A1\_P1\_Student

Riccardo Fusaroli

01/07/2018

# Assignment 1, Part 1: Language development in Autism Spectrum Disorder (ASD) - Brushing up your code skills

In this first part of the assignment we will brush up your programming skills, and make you familiar with the data sets you will be analysing for the next parts of the assignment.

In this first part of the assignment you will: 1) Create a Github (or gitlab) account and link it to your RStudio 2) Use small nifty lines of code to transform several data sets into just one. The final data set will contain only the variables that are needed for the analysis in the next parts of the assignment 3) Become familiar with the tidyverse package (especially the sub-packages stringr and dplyr), which you will find handy for later assignments.

## 0. First an introduction on the data

# Language development in Autism Spectrum Disorder (ASD)

Background: Autism Spectrum Disorder is often related to language impairment. However, this phenomenon has not been empirically traced in detail: i) relying on actual naturalistic language production, ii) over extended periods of time. We therefore videotaped circa 30 kids with ASD and circa 30 comparison kids (matched by linguistic performance at visit 1) for ca. 30 minutes of naturalistic interactions with a parent. We repeated the data collection 6 times per kid, with 4 months between each visit. We transcribed the data and counted: i) the amount of words that each kid uses in each video. Same for the parent. ii) the amount of unique words that each kid uses in each video. Same for the parent. iii) the amount of morphemes per utterance (Mean Length of Utterance) displayed by each child in each video. Same for the parent.

## 1. Let's get started on GitHub

Follow the link to a Github tutorial: <https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN>

In the assignments you will be asked to upload your code on Github and the GitHub repositories will be part of the portfolio, therefore all students must make an account and link it to their RStudio (you'll thank us later for this!).

N.B. Create a GitHub repository for the Language Development in ASD set of assignments and link it to a project on your RStudio.

You may also use Gitlab instead of Github (Malte will explain in class)

## 2. Now let's take dirty dirty data sets and make them into a tidy one

If you're not in a project in Rstudio, make sure to set your working directory here. If you created an rstudio project, then your working directory (the directory with your data and code for these assignments) is the project directory. You can always see which working directory is used with getwd(). Note that this may be different in the console than in the Rmd cells (don't ask me why).

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.4.2

## ── Attaching packages ────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.5  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.6  
## ✔ tidyr 0.8.1 ✔ stringr 1.2.0  
## ✔ readr 1.1.1 ✔ forcats 0.2.0

## Warning: package 'tibble' was built under R version 3.4.3

## Warning: package 'tidyr' was built under R version 3.4.4

## Warning: package 'purrr' was built under R version 3.4.4

## Warning: package 'dplyr' was built under R version 3.4.4

## ── Conflicts ───────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

library(stringr)  
setwd("~/Studygroup/Assignment1&2\_LanguageASD")

Load the three data sets, after downloading them from dropbox and saving them in your working directory: \* Demographic data for the participants: <https://www.dropbox.com/s/w15pou9wstgc8fe/demo_train.csv?dl=0> \* Length of utterance data: <https://www.dropbox.com/s/usyauqm37a76of6/LU_train.csv?dl=0> \* Word data: <https://www.dropbox.com/s/8ng1civpl2aux58/token_train.csv?dl=0>

demo<-read.csv("demo\_train.csv", stringsAsFactors = FALSE)  
lu<-read.csv("LU\_train.csv", stringsAsFactors = FALSE)  
token<-read.csv("token\_train.csv", stringsAsFactors = FALSE)

Explore the 3 datasets (e.g. visualize them, summarize them, etc.). You will see that the data is messy, since the psychologists collected the demographic data, a linguist analyzed the length of utterance in May 2014 and the same linguist analyzed the words several months later. In particular: - the same variables might have different names (e.g. identifier of the child) - the same variables might report the values in different ways (e.g. visit) Welcome to real world of messy data :-)

Before being able to combine the data sets we need to make sure the relevant variables have the same names and the same kind of values.

So:

2a. Find a way to transform variable names. Tip: Look into the package dplyr (part of tidyverse) Tip: Or google "how to rename variables in R". Tip: Or look through the chapter on data transformation in R for data science (<http://r4ds.had.co.nz>).

#change colnames  
colnames(demo)[colnames(demo)=="Child.ID"] <- "SUBJ"  
colnames(demo)[colnames(demo)=="Visit"] <- "VISIT"  
  
#change to lower case in order to try to make them equal - this was not nessecary, since we used str\_extract later  
lu <- data.frame(lapply(lu, function(v) {  
 if (is.character(v)) return(tolower(v))  
 else return(v)  
}))  
  
token <- data.frame(lapply(token, function(v) {  
 if (is.character(v)) return(tolower(v))  
 else return(v)  
}))  
  
demo <- data.frame(lapply(demo, function(v) {  
 if (is.character(v)) return(tolower(v))  
 else return(v)  
}))  
  
#change visit variable  
lu$VISIT <- revalue(lu$VISIT, c("visit1." = "1", "visit2." = "2", "visit3." = "3", "visit4." = "4", "visit5." = "5", "visit6." = "6"))  
token$VISIT <- revalue(token$VISIT, c("visit1." = "1", "visit2." = "2", "visit3." = "3", "visit4." = "4", "visit5." = "5", "visit6." = "6"))

2b. Find a way to homogeneize the way "visit" is reported. If you look into the original data sets, you will see that in the LU data and the Token data, Visits are called "visit 1"" instead of just "1"" (which is the case in the demographic data set). Tip: There is a package called stringr, which will be very handy for manipulating (text) strings also in furture assignments. We will return to this package later, but for now use the str\_extract () to extract only the number from the variable Visit in each data set.

#extract numbers  
lu$VISIT <- str\_extract(lu$VISIT,"[0-9]+")  
token$VISIT <- str\_extract(token$VISIT,"[0-9]+")  
  
#alternatives: [:digit:] -> find a number/extract whatever is in []

2c. We also need to make a small adjustment to the content of the Child.ID coloumn in the demographic data. Within this column, names that are not abbreviations do not end with "." (i.e. Adam), which is the case in the other two data sets (i.e. Adam.). If The content of the two variables isn't identical the data sets will not be merged sufficiently. We wish to remove the "." at the end of names in the LU data and the tokens data.

Tip: stringr is helpful again. Look up str\_replace\_all Tip: You can either have one line of code for each child name that is to be changed (easier, more typing) or specify the pattern that you want to match (more complicated: look up "regular expressions", but less typing)

Tip: You will have to do identical work for both data sets, so to save time on the copy/paste use the cmd+f/ctrl+f function. Add the data frame name (e.g. LU\_data) in the first box, and the data frame name (e.g. Tokens\_data) you wish to change it to in the other box, and press replace. Or create a function that takes the data set and does the transformation. Then call the function on both data sets.

#removing the . at the end of the names  
  
lu$SUBJ = as.character(lu$SUBJ)  
token$SUBJ = as.character(token$SUBJ)  
  
#detect all the names with only 1 . and remove  
for (row in 1:nrow(lu)) {  
 if (str\_count(lu[row, "SUBJ"], "[.]") == 1) {  
 lu[row,"SUBJ"] <- str\_replace(lu[row, "SUBJ"],"[.]", "")  
 }  
}  
  
for (row in 1:nrow(token)) {  
 if (str\_count(token[row, "SUBJ"], "[.]") == 1) {  
 token[row,"SUBJ"] <- str\_replace(token[row, "SUBJ"],"[.]", "")  
 }  
}  
  
#alternative: ".$" -> looking for a dot in the end of the string

2d. Now that the nitty gritty details of the different data sets are fixed, we want to make a subset of each data set only containig the variables that we wish to use in the final data set. For this we use the tidyverse package dplyr, which contains the function select().

The variables we need are: Child.ID, Visit, Ethnicity, Diagnosis, Gender, Age, ADOS, MullenRaw, ExpressiveLangRaw, MOT\_MLU, MOT\_LUstd, CHI\_MLU, CHI\_LUstd, types\_MOT, types\_CHI, tokens\_MOT, tokens\_CHI.

* ADOS indicates the severity of the autistic symptoms (the higher the score, the worse the symptoms)
* MLU stands for mean length of utterance
* types stands for unique words (e.g. even if "doggie" is used 100 times it only counts for 1)
* tokens stands for overall amount of words (if "doggie" is used 100 times it counts for 100)
* MullenRaw indicates non verbal IQ
* ExpressiveLangRaw indicates verbal IQ

It would be smart to rename the last two into something you can remember (i.e. nonVerbalIQ, verbalIQ)

#renaming verbal IQ  
colnames(demo)[colnames(demo)=="MullenRaw"] <- "nonVerbalIQ"  
colnames(demo)[colnames(demo)=="ExpressiveLangRaw"] <- "VerbalIQ"  
  
#creating subsets  
selected\_demo <- select(demo, SUBJ, VISIT, Ethnicity, Diagnosis, Gender, Age, ADOS, nonVerbalIQ, VerbalIQ)  
selected\_lu <- select(lu, SUBJ, VISIT, MOT\_MLU, MOT\_LUstd, CHI\_MLU, CHI\_LUstd)  
selected\_token <- select(token, SUBJ, VISIT, types\_MOT, types\_CHI, tokens\_MOT, tokens\_CHI)

2e. Finally we are ready to merge all the data sets into just one. Tip: Look up "joins" in R for data science, or google "How to merge datasets in R" Tip: Joining / merging only works for two data frames at the time.

#merging  
merged\_data1 <- join(selected\_demo, selected\_lu, by= c("SUBJ", "VISIT"))  
fulldata <- join(merged\_data1, selected\_token, by= c("SUBJ", "VISIT"))  
  
#inner, left, right, full joins are options

Are we done? If you look at the data set now, you'll se a lot of NA's in the variables ADOS, nonVerbalIQ (MullenRaw) and verbalIQ (ExpressiveLangRaw). These measures were not taken at all visits. Additionally, we only want these measures for the first visit (Riccardo will explain why in class). So let's make sure that we select only these variables as collected during the first visit for each child and repeat these values throughout all other visits.

* A way to get around this in R, is to make a subset of the data containing only the values we are interested in, i.e. the values at visit 1. We only want to keep the relevant variables in this data set, i.e the ones which values are to be repeated. Either the the subset() function or select() and filter() can be used here. Solve this issue with both functions to familiarize yourself with these useful functions. N.B. save the subset of the dataset as a new dataset, do not overwrite the old one.
* In order to merge these new variables to the final data set, they'll need new names. E.g change the ADOS variable to ADOS1.
* Once you've changed the names, the subset can be merged to the final data set, and the score at the 1st visit in each variable will be spread to all 6.
* Lastly, there are too many unneccesary variables in the data set by now. Use the select() to choose only the variables you want in the data (e.g. remove the old ADOS, verbal and nonVerbal IQ variables, so you will not get confused later on) and define the order of the variables. Hint: You want the child identifier, followed by diagnosis, followed by demographic, cognitive and clinical features, followed by indexes of linguistic performance (utterances lenghts, types and tokens of words).

#using subset()  
subset1 <- subset(fulldata, VISIT == "1", select = c("SUBJ", "VISIT", "ADOS", "nonVerbalIQ", "VerbalIQ"))  
  
#using filter()  
subset2 <- filter(fulldata, VISIT == "1")

## Warning: package 'bindrcpp' was built under R version 3.4.4

subset2 <- select(subset2, SUBJ, VISIT, ADOS, nonVerbalIQ, VerbalIQ)  
  
#renaming ADOS  
colnames(subset1)[colnames(subset1)=="ADOS"] <- "ADOS1"  
colnames(subset1)[colnames(subset1)=="nonVerbalIQ"] <- "nonVerbalIQ1"  
colnames(subset1)[colnames(subset1)=="VerbalIQ"] <- "VerbalIQ1"  
  
#merge  
fulldata <- join(fulldata, subset1, by= "SUBJ")  
  
#selecting relevant data  
relevant\_data <- subset(fulldata, select = -c(ADOS, nonVerbalIQ, VerbalIQ))  
  
#reordering  
ordered\_data <- relevant\_data[c("SUBJ", "VISIT", "Diagnosis", "Ethnicity", "Gender", "Age", "ADOS1", "nonVerbalIQ1", "VerbalIQ1", "MOT\_MLU", "MOT\_LUstd", "CHI\_MLU", "CHI\_LUstd", "types\_MOT", "types\_CHI", "tokens\_MOT", "tokens\_CHI")]

Now, we are almost ready to start analysing the data. However, here are some additional finishing touches:

* in some experiments your participants must be anonymous. Therefore we wish to turn the CHILD.ID into numbers.

Tip: You will probably need to turn it into a factor first, then a number Tip: google "R how to convert character to integer" or look up the as.??? functions

* In order to make it easier to work with this nice, clean dataset in the future, it is practical to make sure the variables have sensible values. E.g. right now gender is marked 1 and 2, but in two weeks you will not be able to remember, which gender were connected to which number, so change the values from 1 and 2 to F and M in the gender variable. For the same reason, you should also change the values of Diagnosis from A and B to ASD (autism spectrum disorder) and TD (typically developing). Tip: Try taking a look at ifelse(), or google "how to rename levels in R".

#anonymous data  
ordered\_data$SUBJ = as.factor(ordered\_data$SUBJ)  
ordered\_data$SUBJ <- as.numeric(ordered\_data$SUBJ)  
  
#change gender  
ordered\_data <- mutate(ordered\_data, Gender = ifelse (Gender %in% c("1"), "M", "F"))  
#  
ordered\_data <- mutate(ordered\_data, Diagnosis = ifelse (Diagnosis %in% c("a"), "ASD", "TD"))

Write the data set into a csv file.

write.csv(ordered\_data, file = "Portfolio1THEA.csv")

1. Now that we have a nice clean data set to use for the analysis next week, we shall play a bit around with it. The following exercises are not relevant for the analysis, but are here so you can get familiar with the functions within the tidyverse package.

Here's the link to a very helpful book, which explains each function: <http://r4ds.had.co.nz/index.html>

USING FILTER List all kids who: 1. have a mean length of utterance (across all visits) of more than 2.7 morphemes. 2. have a mean length of utterance of less than 1.5 morphemes at the first visit 3. have not completed all trials. Tip: Use pipes %>% to solve this

detach(package:plyr)  
  
#1.  
kids1 <- ordered\_data %>%  
 filter(CHI\_MLU > 2.7)  
unique(kids1)

## SUBJ VISIT Diagnosis Ethnicity Gender Age ADOS1 nonVerbalIQ1  
## 1 2 5 TD white M 35.90 0 28  
## 2 2 6 TD white M 40.13 0 28  
## 3 3 4 TD white F 35.53 1 29  
## 4 3 5 TD white F 39.47 1 29  
## 5 3 6 TD white F 45.07 1 29  
## 6 5 4 TD white F 31.07 0 24  
## 7 5 5 TD white F 35.00 0 24  
## 8 6 6 TD white M 40.27 0 21  
## 9 7 2 ASD white M 33.17 13 34  
## 10 7 3 ASD white M 37.07 13 34  
## 11 7 4 ASD white M 41.07 13 34  
## 12 7 6 ASD white M 49.70 13 34  
## 13 8 3 ASD white/latino M 38.90 8 31  
## 14 8 4 ASD white/latino M 43.13 8 31  
## 15 8 5 ASD white/latino M 47.40 8 31  
## 16 8 6 ASD white/latino M 51.37 8 31  
## 17 9 4 ASD white M 45.53 9 34  
## 18 9 6 ASD white M 54.13 9 34  
## 19 4 4 TD white M 32.07 5 32  
## 20 12 6 TD white M 40.43 3 27  
## 21 14 3 TD white M 28.27 0 30  
## 22 14 4 TD white M 32.07 0 30  
## 23 14 5 TD white M 35.87 0 30  
## 24 14 6 TD white M 41.50 0 30  
## 25 15 4 TD white M 31.03 0 25  
## 26 15 5 TD white M 35.37 0 25  
## 27 15 6 TD white M 39.40 0 25  
## 28 16 5 TD white M 35.10 0 23  
## 29 16 6 TD white M 39.43 0 23  
## 30 17 6 TD white M 40.30 0 24  
## 31 18 2 TD white M 26.27 0 29  
## 32 18 3 TD white M 30.63 0 29  
## 33 18 5 TD white M 38.17 0 29  
## 34 18 6 TD white M 42.93 0 29  
## 35 20 2 TD white M 26.13 0 29  
## 36 20 3 TD white M 30.03 0 29  
## 37 20 4 TD white M 34.43 0 29  
## 38 20 5 TD white M 38.70 0 29  
## 39 20 6 TD white M 44.07 0 29  
## 40 22 6 ASD white M 37.30 9 26  
## 41 25 5 TD white F 35.13 0 21  
## 42 25 6 TD white F 39.23 0 21  
## 43 26 3 TD white M 30.13 0 29  
## 44 26 4 TD white M 34.00 0 29  
## 45 26 5 TD white M 37.93 0 29  
## 46 26 6 TD white M 42.47 0 29  
## 47 29 3 TD white M 32.07 0 27  
## 48 29 4 TD white M 35.03 0 27  
## 49 32 1 ASD white M 30.40 11 32  
## 50 32 3 ASD white M 38.60 11 32  
## 51 32 4 ASD white M 42.63 11 32  
## 52 32 5 ASD white M 46.93 11 32  
## 53 32 6 ASD white M 51.00 11 32  
## 54 41 5 ASD white M 55.17 7 33  
## 55 41 6 ASD white M 58.77 7 33  
## 56 40 3 TD white F 28.07 1 29  
## 57 40 5 TD white F 36.13 1 29  
## 58 43 6 TD white M 39.93 0 25  
## 59 44 4 TD white M 32.07 0 24  
## 60 44 5 TD white M 36.40 0 24  
## 61 44 6 TD white M 40.13 0 24  
## 62 47 4 ASD lebanese M NA 13 27  
## 63 48 5 ASD white M 53.40 14 42  
## 64 45 5 TD white M 35.83 3 30  
## 65 53 4 TD white F 32.13 0 27  
## 66 53 5 TD white F 39.10 0 27  
## 67 53 6 TD white F 40.37 0 27  
## 68 54 3 TD white M 31.63 0 27  
## 69 54 4 TD white M 35.63 0 27  
## 70 54 5 TD white M 39.47 0 27  
## 71 54 6 TD white M 43.40 0 27  
## 72 57 4 TD asian F 34.43 1 22  
## 73 57 5 TD asian F 37.67 1 22  
## 74 57 6 TD asian F 42.10 1 22  
## 75 58 3 TD white M 30.77 0 29  
## 76 58 4 TD white M 35.03 0 29  
## 77 58 5 TD white M 38.60 0 29  
## 78 59 2 TD white M 28.60 0 30  
## 79 59 3 TD white M 32.50 0 30  
## 80 59 4 TD white M 36.40 0 30  
## 81 59 5 TD white M 40.07 0 30  
## 82 59 6 TD white M 44.43 0 30  
## 83 60 4 TD white M 30.83 1 24  
## 84 60 5 TD white M 35.17 1 24  
## 85 60 6 TD white M 39.30 1 24  
## 86 61 4 TD white M 32.03 0 26  
## 87 61 6 TD white M 41.93 0 26  
## 88 65 4 TD white M 33.60 4 29  
## 89 65 6 TD white M 41.00 4 29  
## 90 66 1 ASD white M 34.00 13 30  
## 91 66 2 ASD white M 38.63 13 30  
## 92 66 3 ASD white M 42.47 13 30  
## 93 66 4 ASD white M 47.00 13 30  
## 94 66 5 ASD white M 51.13 13 30  
## 95 66 6 ASD white M 54.73 13 30  
## VerbalIQ1 MOT\_MLU MOT\_LUstd CHI\_MLU CHI\_LUstd types\_MOT types\_CHI  
## 1 14 5.209615 2.814165 3.238095 2.355940 601 182  
## 2 14 4.664013 2.765261 2.865169 2.247884 595 210  
## 3 18 5.301053 2.912026 3.929204 2.673094 449 206  
## 4 18 4.566038 2.792687 3.298578 2.122091 534 207  
## 5 18 5.229885 3.014147 3.710345 2.162110 486 173  
## 6 18 4.262195 2.494374 2.775641 2.030453 400 121  
## 7 18 4.384946 2.698504 2.835821 1.913011 428 121  
## 8 15 4.287582 2.747960 2.757143 2.195357 260 168  
## 9 27 4.964664 2.499220 3.453039 2.267779 307 171  
## 10 27 4.147059 2.803222 3.119318 2.206002 351 262  
## 11 27 5.309804 2.842621 4.302326 2.531521 335 200  
## 12 27 4.588477 2.783585 3.413502 2.323497 304 245  
## 13 27 3.818681 2.420551 3.518072 2.674563 388 165  
## 14 27 4.301624 2.367015 3.257143 2.208333 356 163  
## 15 27 4.602851 2.611098 4.043478 2.367869 397 146  
## 16 27 3.532374 2.574647 3.278195 2.537992 410 166  
## 17 27 4.744000 2.538989 2.907258 2.374993 384 187  
## 18 27 4.587179 2.685925 2.766520 2.358437 462 179  
## 19 31 4.658333 2.519080 3.026596 2.045840 375 134  
## 20 18 4.235585 2.511690 2.705128 1.868147 400 73  
## 21 16 4.974684 2.728076 3.185771 2.163540 318 169  
## 22 16 3.988304 2.626870 3.000000 2.175935 197 122  
## 23 16 4.910494 2.669214 4.364754 2.577119 290 222  
## 24 16 4.468493 2.574313 3.504950 2.503952 339 201  
## 25 17 4.083700 2.429695 2.798283 2.025078 463 203  
## 26 17 4.487842 2.740111 3.230114 2.483102 511 247  
## 27 17 4.847418 2.694579 3.701195 2.707956 388 210  
## 28 17 4.446281 2.396452 3.375000 2.348335 390 238  
## 29 17 4.664286 2.732374 3.811404 2.794803 359 178  
## 30 15 4.347709 2.570906 2.869048 2.086121 327 156  
## 31 26 4.750455 2.491813 2.744108 1.909177 391 196  
## 32 26 4.164789 2.751112 2.807692 2.169244 408 200  
## 33 26 5.433579 2.515307 3.109524 2.134168 517 219  
## 34 26 4.445872 2.582458 2.948207 2.324584 491 250  
## 35 33 4.357911 2.588408 2.722034 2.019571 461 160  
## 36 33 4.116057 2.456142 3.340000 2.284236 487 201  
## 37 33 4.131579 2.684469 3.212821 2.422827 565 207  
## 38 33 3.877102 2.561362 3.090278 2.369989 575 219  
## 39 33 4.013353 2.749934 2.909535 2.244065 516 235  
## 40 14 5.379798 2.712708 2.902778 2.499035 433 155  
## 41 19 4.746606 2.668579 3.700000 2.282542 436 178  
## 42 19 4.211321 2.707896 3.091195 2.280872 478 221  
## 43 22 4.127941 2.433186 2.804217 2.144111 455 217  
## 44 22 5.362500 2.914683 3.731092 2.895077 553 291  
## 45 22 4.267409 2.686150 2.741379 2.069361 578 298  
## 46 22 4.472993 2.718080 3.061453 2.368702 555 237  
## 47 22 3.552459 2.378463 2.987526 2.167577 396 233  
## 48 22 3.667638 2.511264 2.727273 2.173550 480 186  
## 49 33 4.690751 2.545488 3.400000 1.722577 278 119  
## 50 33 4.316279 2.713869 3.919689 2.370211 333 307  
## 51 33 4.857143 2.675294 3.523810 2.537698 398 188  
## 52 33 4.345515 2.772711 3.291990 2.229706 437 261  
## 53 33 4.111413 2.889464 3.364341 2.596772 452 273  
## 54 26 4.658802 2.756894 2.746875 2.228015 407 228  
## 55 26 4.240798 2.555277 3.077419 2.352858 429 217  
## 56 28 4.577491 2.647231 2.869048 2.421529 420 175  
## 57 28 4.917927 2.721358 3.518519 2.542533 447 210  
## 58 17 4.061475 2.517218 2.852632 1.957272 397 213  
## 59 19 4.611247 2.590743 3.830040 2.388784 339 193  
## 60 19 3.921444 2.453860 3.774908 2.596669 358 213  
## 61 19 3.391525 2.517921 3.072797 2.518603 357 219  
## 62 13 3.812930 2.397431 2.990000 1.989950 430 180  
## 63 27 4.232258 2.305867 2.866505 2.185376 396 262  
## 64 20 4.113846 2.521775 3.200000 2.780354 307 131  
## 65 20 4.190698 2.622072 3.162242 2.282469 343 211  
## 66 20 3.673418 2.398771 2.738806 2.181774 315 131  
## 67 20 4.676101 2.442830 2.760000 2.025438 322 34  
## 68 27 4.737127 2.743557 3.112450 1.990793 323 151  
## 69 27 4.880240 2.722280 3.480952 2.573143 340 229  
## 70 27 5.743772 2.723008 3.537143 2.669948 425 209  
## 71 27 5.247093 2.733299 3.595000 2.406444 383 217  
## 72 14 4.137014 2.600857 3.028986 2.349659 363 112  
## 73 14 5.185941 2.686471 2.921569 2.131401 388 108  
## 74 14 5.153639 2.756146 2.761290 2.146517 383 140  
## 75 22 5.288991 2.867721 3.303754 2.271628 474 200  
## 76 22 5.338462 2.885118 3.077551 2.265560 554 177  
## 77 22 4.983389 2.948008 2.832168 2.293152 563 219  
## 78 30 4.557471 2.480875 3.217949 2.000945 197 126  
## 79 30 4.078292 2.509964 3.131356 2.255810 219 152  
## 80 30 4.458937 2.765733 3.634069 2.377306 232 217  
## 81 30 4.857143 2.619025 3.822581 2.490460 278 217  
## 82 30 3.706790 2.400582 3.243902 2.224554 249 102  
## 83 22 3.819961 2.452458 3.406504 1.899240 332 153  
## 84 22 3.750000 2.519183 3.607287 2.343832 394 244  
## 85 22 4.186161 2.590683 2.892157 2.215838 437 183  
## 86 17 4.177340 2.596526 2.862069 1.964134 318 205  
## 87 17 3.957230 2.537740 2.909274 2.178949 367 260  
## 88 22 4.697624 2.404849 2.714829 2.119628 330 157  
## 89 22 4.113158 2.372060 2.848000 2.169077 303 158  
## 90 30 3.604140 2.550110 2.876344 1.917878 400 149  
## 91 30 4.604341 2.561833 2.784000 2.214801 413 149  
## 92 30 4.907591 2.667128 4.131868 2.436936 459 196  
## 93 30 4.085409 2.536530 3.359833 2.479248 539 214  
## 94 30 4.223572 2.645720 2.965517 2.166879 521 145  
## 95 30 4.080446 2.591821 3.441558 2.434387 505 226  
## tokens\_MOT tokens\_CHI  
## 1 2553 472  
## 2 2586 686  
## 3 2397 754  
## 4 2672 588  
## 5 2564 460  
## 6 1934 390  
## 7 1879 346  
## 8 1138 666  
## 9 1270 562  
## 10 1445 983  
## 11 1286 674  
## 12 999 698  
## 13 1788 490  
## 14 1711 479  
## 15 2082 539  
## 16 2171 738  
## 17 2685 604  
## 18 3182 538  
## 19 2069 493  
## 20 2271 189  
## 21 1463 733  
## 22 632 243  
## 23 1467 916  
## 24 1498 640  
## 25 2361 538  
## 26 2668 932  
## 27 1959 864  
## 28 1864 755  
## 29 1802 793  
## 30 1395 410  
## 31 2303 733  
## 32 2675 825  
## 33 2762 583  
## 34 2264 710  
## 35 2687 738  
## 36 2479 940  
## 37 2965 1092  
## 38 2881 769  
## 39 2576 1079  
## 40 2389 590  
## 41 1965 686  
## 42 2044 847  
## 43 2589 826  
## 44 2978 1225  
## 45 2940 1145  
## 46 2895 1010  
## 47 1872 1246  
## 48 2233 750  
## 49 1450 483  
## 50 1668 1293  
## 51 2518 714  
## 52 2410 1154  
## 53 3076 1249  
## 54 2314 815  
## 55 2510 897  
## 56 2146 825  
## 57 1999 800  
## 58 1741 702  
## 59 1726 820  
## 60 1600 875  
## 61 1764 719  
## 62 2377 493  
## 63 2326 1054  
## 64 1207 433  
## 65 1559 973  
## 66 1224 304  
## 67 1371 61  
## 68 1609 671  
## 69 1452 684  
## 70 1525 586  
## 71 1701 646  
## 72 1829 348  
## 73 1986 238  
## 74 1866 395  
## 75 2142 897  
## 76 2585 642  
## 77 2772 723  
## 78 686 449  
## 79 1020 660  
## 80 798 978  
## 81 1067 814  
## 82 1024 358  
## 83 1699 694  
## 84 1977 725  
## 85 2347 517  
## 86 1524 1051  
## 87 1731 1294  
## 88 1974 637  
## 89 1460 659  
## 90 2587 469  
## 91 2534 670  
## 92 2841 698  
## 93 3163 693  
## 94 3090 357  
## 95 3072 713

#2.   
kids2 <- ordered\_data %>%  
 filter(CHI\_MLU < 1.5, VISIT == 1)  
unique(kids2)

## SUBJ VISIT Diagnosis Ethnicity Gender Age ADOS1 nonVerbalIQ1  
## 1 2 1 TD white M 19.80 0 28  
## 2 5 1 TD white F 18.30 0 24  
## 3 6 1 TD white M 19.23 0 21  
## 4 9 1 ASD white M 34.03 9 34  
## 5 10 1 ASD bangladeshi F 26.17 17 20  
## 6 11 1 ASD white F 41.00 18 24  
## 7 12 1 TD white M 19.27 3 27  
## 8 14 1 TD white M 20.07 0 30  
## 9 16 1 TD white M 18.97 0 23  
## 10 17 1 TD white M 19.27 0 24  
## 11 19 1 ASD white M 34.80 14 25  
## 12 21 1 ASD white M 35.80 11 28  
## 13 22 1 ASD white M 18.77 9 26  
## 14 23 1 ASD african american M 27.53 21 22  
## 15 25 1 TD white F 18.93 0 21  
## 16 27 1 TD white M 21.03 0 26  
## 17 30 1 ASD white/latino M 27.37 14 25  
## 18 31 1 ASD white M 37.47 20 13  
## 19 33 1 ASD white M 34.87 17 26  
## 20 34 1 ASD white M 36.53 12 31  
## 21 36 1 TD white M 19.30 1 23  
## 22 37 1 ASD african american F 25.33 14 25  
## 23 38 1 ASD white M 33.77 10 27  
## 24 39 1 TD white M 19.20 3 19  
## 25 40 1 TD white F 19.87 1 29  
## 26 41 1 ASD white M 39.50 7 33  
## 27 42 1 ASD white/asian M 33.20 11 26  
## 28 43 1 TD white M 19.23 0 25  
## 29 44 1 TD white M 19.37 0 24  
## 30 47 1 ASD lebanese M 24.90 13 27  
## 31 45 1 TD white M 19.77 3 30  
## 32 46 1 TD white M 20.03 5 24  
## 33 51 1 ASD white M 36.73 20 21  
## 34 53 1 TD white F 20.03 0 27  
## 35 54 1 TD white M 23.07 0 27  
## 36 55 1 ASD white M 31.63 17 28  
## 37 56 1 ASD white M 37.47 19 17  
## 38 57 1 TD asian F 20.87 1 22  
## 39 58 1 TD white M 22.57 0 29  
## 40 60 1 TD white M 19.10 1 24  
## 41 61 1 TD white M 19.97 0 26  
## 42 62 1 ASD white M 35.50 14 27  
## 43 63 1 ASD white F 41.07 15 28  
## 44 64 1 ASD white M 26.00 15 30  
## 45 65 1 TD white M 20.80 4 29  
## 46 67 1 ASD white M 42.00 15 27  
## VerbalIQ1 MOT\_MLU MOT\_LUstd CHI\_MLU CHI\_LUstd types\_MOT types\_CHI  
## 1 14 3.621993 2.164553 1.2522523 0.4739801 378 14  
## 2 18 3.544419 2.272387 1.0378788 0.1909031 363 36  
## 3 15 3.380463 2.214518 1.2168675 0.4121116 215 24  
## 4 27 3.986357 2.500713 1.3947368 0.6897549 324 57  
## 5 17 2.618729 1.935874 1.0000000 0.0000000 212 4  
## 6 14 2.244755 1.510878 1.2641509 0.7177255 152 29  
## 7 18 4.204846 2.384767 1.0375000 0.1899836 289 15  
## 8 16 4.195335 2.280551 1.0877193 0.2828862 235 17  
## 9 17 3.420315 2.273399 1.0396040 0.1950269 287 7  
## 10 15 3.967078 2.302921 1.1647059 0.4813476 277 27  
## 11 11 3.182390 2.269630 1.0277778 0.1643355 281 9  
## 12 20 2.539823 1.994618 1.3595506 0.7606501 283 89  
## 13 14 2.524740 2.141337 0.1857143 0.4564727 321 16  
## 14 8 4.390879 3.068753 1.0000000 0.0000000 485 8  
## 15 19 3.630476 2.452720 1.2371134 0.4926929 343 36  
## 16 18 3.616867 2.261294 1.3661972 0.6763497 317 37  
## 17 19 3.024548 2.278008 1.4324324 1.1159904 328 41  
## 18 11 2.917355 2.073392 1.0833333 0.3996526 193 6  
## 19 14 3.304189 2.372505 1.0086207 0.0924466 295 6  
## 20 13 3.607088 2.340376 0.9000000 0.4358899 366 11  
## 21 21 3.561364 2.229117 1.2641509 0.5546015 291 24  
## 22 11 2.287293 1.928359 1.2500000 0.5747670 206 13  
## 23 22 2.743455 1.892987 1.3766234 0.6353026 214 67  
## 24 13 3.921109 2.376179 1.2307692 0.5756396 281 8  
## 25 28 3.420975 2.244646 1.3322785 0.7633014 342 96  
## 26 26 4.135036 2.563076 0.4805825 0.7221272 381 39  
## 27 19 3.298748 1.995123 1.2043011 0.5784507 274 36  
## 28 17 3.093146 2.362590 1.0262009 0.2791346 333 32  
## 29 19 4.033333 2.460373 1.3034483 0.7818999 373 47  
## 30 13 2.997050 2.202839 1.0175439 0.1312862 252 9  
## 31 20 3.088757 2.098655 1.2592593 0.4382281 275 9  
## 32 20 2.776181 1.904606 0.5584416 0.7811664 258 20  
## 33 9 4.883966 2.773678 1.1666667 0.5000000 387 10  
## 34 20 3.943005 2.384506 1.0761421 0.3010798 260 13  
## 35 27 4.030075 2.695787 1.4258373 1.0238787 255 73  
## 36 10 3.765528 2.329794 1.2500000 0.6123724 303 17  
## 37 10 3.704110 2.276770 1.1000000 0.4898979 265 8  
## 38 14 3.435743 2.257715 1.1818182 0.4575657 331 7  
## 39 22 5.344227 2.917012 1.4086957 0.8433438 441 92  
## 40 22 3.487871 2.432205 1.3139535 0.7664723 214 32  
## 41 17 3.509138 2.189292 1.1846154 0.3879852 178 11  
## 42 11 2.548969 1.706620 1.0444444 0.2060804 195 9  
## 43 10 3.833770 2.417727 0.0000000 0.0000000 386 0  
## 44 24 2.747100 1.853343 1.1809045 0.4454541 338 98  
## 45 22 3.432387 2.124274 1.1830065 1.0317693 345 62  
## 46 16 3.030405 2.107521 1.0375000 0.1899836 303 15  
## tokens\_MOT tokens\_CHI  
## 1 1835 139  
## 2 1408 137  
## 3 1136 101  
## 4 2859 197  
## 5 761 29  
## 6 578 130  
## 7 1808 83  
## 8 1262 62  
## 9 1625 105  
## 10 1643 99  
## 11 1418 37  
## 12 1019 227  
## 13 1787 214  
## 14 2826 122  
## 15 1698 118  
## 16 1361 95  
## 17 2138 103  
## 18 654 26  
## 19 1643 117  
## 20 2054 21  
## 21 1344 260  
## 22 788 35  
## 23 893 180  
## 24 1631 16  
## 25 2035 398  
## 26 1988 337  
## 27 1537 109  
## 28 1547 235  
## 29 2334 176  
## 30 1827 58  
## 31 1417 68  
## 32 1188 91  
## 33 2144 63  
## 34 1347 212  
## 35 1452 286  
## 36 2147 40  
## 37 1215 43  
## 38 1503 38  
## 39 2267 319  
## 40 1118 109  
## 41 1144 154  
## 42 955 47  
## 43 2613 0  
## 44 2084 233  
## 45 1805 244  
## 46 1579 166

#3.  
kids3 <- ordered\_data %>%  
 group\_by(SUBJ) %>%  
 filter(is.na(CHI\_MLU)) %>%  
 summarise(n())  
unique(kids3)

## # A tibble: 17 x 2  
## SUBJ `n()`  
## <dbl> <int>  
## 1 1 1  
## 2 4 1  
## 3 5 1  
## 4 7 1  
## 5 11 1  
## 6 13 2  
## 7 19 1  
## 8 24 3  
## 9 28 6  
## 10 32 1  
## 11 40 1  
## 12 45 1  
## 13 47 1  
## 14 49 1  
## 15 50 1  
## 16 52 2  
## 17 61 1

#does not run

USING ARRANGE

1. Sort kids to find the kid who produced the most words on the 6th visit
2. Sort kids to find the kid who produced the least amount of words on the 1st visit.

#1. showing the ranking  
kids4 <- ordered\_data %>%  
 filter(VISIT == 6) %>%  
 arrange(desc(tokens\_CHI))  
unique(kids4)

## SUBJ VISIT Diagnosis Ethnicity Gender Age ADOS1 nonVerbalIQ1  
## 1 61 6 TD white M 41.93 0 26  
## 2 32 6 ASD white M 51.00 11 32  
## 3 20 6 TD white M 44.07 0 29  
## 4 26 6 TD white M 42.47 0 29  
## 5 48 6 ASD white M 57.37 14 42  
## 6 41 6 ASD white M 58.77 7 33  
## 7 15 6 TD white M 39.40 0 25  
## 8 25 6 TD white F 39.23 0 21  
## 9 16 6 TD white M 39.43 0 23  
## 10 8 6 ASD white/latino M 51.37 8 31  
## 11 44 6 TD white M 40.13 0 24  
## 12 66 6 ASD white M 54.73 13 30  
## 13 18 6 TD white M 42.93 0 29  
## 14 43 6 TD white M 39.93 0 25  
## 15 7 6 ASD white M 49.70 13 34  
## 16 2 6 TD white M 40.13 0 28  
## 17 6 6 TD white M 40.27 0 21  
## 18 65 6 TD white M 41.00 4 29  
## 19 54 6 TD white M 43.40 0 27  
## 20 14 6 TD white M 41.50 0 30  
## 21 64 6 ASD white M 46.07 15 30  
## 22 36 6 TD white M 39.07 1 23  
## 23 38 6 ASD white M 53.77 10 27  
## 24 22 6 ASD white M 37.30 9 26  
## 25 47 6 ASD lebanese M 46.40 13 27  
## 26 39 6 TD white M 38.53 3 19  
## 27 9 6 ASD white M 54.13 9 34  
## 28 27 6 TD white M 41.17 0 26  
## 29 60 6 TD white M 39.30 1 24  
## 30 3 6 TD white F 45.07 1 29  
## 31 58 6 TD white M 43.03 0 29  
## 32 29 6 TD white M 43.80 0 27  
## 33 17 6 TD white M 40.30 0 24  
## 34 57 6 TD asian F 42.10 1 22  
## 35 59 6 TD white M 44.43 0 30  
## 36 55 6 ASD white M NA 17 28  
## 37 10 6 ASD bangladeshi F 46.53 17 20  
## 38 21 6 ASD white M 56.73 11 28  
## 39 30 6 ASD white/latino M 47.50 14 25  
## 40 67 6 ASD white M 62.33 15 27  
## 41 33 6 ASD white M 55.17 17 26  
## 42 4 6 TD white M 40.17 5 32  
## 43 34 6 ASD white M 56.43 12 31  
## 44 12 6 TD white M 40.43 3 27  
## 45 31 6 ASD white M 62.40 20 13  
## 46 51 6 ASD white M 57.43 20 21  
## 47 42 6 ASD white/asian M 53.63 11 26  
## 48 35 6 ASD white M 54.63 21 21  
## 49 37 6 ASD african american F 46.17 14 25  
## 50 63 6 ASD white F 61.70 15 28  
## 51 53 6 TD white F 40.37 0 27  
## 52 19 6 ASD white M 54.43 14 25  
## 53 62 6 ASD white M NA 14 27  
## 54 56 6 ASD white M 62.40 19 17  
## 55 23 6 ASD african american M 48.97 21 22  
## 56 5 6 TD white F NA 0 24  
## 57 11 6 ASD white F 60.33 18 24  
## 58 28 6 TD white M 39.43 0 20  
## 59 40 6 TD white F 40.23 1 29  
## 60 45 6 TD white M 39.93 3 30  
## VerbalIQ1 MOT\_MLU MOT\_LUstd CHI\_MLU CHI\_LUstd types\_MOT types\_CHI  
## 1 17 3.957230 2.537740 2.9092742 2.1789486 367 260  
## 2 33 4.111413 2.889464 3.3643411 2.5967720 452 273  
## 3 33 4.013353 2.749934 2.9095355 2.2440646 516 235  
## 4 22 4.472993 2.718080 3.0614525 2.3687019 555 237  
## 5 27 4.250853 2.596967 2.6794872 1.8979989 374 219  
## 6 26 4.240798 2.555277 3.0774194 2.3528582 429 217  
## 7 17 4.847418 2.694579 3.7011952 2.7079561 388 210  
## 8 19 4.211321 2.707896 3.0911950 2.2808716 478 221  
## 9 17 4.664286 2.732374 3.8114035 2.7948028 359 178  
## 10 27 3.532374 2.574647 3.2781955 2.5379922 410 166  
## 11 19 3.391525 2.517921 3.0727969 2.5186027 357 219  
## 12 30 4.080446 2.591821 3.4415584 2.4343866 505 226  
## 13 26 4.445872 2.582458 2.9482072 2.3245843 491 250  
## 14 17 4.061475 2.517218 2.8526316 1.9572721 397 213  
## 15 27 4.588477 2.783585 3.4135021 2.3234966 304 245  
## 16 14 4.664013 2.765261 2.8651685 2.2478838 595 210  
## 17 15 4.287582 2.747960 2.7571429 2.1953569 260 168  
## 18 22 4.113158 2.372060 2.8480000 2.1690772 303 158  
## 19 27 5.247093 2.733299 3.5950000 2.4064445 383 217  
## 20 16 4.468493 2.574313 3.5049505 2.5039524 339 201  
## 21 24 3.320000 1.990377 2.1553398 1.5439030 335 197  
## 22 21 4.239198 2.596866 2.3815789 2.1396452 411 156  
## 23 22 3.370937 2.314655 2.2745098 1.5852065 342 157  
## 24 14 5.379798 2.712708 2.9027778 2.4990353 433 155  
## 25 13 3.926928 2.532082 2.1586207 1.8389497 417 170  
## 26 13 4.388235 2.571003 2.5614754 1.9013346 324 165  
## 27 27 4.587179 2.685925 2.7665198 2.3584374 462 179  
## 28 18 4.254505 2.576855 2.4800000 1.8810872 385 154  
## 29 22 4.186161 2.590683 2.8921569 2.2158381 437 183  
## 30 18 5.229885 3.014147 3.7103448 2.1621102 486 173  
## 31 22 5.587332 2.899559 2.4292929 1.9981179 548 163  
## 32 22 3.603473 2.723471 2.0717489 1.7854067 475 185  
## 33 15 4.347709 2.570906 2.8690476 2.0861206 327 156  
## 34 14 5.153639 2.756146 2.7612903 2.1465167 383 140  
## 35 30 3.706790 2.400582 3.2439024 2.2245543 249 102  
## 36 10 3.370093 2.089905 1.4734513 0.8681705 319 98  
## 37 17 2.483146 2.109968 1.4967320 1.7863834 158 66  
## 38 20 3.156695 2.184106 2.1450382 1.4148727 256 55  
## 39 19 3.534173 2.459876 1.3052632 0.7549651 372 47  
## 40 16 3.514403 2.240388 1.4012346 0.9125996 311 58  
## 41 14 4.349353 2.639446 1.2883436 0.7889103 454 52  
## 42 31 4.366972 2.542581 2.2258065 2.0060609 322 64  
## 43 13 4.341549 2.704892 1.0843373 0.8877319 425 101  
## 44 18 4.235585 2.511690 2.7051282 1.8681471 400 73  
## 45 11 3.860795 2.583920 1.1680000 0.4855677 309 62  
## 46 9 3.460548 2.287079 1.1610169 0.4312052 351 10  
## 47 19 4.100707 2.162669 1.6447368 1.1888069 330 55  
## 48 9 4.003257 2.537020 2.6315789 1.4585691 585 12  
## 49 11 3.965392 2.431706 0.7536232 0.9387904 326 14  
## 50 10 3.241422 2.096168 0.0156250 0.1240196 444 2  
## 51 20 4.676101 2.442830 2.7600000 2.0254382 322 34  
## 52 11 3.650235 2.613115 1.0000000 0.0000000 281 3  
## 53 11 3.474725 2.095238 1.0588235 0.2352941 284 6  
## 54 10 3.886842 2.583416 1.3333333 0.7453560 370 4  
## 55 8 3.943636 2.819817 0.5000000 0.5000000 388 2  
## 56 18 NA NA NA NA NA NA  
## 57 14 NA NA NA NA NA NA  
## 58 16 NA NA NA NA NA NA  
## 59 28 NA NA NA NA NA NA  
## 60 20 NA NA NA NA NA NA  
## tokens\_MOT tokens\_CHI  
## 1 1731 1294  
## 2 3076 1249  
## 3 2576 1079  
## 4 2895 1010  
## 5 2227 921  
## 6 2510 897  
## 7 1959 864  
## 8 2044 847  
## 9 1802 793  
## 10 2171 738  
## 11 1764 719  
## 12 3072 713  
## 13 2264 710  
## 14 1741 702  
## 15 999 698  
## 16 2586 686  
## 17 1138 666  
## 18 1460 659  
## 19 1701 646  
## 20 1498 640  
## 21 2221 618  
## 22 2438 611  
## 23 1540 605  
## 24 2389 590  
## 25 2508 571  
## 26 1978 568  
## 27 3182 538  
## 28 1788 530  
## 29 2347 517  
## 30 2564 460  
## 31 2887 436  
## 32 2363 418  
## 33 1395 410  
## 34 1866 395  
## 35 1024 358  
## 36 1565 306  
## 37 536 300  
## 38 995 274  
## 39 1748 236  
## 40 1481 210  
## 41 2391 204  
## 42 1352 197  
## 43 2219 195  
## 44 2271 189  
## 45 1349 143  
## 46 1918 137  
## 47 1987 110  
## 48 2202 100  
## 49 1852 79  
## 50 2450 64  
## 51 1371 61  
## 52 1454 37  
## 53 1390 36  
## 54 1396 8  
## 55 2077 2  
## 56 NA NA  
## 57 NA NA  
## 58 NA NA  
## 59 NA NA  
## 60 NA NA

#descenting from highest to lowest  
  
#2. showing the ranking  
kids5 <- ordered\_data %>%  
 filter(VISIT == 1) %>%  
 arrange(tokens\_CHI)  
unique(kids5)

## SUBJ VISIT Diagnosis Ethnicity Gender Age ADOS1 nonVerbalIQ1  
## 1 63 1 ASD white F 41.07 15 28  
## 2 35 1 ASD white M 34.27 21 21  
## 3 39 1 TD white M 19.20 3 19  
## 4 34 1 ASD white M 36.53 12 31  
## 5 31 1 ASD white M 37.47 20 13  
## 6 10 1 ASD bangladeshi F 26.17 17 20  
## 7 37 1 ASD african american F 25.33 14 25  
## 8 19 1 ASD white M 34.80 14 25  
## 9 57 1 TD asian F 20.87 1 22  
## 10 55 1 ASD white M 31.63 17 28  
## 11 56 1 ASD white M 37.47 19 17  
## 12 62 1 ASD white M 35.50 14 27  
## 13 47 1 ASD lebanese M 24.90 13 27  
## 14 14 1 TD white M 20.07 0 30  
## 15 51 1 ASD white M 36.73 20 21  
## 16 45 1 TD white M 19.77 3 30  
## 17 12 1 TD white M 19.27 3 27  
## 18 46 1 TD white M 20.03 5 24  
## 19 27 1 TD white M 21.03 0 26  
## 20 17 1 TD white M 19.27 0 24  
## 21 6 1 TD white M 19.23 0 21  
## 22 30 1 ASD white/latino M 27.37 14 25  
## 23 16 1 TD white M 18.97 0 23  
## 24 42 1 ASD white/asian M 33.20 11 26  
## 25 60 1 TD white M 19.10 1 24  
## 26 33 1 ASD white M 34.87 17 26  
## 27 25 1 TD white F 18.93 0 21  
## 28 23 1 ASD african american M 27.53 21 22  
## 29 11 1 ASD white F 41.00 18 24  
## 30 5 1 TD white F 18.30 0 24  
## 31 2 1 TD white M 19.80 0 28  
## 32 4 1 TD white M 20.10 5 32  
## 33 61 1 TD white M 19.97 0 26  
## 34 67 1 ASD white M 42.00 15 27  
## 35 44 1 TD white M 19.37 0 24  
## 36 38 1 ASD white M 33.77 10 27  
## 37 9 1 ASD white M 34.03 9 34  
## 38 53 1 TD white F 20.03 0 27  
## 39 22 1 ASD white M 18.77 9 26  
## 40 21 1 ASD white M 35.80 11 28  
## 41 64 1 ASD white M 26.00 15 30  
## 42 43 1 TD white M 19.23 0 25  
## 43 65 1 TD white M 20.80 4 29  
## 44 15 1 TD white M 19.00 0 25  
## 45 18 1 TD white M 21.67 0 29  
## 46 3 1 TD white F 23.50 1 29  
## 47 36 1 TD white M 19.30 1 23  
## 48 8 1 ASD white/latino M 31.03 8 31  
## 49 54 1 TD white M 23.07 0 27  
## 50 29 1 TD white M 23.13 0 27  
## 51 58 1 TD white M 22.57 0 29  
## 52 41 1 ASD white M 39.50 7 33  
## 53 40 1 TD white F 19.87 1 29  
## 54 20 1 TD white M 20.77 0 29  
## 55 26 1 TD white M 21.77 0 29  
## 56 48 1 ASD white M 37.03 14 42  
## 57 7 1 ASD white M 28.80 13 34  
## 58 66 1 ASD white M 34.00 13 30  
## 59 59 1 TD white M 23.90 0 30  
## 60 32 1 ASD white M 30.40 11 32  
## 61 1 1 TD white M 18.07 15 NA  
## 62 13 1 ASD white F 31.77 0 28  
## 63 24 1 ASD latino M 35.47 19 26  
## 64 28 1 TD white M 19.93 0 20  
## 65 49 1 TD white M 23.13 0 27  
## 66 52 1 TD white M NA 4 17  
## VerbalIQ1 MOT\_MLU MOT\_LUstd CHI\_MLU CHI\_LUstd types\_MOT types\_CHI  
## 1 10 3.833770 2.417727 0.0000000 0.0000000 386 0  
## 2 9 3.686347 2.650865 1.5000000 0.5000000 228 3  
## 3 13 3.921109 2.376179 1.2307692 0.5756396 281 8  
## 4 13 3.607088 2.340376 0.9000000 0.4358899 366 11  
## 5 11 2.917355 2.073392 1.0833333 0.3996526 193 6  
## 6 17 2.618729 1.935874 1.0000000 0.0000000 212 4  
## 7 11 2.287293 1.928359 1.2500000 0.5747670 206 13  
## 8 11 3.182390 2.269630 1.0277778 0.1643355 281 9  
## 9 14 3.435743 2.257715 1.1818182 0.4575657 331 7  
## 10 10 3.765528 2.329794 1.2500000 0.6123724 303 17  
## 11 10 3.704110 2.276770 1.1000000 0.4898979 265 8  
## 12 11 2.548969 1.706620 1.0444444 0.2060804 195 9  
## 13 13 2.997050 2.202839 1.0175439 0.1312862 252 9  
## 14 16 4.195335 2.280551 1.0877193 0.2828862 235 17  
## 15 9 4.883966 2.773678 1.1666667 0.5000000 387 10  
## 16 20 3.088757 2.098655 1.2592593 0.4382281 275 9  
## 17 18 4.204846 2.384767 1.0375000 0.1899836 289 15  
## 18 20 2.776181 1.904606 0.5584416 0.7811664 258 20  
## 19 18 3.616867 2.261294 1.3661972 0.6763497 317 37  
## 20 15 3.967078 2.302921 1.1647059 0.4813476 277 27  
## 21 15 3.380463 2.214518 1.2168675 0.4121116 215 24  
## 22 19 3.024548 2.278008 1.4324324 1.1159904 328 41  
## 23 17 3.420315 2.273399 1.0396040 0.1950269 287 7  
## 24 19 3.298748 1.995123 1.2043011 0.5784507 274 36  
## 25 22 3.487871 2.432205 1.3139535 0.7664723 214 32  
## 26 14 3.304189 2.372505 1.0086207 0.0924466 295 6  
## 27 19 3.630476 2.452720 1.2371134 0.4926929 343 36  
## 28 8 4.390879 3.068753 1.0000000 0.0000000 485 8  
## 29 14 2.244755 1.510878 1.2641509 0.7177255 152 29  
## 30 18 3.544419 2.272387 1.0378788 0.1909031 363 36  
## 31 14 3.621993 2.164553 1.2522523 0.4739801 378 14  
## 32 31 3.265082 2.151724 1.5600000 1.2273549 288 59  
## 33 17 3.509138 2.189292 1.1846154 0.3879852 178 11  
## 34 16 3.030405 2.107521 1.0375000 0.1899836 303 15  
## 35 19 4.033333 2.460373 1.3034483 0.7818999 373 47  
## 36 22 2.743455 1.892987 1.3766234 0.6353026 214 67  
## 37 27 3.986357 2.500713 1.3947368 0.6897549 324 57  
## 38 20 3.943005 2.384506 1.0761421 0.3010798 260 13  
## 39 14 2.524740 2.141337 0.1857143 0.4564727 321 16  
## 40 20 2.539823 1.994618 1.3595506 0.7606501 283 89  
## 41 24 2.747100 1.853343 1.1809045 0.4454541 338 98  
## 42 17 3.093146 2.362590 1.0262009 0.2791346 333 32  
## 43 22 3.432387 2.124274 1.1830065 1.0317693 345 62  
## 44 17 3.960254 2.133094 1.5740741 0.6460283 329 69  
## 45 26 4.910190 2.496744 1.5988372 0.7206254 375 90  
## 46 18 3.757269 2.350833 1.8776978 0.9630619 334 51  
## 47 21 3.561364 2.229117 1.2641509 0.5546015 291 24  
## 48 27 3.459370 2.216235 2.0972222 1.2819226 379 102  
## 49 27 4.030075 2.695787 1.4258373 1.0238787 255 73  
## 50 22 3.494327 2.302219 1.5333333 1.0666667 322 91  
## 51 22 5.344227 2.917012 1.4086957 0.8433438 441 92  
## 52 26 4.135036 2.563076 0.4805825 0.7221272 381 39  
## 53 28 3.420975 2.244646 1.3322785 0.7633014 342 96  
## 54 33 3.587855 2.282255 1.7948718 1.0664941 467 108  
## 55 22 4.643200 2.619140 1.5827338 0.8388008 373 71  
## 56 27 3.341418 2.073518 1.9144981 1.3034604 271 101  
## 57 27 4.098446 2.612375 2.2768595 1.4437818 317 146  
## 58 30 3.604140 2.550110 2.8763441 1.9178784 400 149  
## 59 30 4.159159 2.509181 1.9397163 1.2603827 236 120  
## 60 33 4.690751 2.545488 3.4000000 1.7225773 278 119  
## 61 NA NA NA NA NA NA NA  
## 62 33 NA NA NA NA NA NA  
## 63 16 NA NA NA NA NA NA  
## 64 16 NA NA NA NA NA NA  
## 65 17 NA NA NA NA NA NA  
## 66 16 NA NA NA NA NA NA  
## tokens\_MOT tokens\_CHI  
## 1 2613 0  
## 2 927 3  
## 3 1631 16  
## 4 2054 21  
## 5 654 26  
## 6 761 29  
## 7 788 35  
## 8 1418 37  
## 9 1503 38  
## 10 2147 40  
## 11 1215 43  
## 12 955 47  
## 13 1827 58  
## 14 1262 62  
## 15 2144 63  
## 16 1417 68  
## 17 1808 83  
## 18 1188 91  
## 19 1361 95  
## 20 1643 99  
## 21 1136 101  
## 22 2138 103  
## 23 1625 105  
## 24 1537 109  
## 25 1118 109  
## 26 1643 117  
## 27 1698 118  
## 28 2826 122  
## 29 578 130  
## 30 1408 137  
## 31 1835 139  
## 32 1564 143  
## 33 1144 154  
## 34 1579 166  
## 35 2334 176  
## 36 893 180  
## 37 2859 197  
## 38 1347 212  
## 39 1787 214  
## 40 1019 227  
## 41 2084 233  
## 42 1547 235  
## 43 1805 244  
## 44 2139 249  
## 45 2585 254  
## 46 2674 260  
## 47 1344 260  
## 48 2009 269  
## 49 1452 286  
## 50 1870 319  
## 51 2267 319  
## 52 1988 337  
## 53 2035 398  
## 54 2555 406  
## 55 2740 433  
## 56 1591 450  
## 57 1428 461  
## 58 2587 469  
## 59 1170 473  
## 60 1450 483  
## 61 NA NA  
## 62 NA NA  
## 63 NA NA  
## 64 NA NA  
## 65 NA NA  
## 66 NA NA

USING SELECT

1. Make a subset of the data including only kids with ASD, mlu and word tokens
2. What happens if you include the name of a variable multiple times in a select() call?

#1.  
kids6 <- subset(ordered\_data, Diagnosis == "ASD", select = c("CHI\_MLU", "tokens\_CHI", "SUBJ"))   
  
kids6 <- subset(ordered\_data, Diagnosis == "ASD", select = c(Diagnosis, CHI\_MLU, tokens\_CHI))  
kids6

## Diagnosis CHI\_MLU tokens\_CHI  
## 26 ASD 2.2768595 461  
## 27 ASD 3.4530387 562  
## 28 ASD 3.1193182 983  
## 29 ASD 4.3023256 674  
## 30 ASD NA NA  
## 31 ASD 3.4135021 698  
## 32 ASD 2.0972222 269  
## 33 ASD 2.5630252 555  
## 34 ASD 3.5180723 490  
## 35 ASD 3.2571429 479  
## 36 ASD 4.0434783 539  
## 37 ASD 3.2781955 738  
## 38 ASD 1.3947368 197  
## 39 ASD 2.5743590 487  
## 40 ASD 2.5324074 468  
## 41 ASD 2.9072581 604  
## 42 ASD 2.4232804 404  
## 43 ASD 2.7665198 538  
## 44 ASD 1.0000000 29  
## 45 ASD 0.7606838 149  
## 46 ASD 1.1698113 62  
## 47 ASD 1.7142857 205  
## 48 ASD 1.6363636 349  
## 49 ASD 1.4967320 300  
## 50 ASD 1.2641509 130  
## 51 ASD 1.3211679 169  
## 52 ASD 1.3974359 210  
## 53 ASD 1.9115646 494  
## 54 ASD 1.5299145 194  
## 55 ASD NA NA  
## 68 ASD NA NA  
## 69 ASD NA NA  
## 100 ASD 1.0277778 37  
## 101 ASD 1.0125000 82  
## 102 ASD 1.0000000 58  
## 103 ASD NA NA  
## 104 ASD 1.0000000 9  
## 105 ASD 1.0000000 37  
## 112 ASD 1.3595506 227  
## 113 ASD 1.4867257 335  
## 114 ASD 1.9052632 330  
## 115 ASD 1.6179402 455  
## 116 ASD 1.5840000 197  
## 117 ASD 2.1450382 274  
## 118 ASD 0.1857143 214  
## 119 ASD 1.0800000 362  
## 120 ASD 1.6109325 503  
## 121 ASD 2.2485380 717  
## 122 ASD 2.2506739 792  
## 123 ASD 2.9027778 590  
## 124 ASD 1.0000000 122  
## 125 ASD 1.0400000 78  
## 126 ASD 2.0000000 96  
## 127 ASD 0.2727273 93  
## 128 ASD 0.2000000 5  
## 129 ASD 0.5000000 2  
## 130 ASD NA NA  
## 131 ASD NA NA  
## 132 ASD NA NA  
## 163 ASD 1.4324324 103  
## 164 ASD 1.4273504 157  
## 165 ASD 1.0863309 148  
## 166 ASD 1.0974026 162  
## 167 ASD 1.4133333 197  
## 168 ASD 1.3052632 236  
## 169 ASD 1.0833333 26  
## 170 ASD 1.0000000 32  
## 171 ASD 1.0000000 40  
## 172 ASD 1.2032520 143  
## 173 ASD 1.0000000 34  
## 174 ASD 1.1680000 143  
## 175 ASD 3.4000000 483  
## 176 ASD NA NA  
## 177 ASD 3.9196891 1293  
## 178 ASD 3.5238095 714  
## 179 ASD 3.2919897 1154  
## 180 ASD 3.3643411 1249  
## 181 ASD 1.0086207 117  
## 182 ASD 1.0114286 177  
## 183 ASD 0.6842105 39  
## 184 ASD 0.8947368 39  
## 185 ASD 0.2553191 179  
## 186 ASD 1.2883436 204  
## 187 ASD 0.9000000 21  
## 188 ASD 1.0265487 116  
## 189 ASD 1.0066667 151  
## 190 ASD 1.1250000 79  
## 191 ASD 0.4130435 193  
## 192 ASD 1.0843373 195  
## 193 ASD 1.5000000 3  
## 194 ASD 1.0000000 7  
## 195 ASD 1.0000000 1  
## 196 ASD 1.0000000 14  
## 197 ASD 1.0000000 33  
## 198 ASD 2.6315789 100  
## 205 ASD 1.2500000 35  
## 206 ASD 1.0000000 10  
## 207 ASD 1.0833333 26  
## 208 ASD 1.2100840 142  
## 209 ASD 0.6756757 42  
## 210 ASD 0.7536232 79  
## 211 ASD 1.3766234 180  
## 212 ASD 1.9481132 368  
## 213 ASD 1.8056872 365  
## 214 ASD 2.1962617 438  
## 215 ASD 2.5758929 537  
## 216 ASD 2.2745098 605  
## 224 ASD 0.4805825 337  
## 225 ASD 1.3432836 433  
## 226 ASD 1.8812665 694  
## 227 ASD 2.5159817 510  
## 228 ASD 2.7468750 815  
## 229 ASD 3.0774194 897  
## 235 ASD 1.2043011 109  
## 236 ASD 1.2571429 126  
## 237 ASD 1.8880000 232  
## 238 ASD 1.7000000 135  
## 239 ASD 1.6511628 340  
## 240 ASD 1.6447368 110  
## 253 ASD 1.0175439 58  
## 254 ASD 1.2195122 148  
## 255 ASD 2.0704225 404  
## 256 ASD 2.9900000 493  
## 257 ASD NA NA  
## 258 ASD 2.1586207 571  
## 259 ASD 1.9144981 450  
## 260 ASD 1.9832215 536  
## 261 ASD 2.3053892 1023  
## 262 ASD 2.6837838 822  
## 263 ASD 2.8665049 1054  
## 264 ASD 2.6794872 921  
## 278 ASD 1.1666667 63  
## 279 ASD 1.2258065 38  
## 280 ASD 1.0000000 3  
## 281 ASD 1.1200000 55  
## 282 ASD 1.0000000 32  
## 283 ASD 1.1610169 137  
## 298 ASD 1.2500000 40  
## 299 ASD 1.0000000 4  
## 300 ASD 1.0000000 45  
## 301 ASD 1.4102564 255  
## 302 ASD 1.4734513 306  
## 303 ASD 1.1000000 43  
## 304 ASD 1.3628319 147  
## 305 ASD 1.1666667 77  
## 306 ASD 1.0000000 70  
## 307 ASD 0.0000000 0  
## 308 ASD 1.3333333 8  
## 339 ASD 1.0444444 47  
## 340 ASD 1.0000000 19  
## 341 ASD 1.7500000 14  
## 342 ASD 1.0588235 36  
## 343 ASD 0.0000000 0  
## 344 ASD 1.0000000 44  
## 345 ASD 0.1621622 41  
## 346 ASD 0.0000000 1  
## 347 ASD 0.2000000 30  
## 348 ASD 0.0156250 64  
## 349 ASD 1.1809045 233  
## 350 ASD 1.4432990 260  
## 351 ASD 1.8000000 393  
## 352 ASD 2.0127796 574  
## 353 ASD 1.7591463 531  
## 354 ASD 2.1553398 618  
## 361 ASD 2.8763441 469  
## 362 ASD 2.7840000 670  
## 363 ASD 4.1318681 698  
## 364 ASD 3.3598326 693  
## 365 ASD 2.9655172 357  
## 366 ASD 3.4415584 713  
## 367 ASD 1.0375000 166  
## 368 ASD 1.0990991 356  
## 369 ASD 1.5828571 254  
## 370 ASD 1.3204225 334  
## 371 ASD 1.6688963 511  
## 372 ASD 1.4012346 210

#2.  
ordered\_data %>%  
 select(c("VISIT", "VISIT"))

## VISIT  
## 1 1  
## 2 1  
## 3 2  
## 4 3  
## 5 4  
## 6 5  
## 7 6  
## 8 1  
## 9 2  
## 10 3  
## 11 4  
## 12 5  
## 13 6  
## 14 1  
## 15 2  
## 16 3  
## 17 4  
## 18 5  
## 19 6  
## 20 1  
## 21 2  
## 22 3  
## 23 4  
## 24 5  
## 25 6  
## 26 1  
## 27 2  
## 28 3  
## 29 4  
## 30 5  
## 31 6  
## 32 1  
## 33 2  
## 34 3  
## 35 4  
## 36 5  
## 37 6  
## 38 1  
## 39 2  
## 40 3  
## 41 4  
## 42 5  
## 43 6  
## 44 1  
## 45 2  
## 46 3  
## 47 4  
## 48 5  
## 49 6  
## 50 1  
## 51 2  
## 52 3  
## 53 4  
## 54 5  
## 55 6  
## 56 1  
## 57 2  
## 58 3  
## 59 4  
## 60 5  
## 61 6  
## 62 1  
## 63 2  
## 64 3  
## 65 4  
## 66 5  
## 67 6  
## 68 1  
## 69 2  
## 70 1  
## 71 2  
## 72 3  
## 73 4  
## 74 5  
## 75 6  
## 76 1  
## 77 2  
## 78 3  
## 79 4  
## 80 5  
## 81 6  
## 82 1  
## 83 2  
## 84 3  
## 85 4  
## 86 5  
## 87 6  
## 88 1  
## 89 2  
## 90 3  
## 91 4  
## 92 5  
## 93 6  
## 94 1  
## 95 2  
## 96 3  
## 97 4  
## 98 5  
## 99 6  
## 100 1  
## 101 2  
## 102 3  
## 103 4  
## 104 5  
## 105 6  
## 106 1  
## 107 2  
## 108 3  
## 109 4  
## 110 5  
## 111 6  
## 112 1  
## 113 2  
## 114 3  
## 115 4  
## 116 5  
## 117 6  
## 118 1  
## 119 2  
## 120 3  
## 121 4  
## 122 5  
## 123 6  
## 124 1  
## 125 2  
## 126 3  
## 127 4  
## 128 5  
## 129 6  
## 130 1  
## 131 2  
## 132 3  
## 133 1  
## 134 2  
## 135 3  
## 136 4  
## 137 5  
## 138 6  
## 139 1  
## 140 2  
## 141 3  
## 142 4  
## 143 5  
## 144 6  
## 145 1  
## 146 2  
## 147 3  
## 148 4  
## 149 5  
## 150 6  
## 151 1  
## 152 2  
## 153 3  
## 154 4  
## 155 5  
## 156 6  
## 157 1  
## 158 2  
## 159 3  
## 160 4  
## 161 5  
## 162 6  
## 163 1  
## 164 2  
## 165 3  
## 166 4  
## 167 5  
## 168 6  
## 169 1  
## 170 2  
## 171 3  
## 172 4  
## 173 5  
## 174 6  
## 175 1  
## 176 2  
## 177 3  
## 178 4  
## 179 5  
## 180 6  
## 181 1  
## 182 2  
## 183 3  
## 184 4  
## 185 5  
## 186 6  
## 187 1  
## 188 2  
## 189 3  
## 190 4  
## 191 5  
## 192 6  
## 193 1  
## 194 2  
## 195 3  
## 196 4  
## 197 5  
## 198 6  
## 199 1  
## 200 2  
## 201 3  
## 202 4  
## 203 5  
## 204 6  
## 205 1  
## 206 2  
## 207 3  
## 208 4  
## 209 5  
## 210 6  
## 211 1  
## 212 2  
## 213 3  
## 214 4  
## 215 5  
## 216 6  
## 217 1  
## 218 2  
## 219 3  
## 220 4  
## 221 5  
## 222 6  
## 223 1  
## 224 1  
## 225 2  
## 226 3  
## 227 4  
## 228 5  
## 229 6  
## 230 2  
## 231 3  
## 232 4  
## 233 5  
## 234 6  
## 235 1  
## 236 2  
## 237 3  
## 238 4  
## 239 5  
## 240 6  
## 241 1  
## 242 2  
## 243 3  
## 244 4  
## 245 5  
## 246 6  
## 247 1  
## 248 2  
## 249 3  
## 250 4  
## 251 5  
## 252 6  
## 253 1  
## 254 2  
## 255 3  
## 256 4  
## 257 5  
## 258 6  
## 259 1  
## 260 2  
## 261 3  
## 262 4  
## 263 5  
## 264 6  
## 265 1  
## 266 2  
## 267 3  
## 268 4  
## 269 5  
## 270 6  
## 271 1  
## 272 2  
## 273 3  
## 274 4  
## 275 5  
## 276 1  
## 277 2  
## 278 1  
## 279 2  
## 280 3  
## 281 4  
## 282 5  
## 283 6  
## 284 1  
## 285 2  
## 286 1  
## 287 2  
## 288 3  
## 289 4  
## 290 5  
## 291 6  
## 292 1  
## 293 2  
## 294 3  
## 295 4  
## 296 5  
## 297 6  
## 298 1  
## 299 2  
## 300 4  
## 301 5  
## 302 6  
## 303 1  
## 304 2  
## 305 3  
## 306 4  
## 307 5  
## 308 6  
## 309 1  
## 310 2  
## 311 3  
## 312 4  
## 313 5  
## 314 6  
## 315 1  
## 316 2  
## 317 3  
## 318 4  
## 319 5  
## 320 6  
## 321 1  
## 322 2  
## 323 3  
## 324 4  
## 325 5  
## 326 6  
## 327 1  
## 328 2  
## 329 3  
## 330 4  
## 331 5  
## 332 6  
## 333 1  
## 334 2  
## 335 3  
## 336 4  
## 337 5  
## 338 6  
## 339 1  
## 340 2  
## 341 3  
## 342 6  
## 343 1  
## 344 2  
## 345 3  
## 346 4  
## 347 5  
## 348 6  
## 349 1  
## 350 2  
## 351 3  
## 352 4  
## 353 5  
## 354 6  
## 355 1  
## 356 2  
## 357 3  
## 358 4  
## 359 5  
## 360 6  
## 361 1  
## 362 2  
## 363 3  
## 364 4  
## 365 5  
## 366 6  
## 367 1  
## 368 2  
## 369 3  
## 370 4  
## 371 5  
## 372 6

#it only gets included once

USING MUTATE, SUMMARISE and PIPES 1. Add a column to the data set that represents the mean number of words spoken during all visits. 2. Use the summarise function and pipes to add an column in the data set containing the mean amount of words produced by each trial across all visits. HINT: group by Child.ID 3. The solution to task above enables us to assess the average amount of words produced by each child. Why don't we just use these average values to describe the language production of the children? What is the advantage of keeping all the data?

#1.   
ordered\_data <- ordered\_data %>%  
 mutate(mean\_spokenwords = mean(tokens\_CHI, na.rm = T))  
  
#2.   
ordered\_data <- ordered\_data %>%  
 group\_by(SUBJ) %>%  
 mutate(mean\_CHIwords = mean(tokens\_CHI, na.rm = T))  
  
#the advantage would be to remain the progress of the children's language