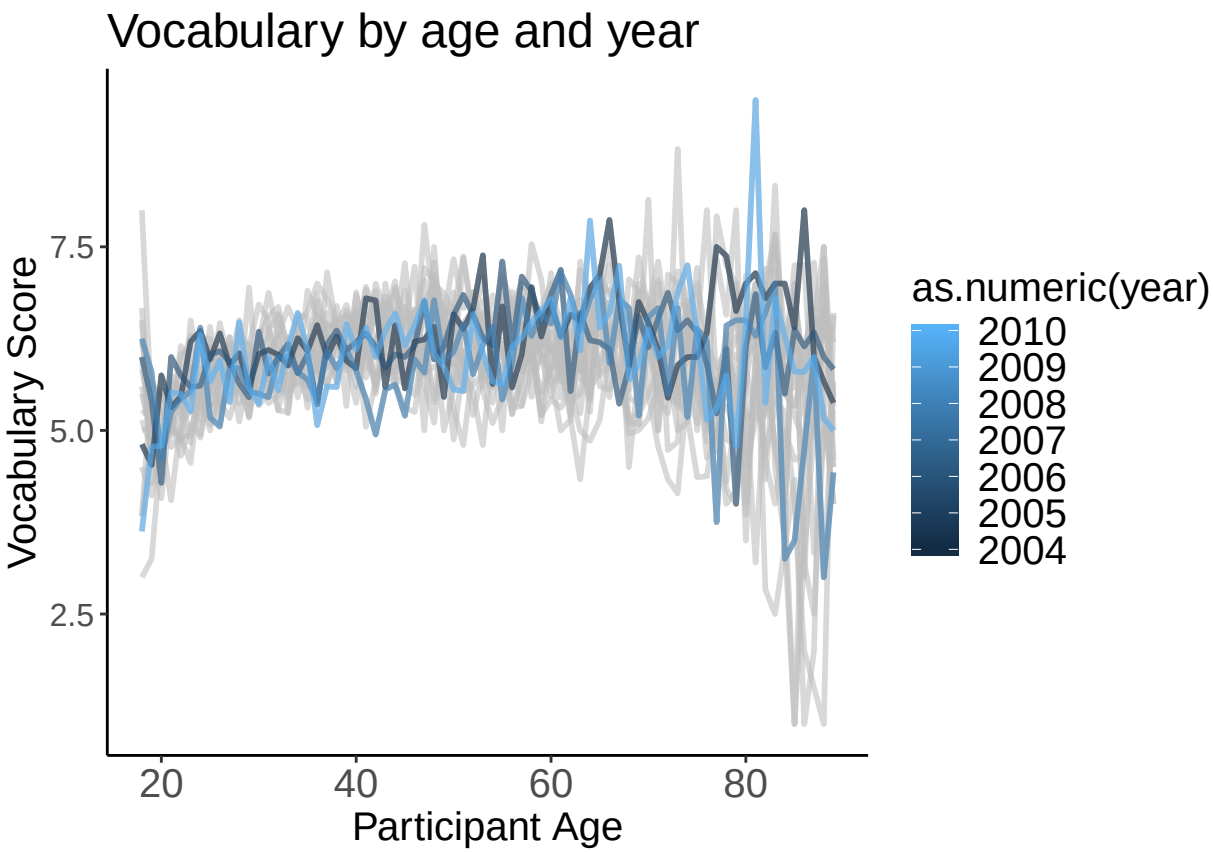
Here, we treated "year" as a categorical value instead of a continuous variable. This means that each year gets a different color. In our case, it's a bit too much (we have 20 different years) and actually quite confusing (because the years are continuous), but this could be useful in other occasions. Instead, we can treat it as numeric, which will give each year a different hue of blue:



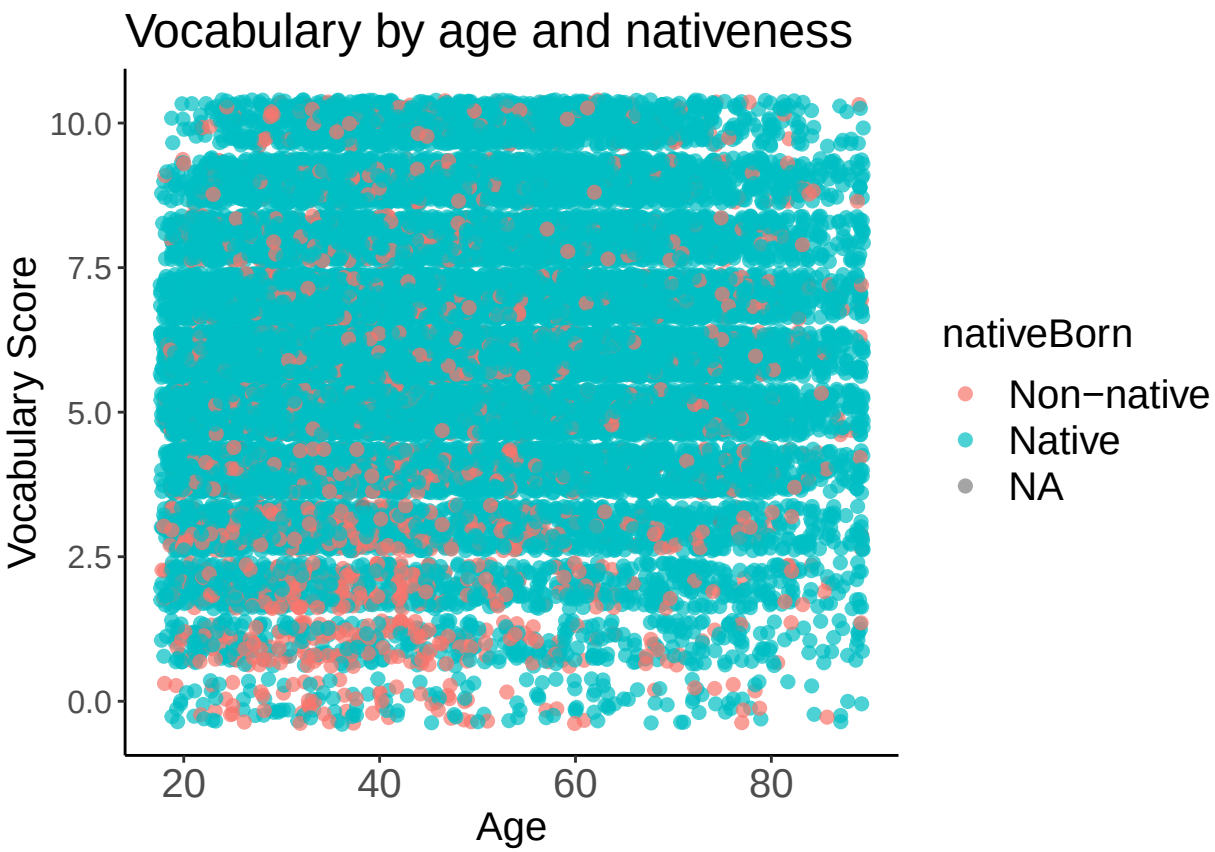
We can also highligh a specific year of interest using gg_highlight:



Or a range of relevant years:

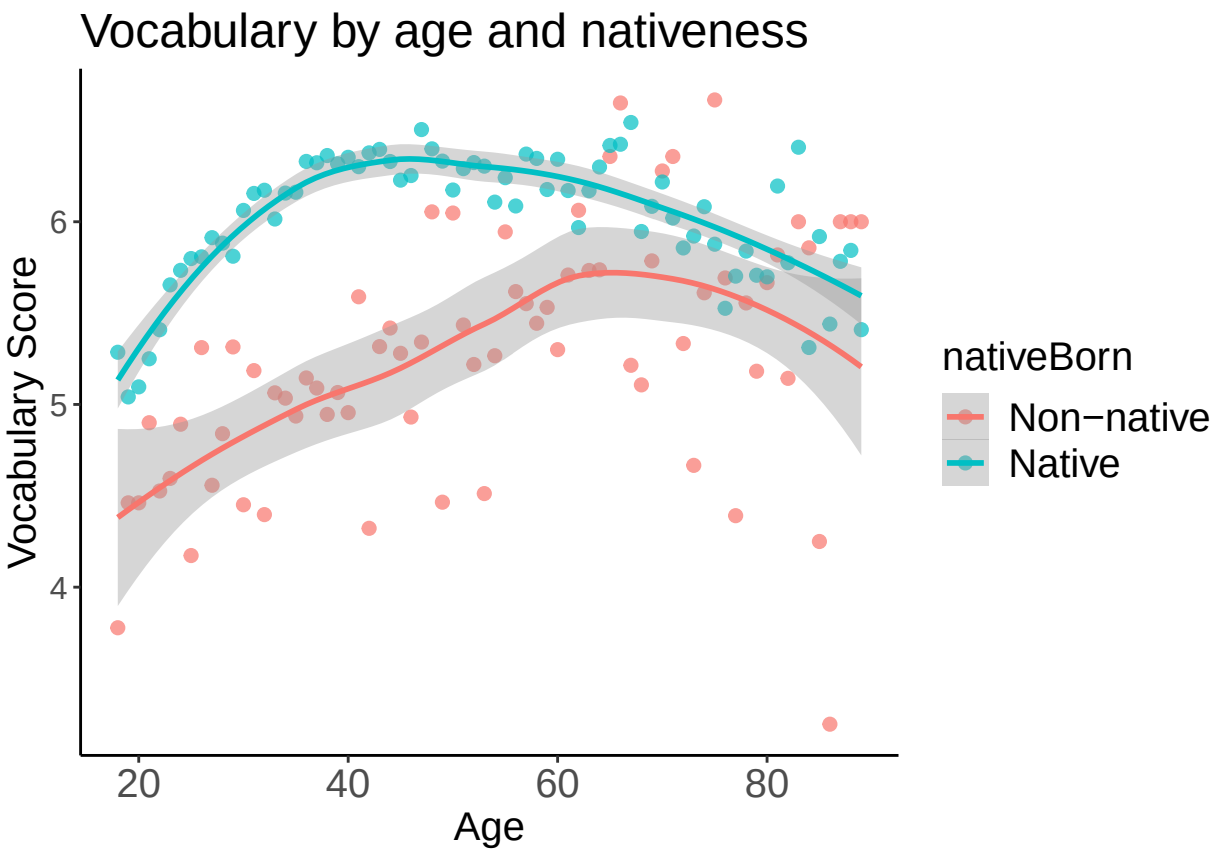


Now, let's continue to examine the relation between age and vocabulary scores in our data set by adding another feature of interest - Nativeness. How much does nativeness matter? Let's check this by adding a different color to the plot based on nativeness.



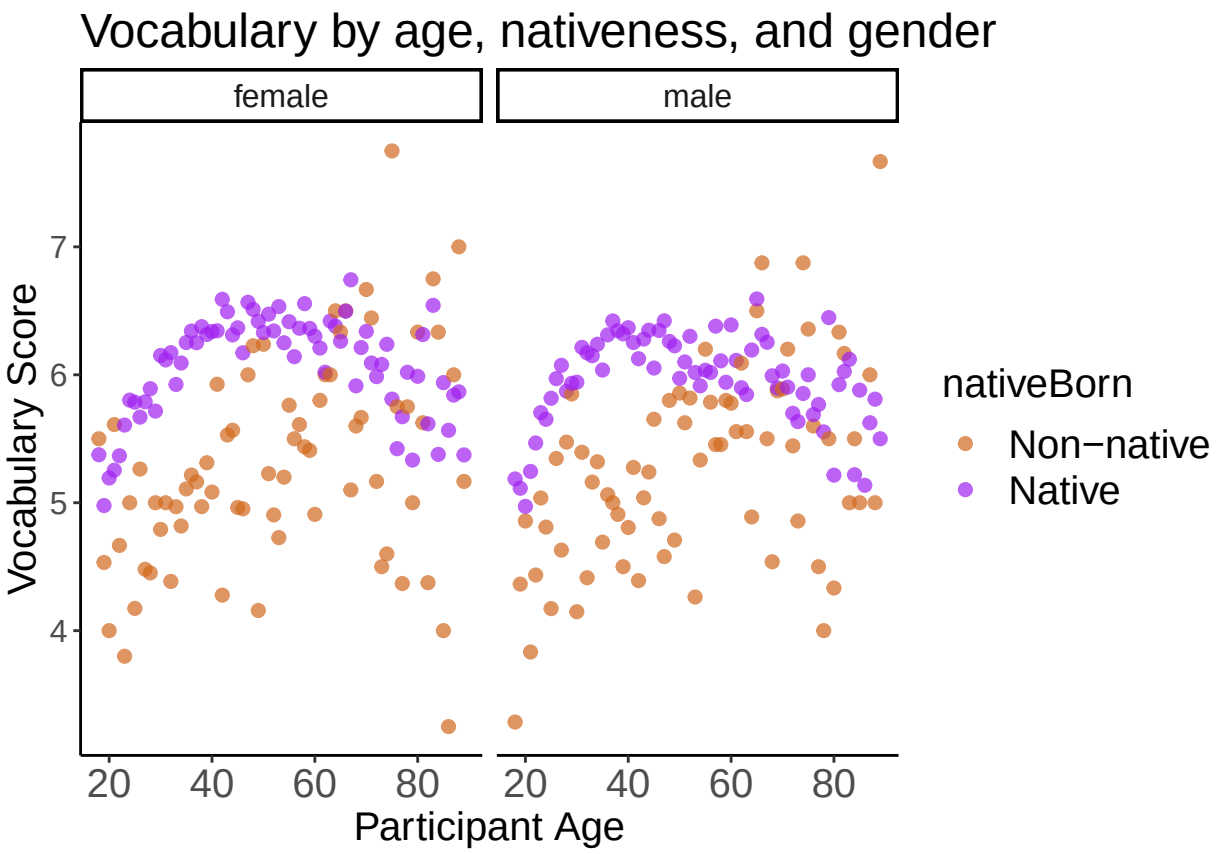
OK, this is obviously not very useful. There are just too many points!

Try summarising the points and looking at the averages.



We also want to look at gender, so we can add that too.

First, we'll summarize the data by gender, age and nativeness. Then, we can make a nice clear plot with faceting. Try facet_grid instead of facet_wrap this time. And also, I want to change the colors:



What do you see? Do nativeness and gender have similar effects?

Time to do some plotting yourself! :)

1. Creat a new r code chunk and give it a descriptive name.
2. Make a new plot of the data d. You could try:

- Summarising the full data across some other variable (using the function ddply())
- Faceting the data in different ways (What happens if you replace the . in facet_grid with a variable?)
- Combining multiple geoms! (What happens if you add geom_smooth on top of plot 21?)
- Changing the colors to make certain contrasts pop out (what if you swap the order of the colors in plot 21?)
- Changing the transparency, size of points, or size of text.
- Changing the shape of points– what do you think the function for changing the shape of points should be? (See the legend below for help.)

Combining Plots

You can also combine different plots to one grid using the "cowplot" package. For that, you'll need the plots you named and saved into your workspace:

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

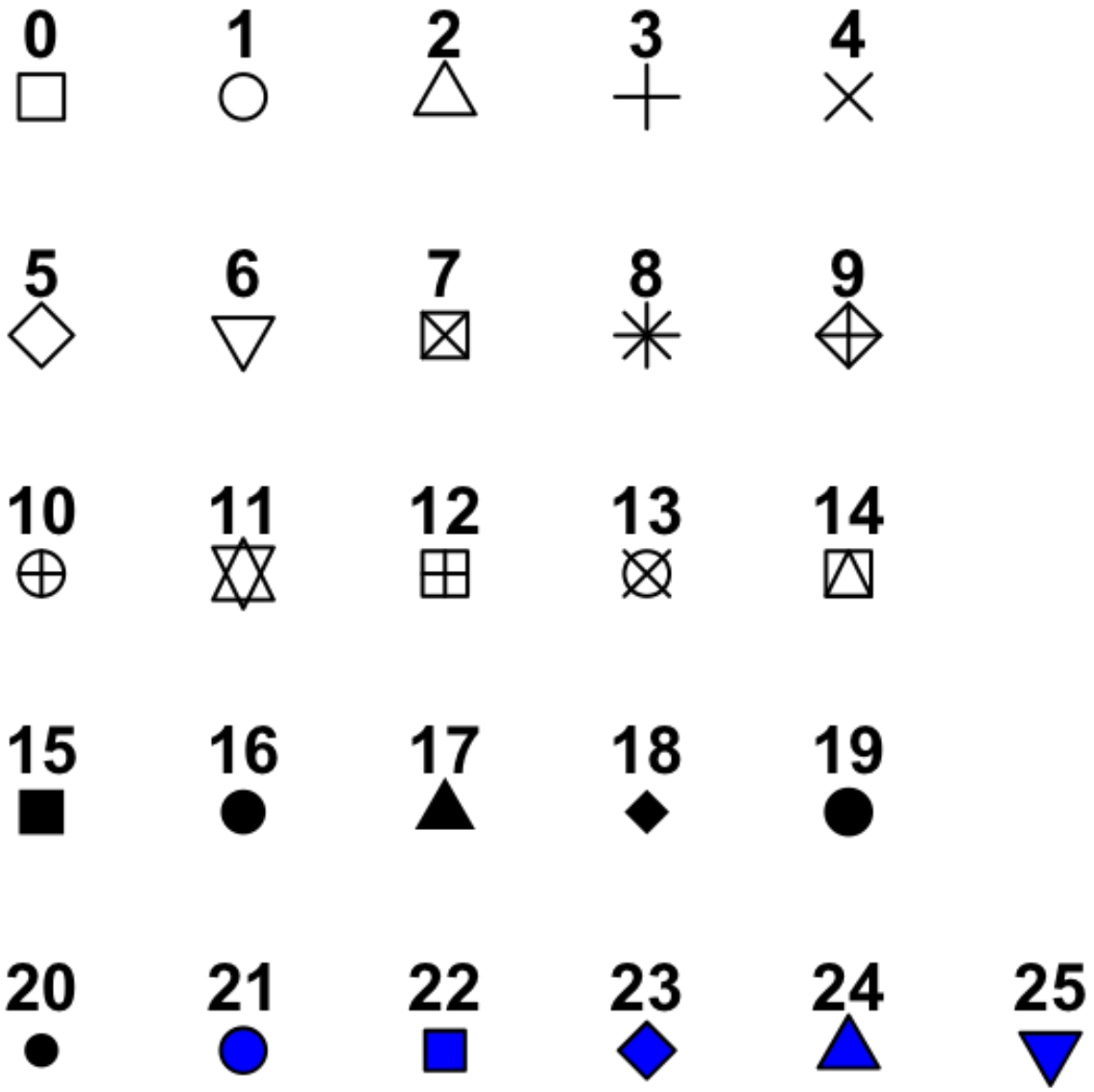
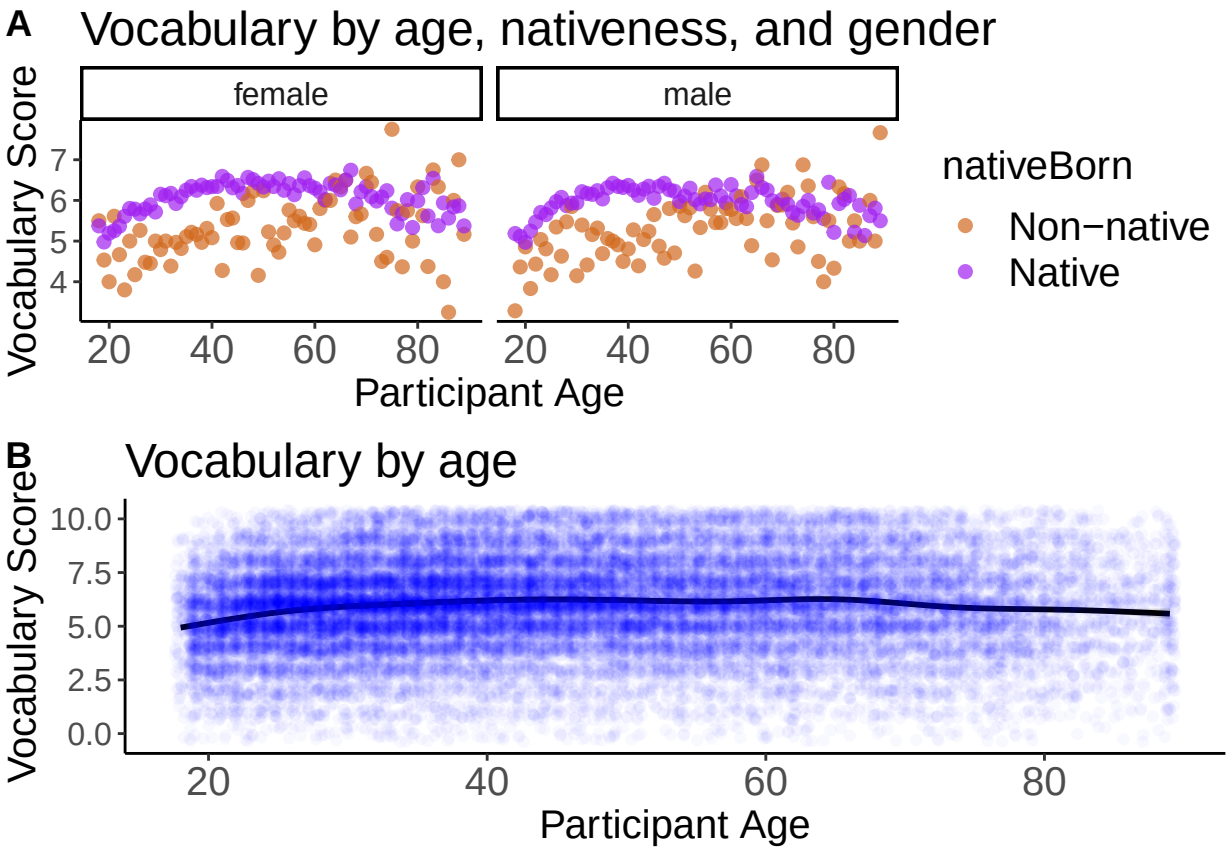


Figure 1:



Animating Plots

If you have a line plot where the x-axis representes some change over time (e.g., with age, with year, with testing blocks), you could try to animate it. This is really cool and useful for presentations!

For this, we'll use the tweenr and gg_animate packages.

Analysis time!

Let's dig deeper into the relationship between education, age, and native language background. We will model this with a mixed-effects regression.

First, let's prepare our variables by centering the continuous ones and setting up the contrasts– specifying how the regression model deals with categorical predictors. This causes the model to make the comparisons we want it to make.

Now, let's make a model based on our predictions. We will run an (overly) simple model in terms of the random effects (the 1|year part) because it can take a while to figure this out. We'll also use the rule-of-thumb that t-value > 2 means a significant effect.

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: vocab ~ c.age * c.educ * nativeBorn * gender + (1 | year)
##    Data: d
##
## REML criterion at convergence: 110053.2
##
```

```
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7601 -0.6206  0.0319  0.6741  4.8070
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  year     (Intercept) 0.02289  0.1513
##  Residual             3.24812  1.8023
## Number of obs: 27360, groups:  year, 20
##
## Fixed effects:
##                                  Estimate Std. Error t value
## (Intercept)                     6.1368043  0.0372267 164.850
## c.age                           0.0172005  0.0008817  19.509
## c.educ                          0.3672179  0.0057595  63.759
## nativeBorn1                    -0.8160241  0.0533981 -15.282
## gender1                        -0.1535117  0.0232579  -6.600
## c.age:c.educ                    0.0006742  0.0003081   2.188
## c.age:nativeBorn1               0.0060011  0.0032919   1.823
## c.educ:nativeBorn1             -0.1099280  0.0143827  -7.643
## c.age:gender1                  -0.0073209  0.0013559  -5.399
## c.educ:gender1                  0.0092342  0.0083035   1.112
## nativeBorn1:gender1             0.0619784  0.0796482   0.778
## c.age:c.educ:nativeBorn1       -0.0008238  0.0008849  -0.931
## c.age:c.educ:gender1           -0.0016587  0.0004564  -3.635
## c.age:nativeBorn1:gender1       0.0100654  0.0049119   2.049
## c.educ:nativeBorn1:gender1     -0.0323328  0.0206105  -1.569
## c.age:c.educ:nativeBorn1:gender1 -0.0003474  0.0012515  -0.278
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

To make this appear as pretty table, we can use the "kable" function.

Table 1: Vocabulary score by age, education, nativeness and gender

	Estimate	Std.Error	t-value
(Intercept)	6.136804	0.037227	164.849728
Age	0.017200	0.000882	19.509072
Years of Education	0.367218	0.005759	63.758943
Nativeness (Non-native vs. Native)	-0.816024	0.053398	-15.281901
Gender (Male vs. Female)	-0.153512	0.023258	-6.600402
Age X Education	0.000674	0.000308	2.188081
Age X Nativeness	0.006001	0.003292	1.822971
Education X Nativeness	-0.109928	0.014383	-7.643080
Age X Gender	-0.007321	0.001356	-5.399183
Education X Gender	0.009234	0.008303	1.112089
Nativeness X Gender	0.061978	0.079648	0.778152
Age X Education X Nativeness	-0.000824	0.000885	-0.930911
Age X Education X Gender	-0.001659	0.000456	-3.634637
Age X Nativeness X Gender	0.010065	0.004912	2.049208
Education X Nativeness X Gender	-0.032333	0.020611	-1.568752
Age X Education X Nativeness X Gender	-0.000347	0.001251	-0.277588

	Estimate	Std.Error	t-value

But what's actually going on?

It's fairly easy to understand the main effects of education, age, gender and nativeness:

- Age is a significant positive predictor of vocab scores (higher age = higher score)
- Nativeness is a significant negative predictor of vocab scores (non native = lower score than native)
- Gender is a significant negative predictor of vocab scores (males = lower score than females)
- Education is a significant positive predictor of vocab scores (higher education = higher score)

But when it comes to the interactions, it's harder to interpret.

The sign of the interactions tells us about our effects.

For example:

- the effect of age is positive (higher age = higher score)
- the effect of gender is negative (males = lower score)
- the interaction between age and gender is negative (–> the positive effect of age on vocab score is smaller for males)

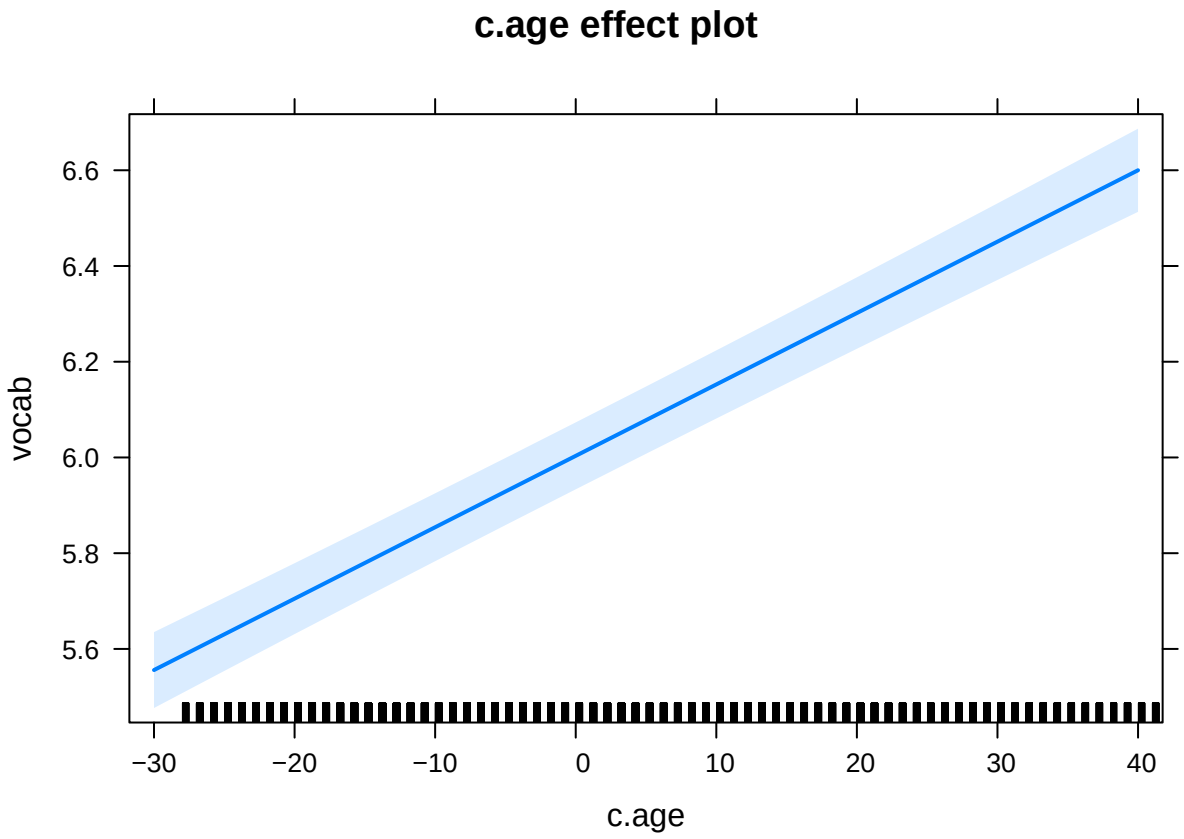
But once we get to the triple interaction of age, education, and gender?? This gets hard to understand by just looking at the signs...

So - we can plot the model using the "effects" package! :)

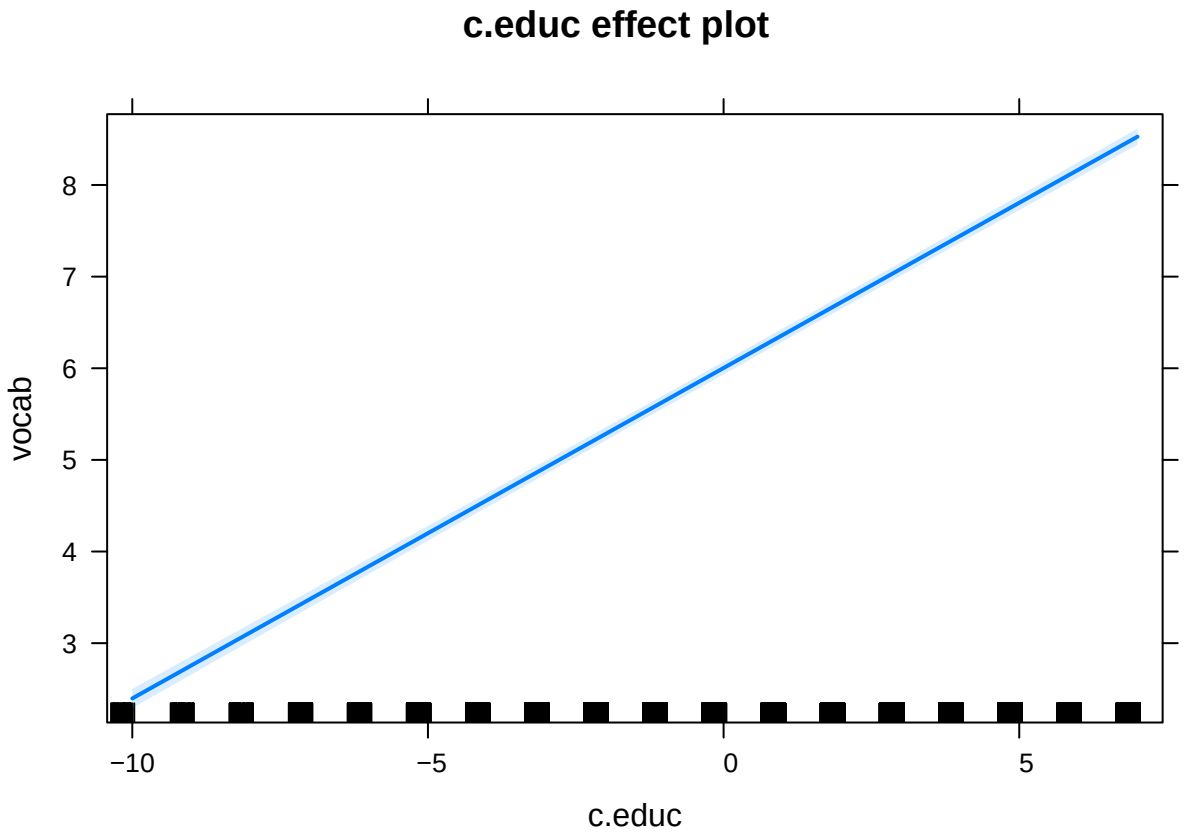
Plotting the model

First, let's confirm that we understood the main effects:

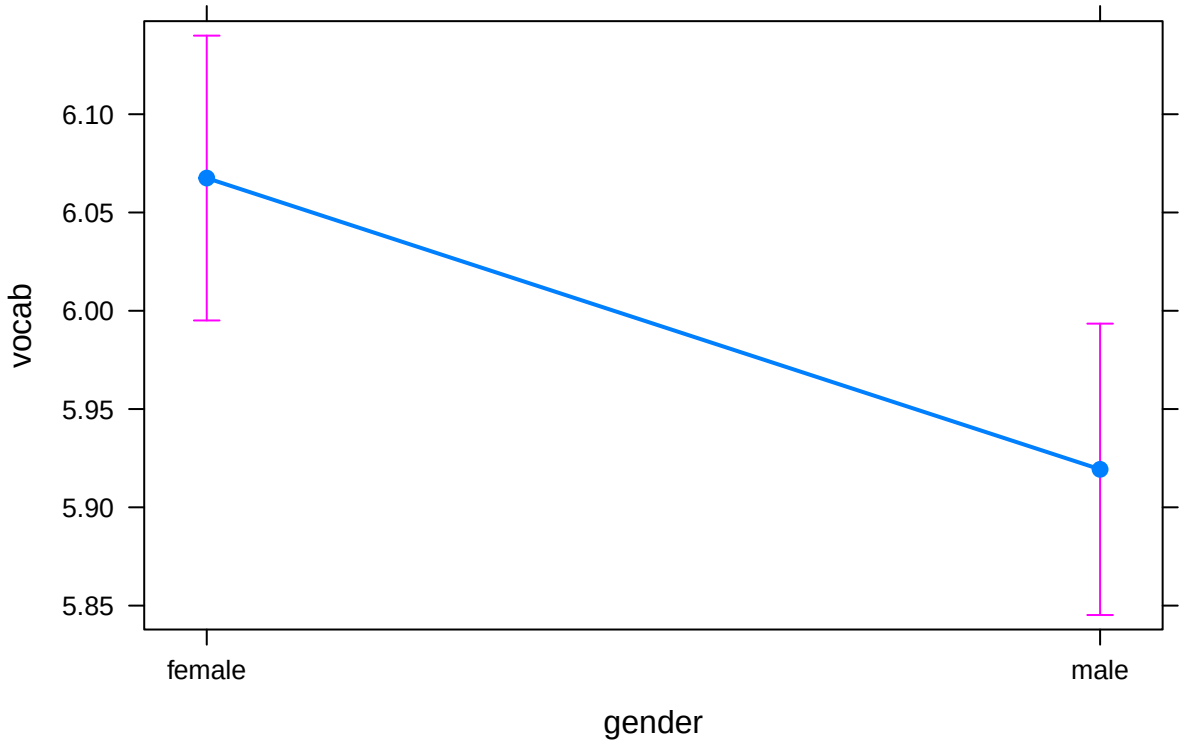
```
## NOTE: c.age is not a high-order term in the model
```



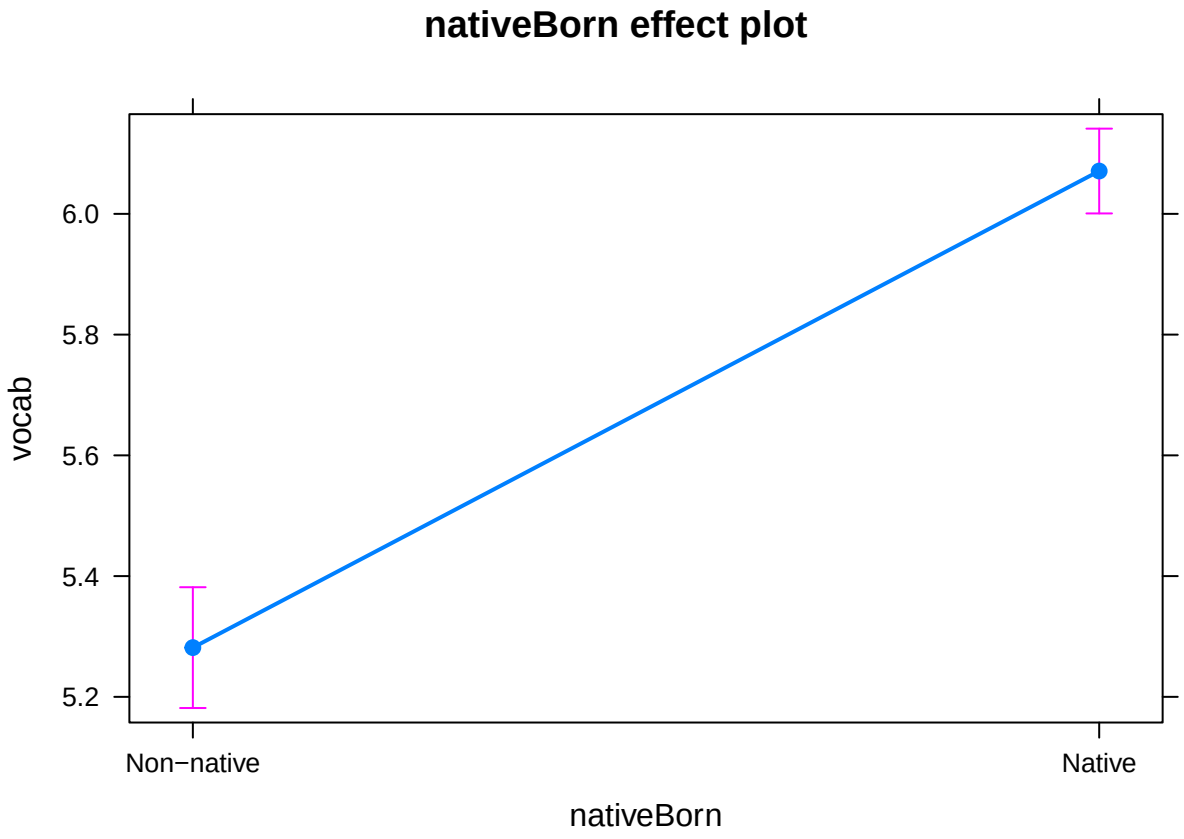

```
## NOTE: c.educ is not a high-order term in the model
```




```
## NOTE: gender is not a high-order term in the model
```



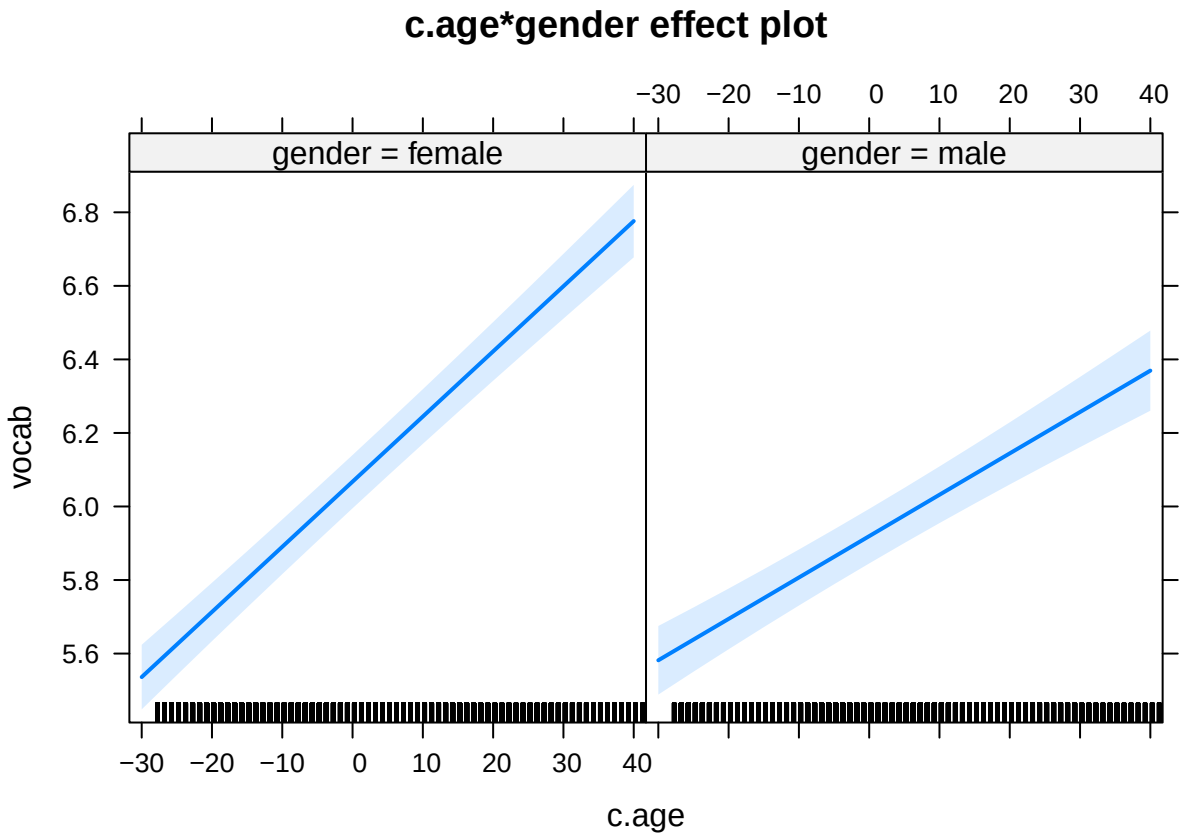

```
## NOTE: nativeBorn is not a high-order term in the model
```



Now let's plot the interactions:

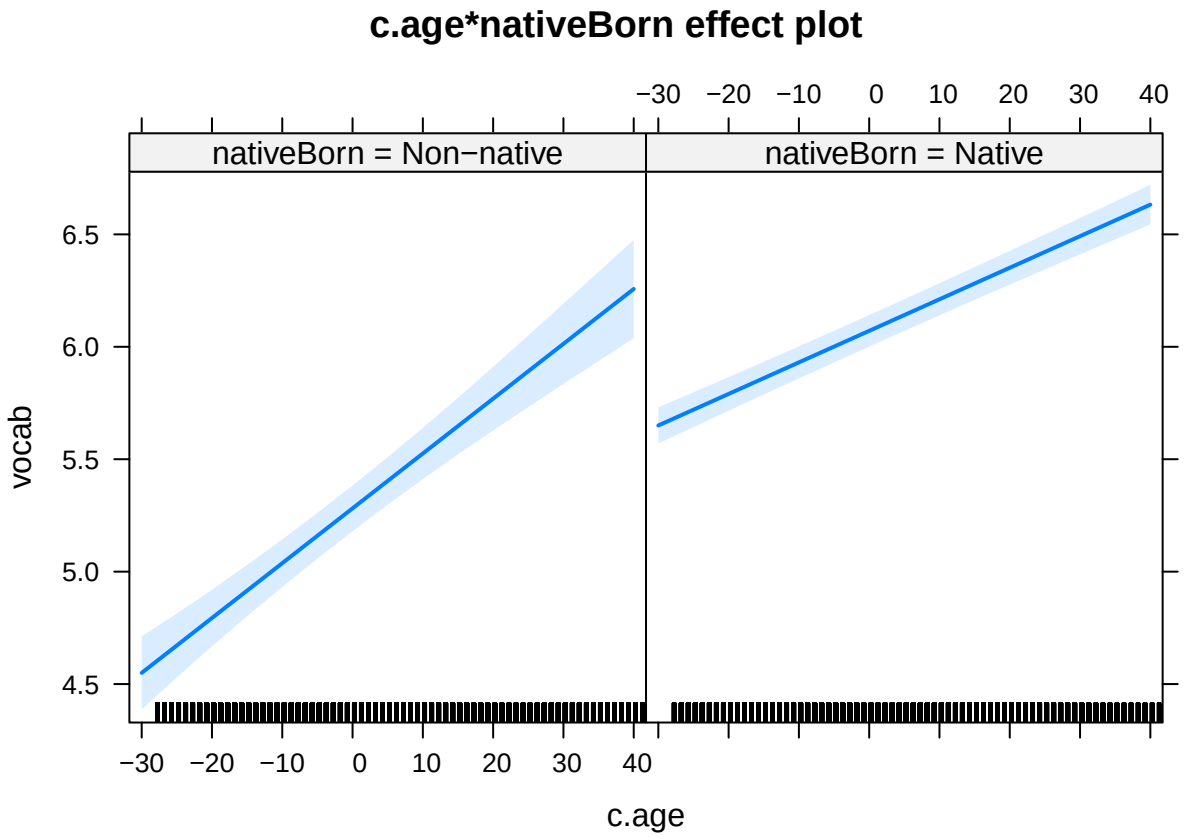
Note: When plotting effects you need to give the name of the effect in the same order as you wrote it into the model e.g. c.age BY gender. If you write it in a different order, e.g. gender BY c.age, it will not recognise this term and the plot will break

```
## NOTE: c.age:gender is not a high-order term in the model
```



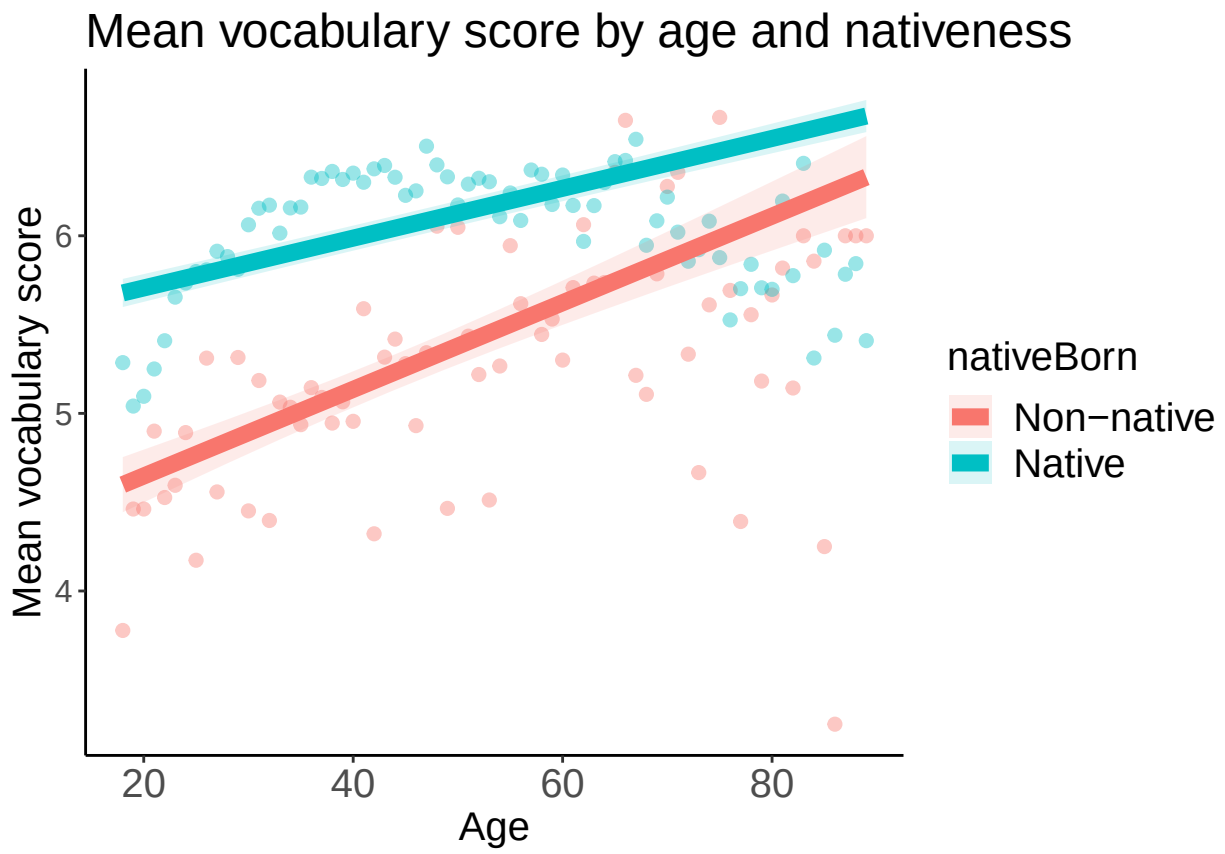
Indeed, we can see that the relationship between age and vocabulary scores is weaker (=smaller slope) for males.

```
## NOTE: c.age:nativeBorn is not a high-order term in the model
```



It might be useful to plot these slopes from the model on top on the raw data. This is very easy to do in ggplot: we can combine data from different files together.

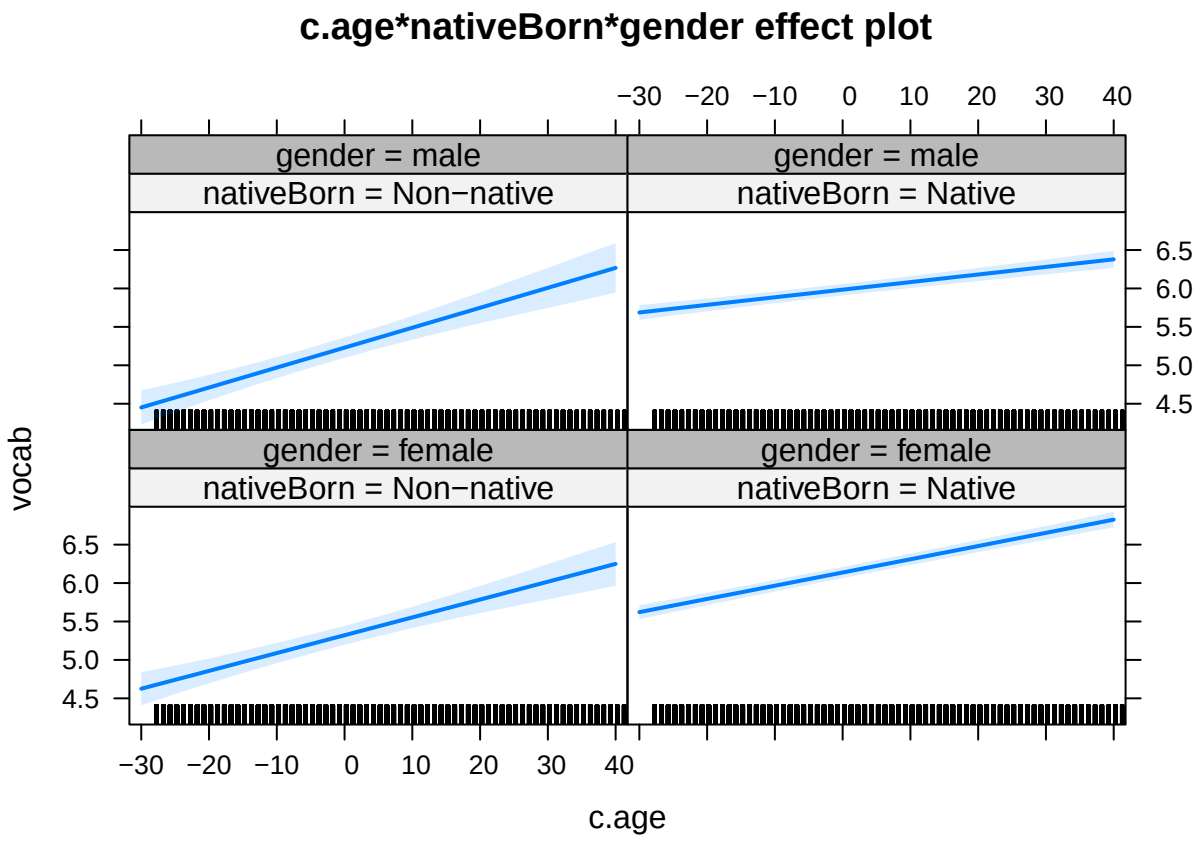
```
## NOTE: c.age:nativeBorn is not a high-order term in the model
```



Great! We see that except for the fact that natives have higher scores, nativeness is changing the relationship between age and vocabulary scores (i.e., the slope is steeper for non-natives).

Plotting (and understanding) triple interactions

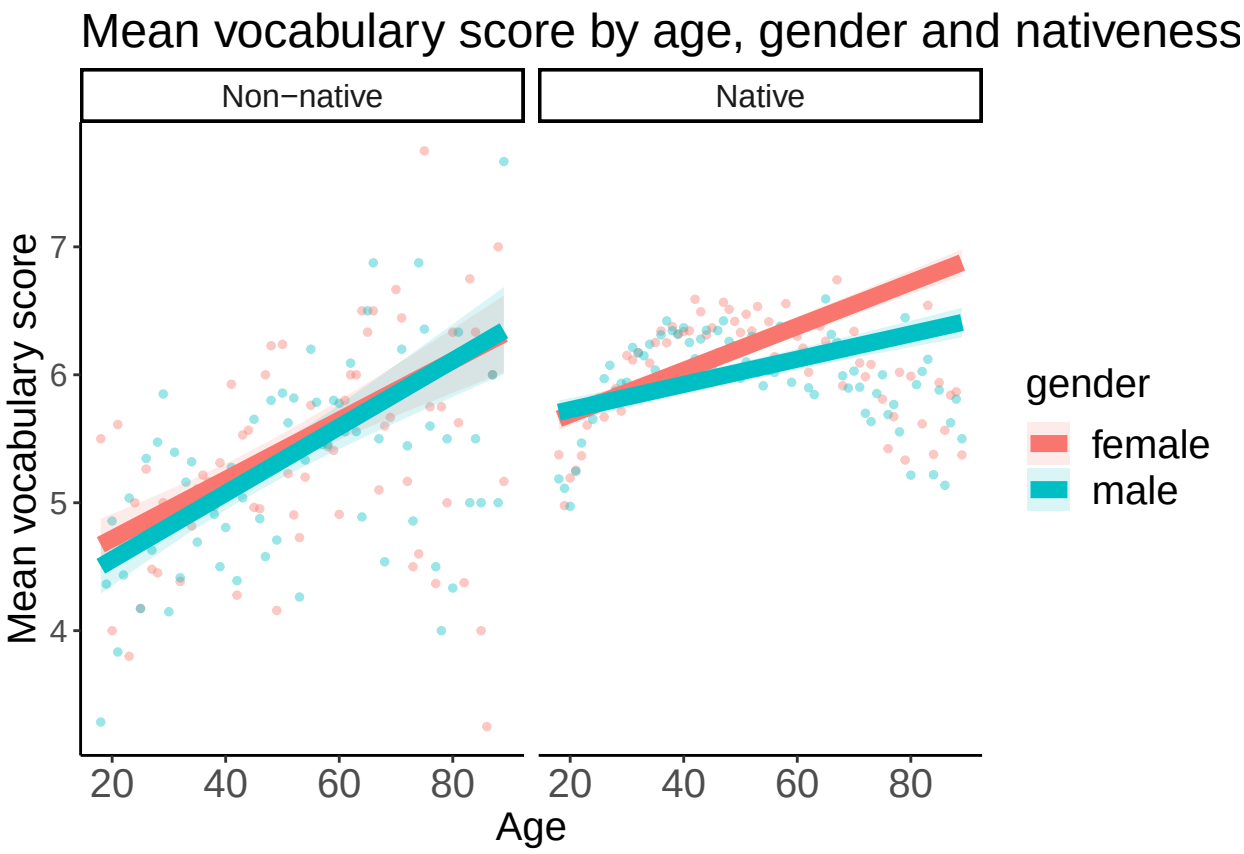
```
## NOTE: c.age:nativeBorn:gender is not a high-order term in the model
```



This plot is actually not so clear and it's hard to see the differences, but the idea is that the significant relationship between age and gender (i.e., males showing a weaker relationship) is somewhat modluated by nativeness (so it is true only for the natives).

Maybe we can try to plot the model AND the raw data together to get a better picture.

```
## NOTE: c.age:nativeBorn:gender is not a high-order term in the model
```



Time to do some model plotting yourself! :)

1. Creat a new r code chunk, and give it a useful name
2. Plot the interaction between nativeness and gender
3. Plot the interaction between nativeness and education
4. Plot the interaction between gender and education
5. Plot the triple interaction between gender, education and nativeness
6. Plot the raw data + model estimates for one of the interactions above
7. Check: do the plots fit the model's output and the significance of the interactions?