Data analysis: Investigating the relatioship between parental input frequency and children's vocabulary acquisition

Mijin Gwak

1. Determining age of acquisition

Preliminaries:

```
library(wordbankr) # Wordbank data
library(tidyverse) # for general use
library(knitr) # for tables and formatting
library(kableExtra)
```

1.1. Prepare the data

Download data from Word Bank:

```
# CDI:Words and Gestures (WG)

Admin_WG <- get_administration_data(language = "Korean", form = "WG")
Inst_WG <- get_instrument_data(language = "Korean", form = "WG")
Item_WG <- get_item_data(language = "Korean", form = "WG")

#CDI: Words and Sentences (WS)

Admin_WS <- get_administration_data(language = "Korean", form = "WS")
Inst_WS <- get_instrument_data(language = "Korean", form = "WS")
Item_WS <- get_item_data(language = "Korean", form = "WS")
```

Merge all the information (i.e., admin, instrument, and item):

```
WG <- merge(Admin_WG, Inst_WG) %>%
  merge(., Item_WG) %>%
  filter(type == "word") %>% # select words sector
  select(age, sex, mom_ed, birth_order, num_item_id, value)

WS <- merge(Admin_WS, Inst_WS) %>%
  merge(., Item_WS) %>%
  filter(type == "word") %>%
  select(age, sex, mom_ed, birth_order, num_item_id, value)
```

1.2. Inspecting lexical items in each dataset

Numbers of words in WG and WS datasets, respectively:

```
n_distinct(WG$num_item_id)
```

```
## [1] 284
```

```
n_distinct(WS$num_item_id)
```

```
## [1] 641
```

WG:

```
lexical_items_WG <- Item_WG %>%
  filter(type == "word") %>%
  select(definition, lexical_category)

head(lexical_items_WG) %>%
  kable(caption = "First parts of lexical items from CDI:WG dataset") %>%
  kable_styling()
```

First parts of lexical items from CDI:WG dataset

definition	lexical_category
꼬끼오	other
꽝꽝	other
꽥꽥	other
꿀꿀	other
똑똑	other
멍멍	other

WS:

```
lexical_items_WS <- Item_WS %>%
  filter(type == "word") %>%
  select(definition, lexical_category)

# change the column name uni_lemma to translation

names(lexical_items_WS)[names(lexical_items_WS) == "uni_lemma"] <- 'translation'

head(lexical_items_WS) %>%
  kable(caption = "First parts of lexical items from CDI:WS dataset") %>%
  kable_styling()
```

First parts of lexical items from CDI:WS dataset

definition	lexical_category
꽥꽥	other
꿀꿀	other
똑똑	other
멍멍	other
빵빵	other
야옹	other

1.3. Descriptive statistics

1.3.1. Total numbers of children

Get the total numbers of children in WG and WS datasets, respectively, by counting the rows:

```
nrow(Admin_WG)
```

```
## [1] 618
```

```
nrow(Admin_WS)
```

```
## [1] 1370
```

1.3.2. Age

1.3.2.1. Data screening

Numbers of participants without age information in WG and WS datasets, respectively:

```
sum(is.na(Admin_WG$age))
```

```
## [1] 0
```

```
sum(is.na(Admin_WS$age))
```

```
## [1] 0
```

1.3.2.2. Minimum, maximum, mean, and standard deviation of age

WG:

```
Admin_WG %>%
  summarise(
    Minimum = min(age),
    Maximum = max(age),
    Mean = mean(age),
    SD = sd(age)
  ) %>%
  kable() %>%
  kable_styling()
```

Minimum	Maximum	Mean	SD
8	17	12.67314	2.781618

WS:

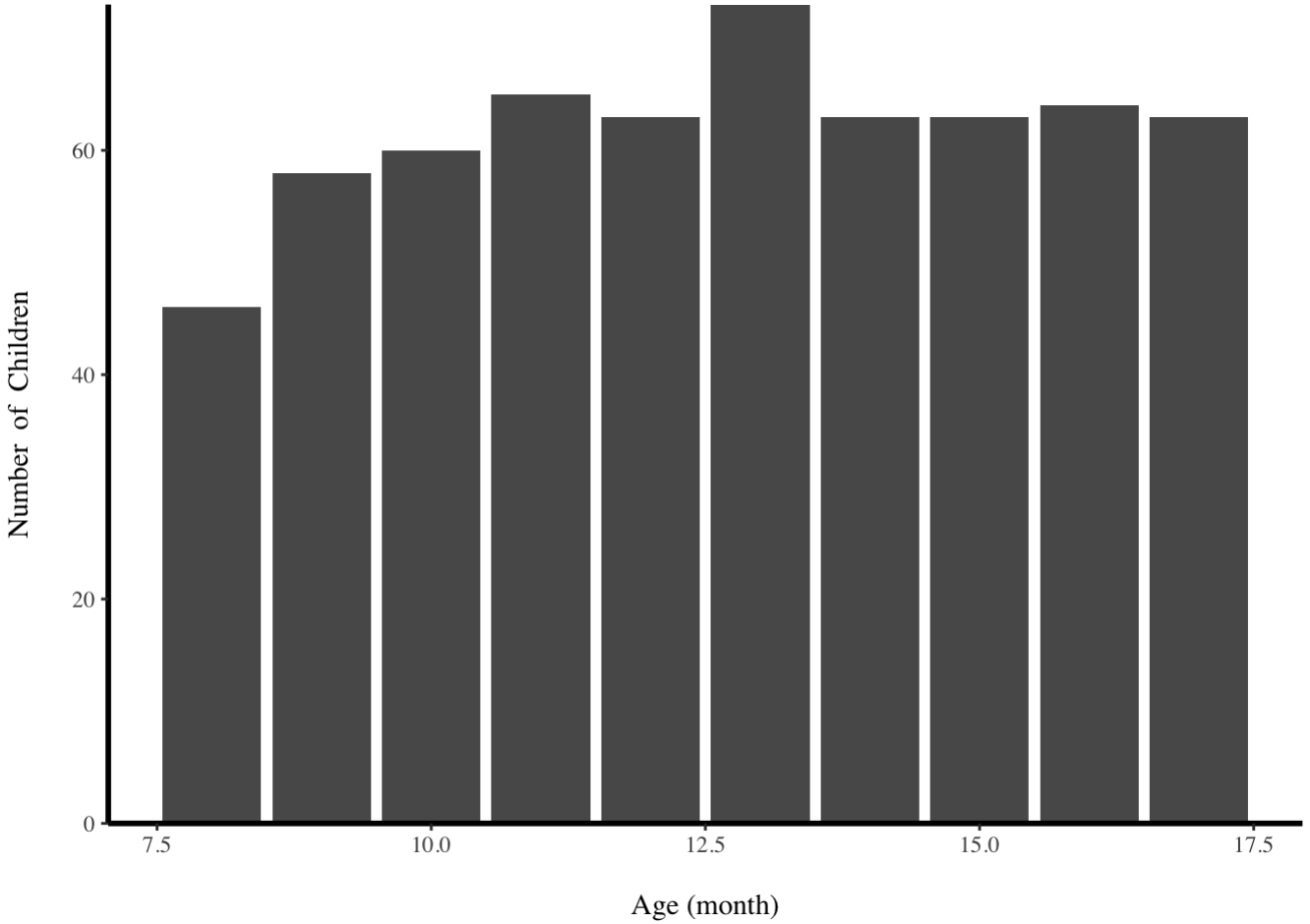
```
Admin_WS %>%
  summarise(
    Minimum = min(age),
    Maximum = max(age),
    Mean = mean(age),
    SD = sd(age)
  ) %>%
  kable() %>%
  kable_styling()
```

Minimum	Maximum	Mean	SD
18	36	27.15401	5.361341

1.3.3. Age Distribution

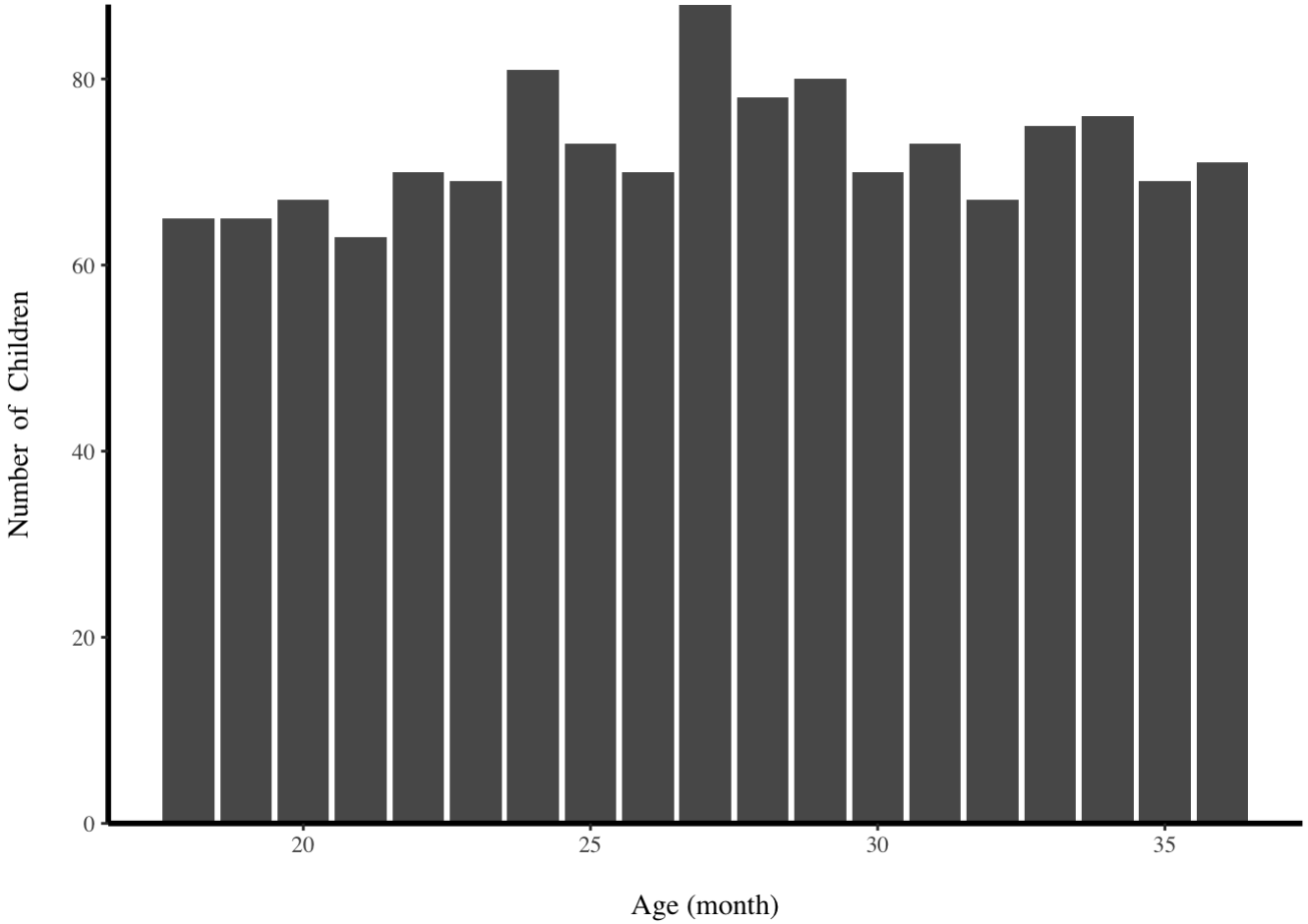
WG:

```
ggplot(Admin_WG, aes(x = age)) +
  geom_bar() +
  scale_y_continuous(expand = c(0,0)) + # set the bars to start from 0
  labs(x = paste(" " , "Age (month)", sep="\n"),
       y = paste("Number  of  Children", " ", sep="\n")) +
  theme(text=element_text(family="serif"),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(colour = "black", size = 1),
        axis.line.y = element_line(colour = "black", size = 1))
```



WS:

```
ggplot(Admin_WS, aes(x = age)) +
  geom_bar() +
  scale_y_continuous(expand = c(0,0)) +
  labs(x = paste(" " , "Age (month)", sep="\n"),
       y = paste("Number  of  Children", " ", sep="\n")) +
  theme(text=element_text(family="serif"),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(colour = "black", size = 1),
        axis.line.y = element_line(colour = "black", size = 1))
```



1.3.4. Gender

WG:

```
table(Admin_WG$sex) %>%
  kable(col.names = c("Gender", "Number of Children")) %>%
  kable_styling()
```

Gender	Number of Children
Female	317
Male	301
Other	0

WS:

```
table(Admin_WS$sex) %>%
  kable(col.names = c("Gender", "Number of Children")) %>%
  kable_styling()
```

Gender	Number of Children
Female	668
Male	700
Other	0

1.3.5. Background information

1.3.5.1. Mothers' education

Data screening

Number of participants without the mother's education information in WG and WS datasets, respectively:

```
sum(is.na(Admin_WG$mom_ed))
```

```
## [1] 588
```

```
sum(is.na(Admin_WS$mom_ed))
```

```
## [1] 1238
```

Information on the mother's education is not available for every child. Therefore, this information will not be removed:

```
WG$mom_ed <- NULL
WS$mom_ed <- NULL
```

1.3.5.2. Order of birth

Data screening

Numbers of participants without birth-order information in WG and WS datasets, respectively:

```
sum(is.na(Admin_WG$birth_order))
```

```
## [1] 588
```

```
sum(is.na(Admin_WS$birth_order))
```

```
## [1] 1238
```

Information on the order of birth is not available for every child. Therefore, this information will not be removed:

```
WG$birth_order <- NULL
WS$birth_order <- NULL
```

1.4. Determining age of acquisition

1.4.1. Prepare the data

WG:

```
# Select relevant variables and pivot to count the values (i.e., understands, produces, or none)

analysis_WG <- WG %>%
  select(num_item_id, value, age) %>%
  group_by(age) %>%
  mutate(row = row_number()) %>% # to protect duplicate rows when pivoting
  pivot_wider(names_from = age, values_from = value) %>%
  select(-row)

analysis_WG[is.na(analysis_WG)] <- "" # replace NAs with empty string

analysis_WG <- analysis_WG[order(analysis_WG$num_item_id),] # Change the order by item number
```

WS:

```
analysis_WS <- WS %>%
  select(num_item_id, value, age) %>%
  group_by(age) %>%
  mutate(row = row_number()) %>% # to protect duplicate rows when pivoting
  pivot_wider(names_from = age, values_from = value) %>%
  select(-row)
```

1.4.2. Creat a function to determine age of acquisition

1.4.2.1. Arguments of this function

analysis: dataframe produced in the data preparation (i.e., analysis_WG, analysis_WS)
num_children_by_age: number of children per age group which can be calculated by counting the ages from Admin dataset
num_items: number of lexical items which was calculated previously

The age of acquisition in this study is defined as “the youngest age in which 50% of the age group either understands or produces the word”. The function determines the age of acquisition by counting “understands” and “produces” values per age and per item. If the half of an age group either understands or produces, the word is considered “acquired” at that age. It starts counting from youngest to the oldest and stops once this condition is fulfilled at a certain age, and proceeds with the next item.

1.4.2.2. Function

```r
determine_AoA <- function(analysis, num_children_by_age, num_items){

  # create empty dataframe to which determined age of acquisition will be appended
  AoA <- data.frame(aoa = numeric(0))

  # loop through all the items
  for (i in num_items) {

    df <- analysis[analysis$num_item_id == i, ]

    # extract column names to be used to loop through all columns
    col_names <- as.character(sort(as.numeric(colnames(df)[-1])))

    # loop through all the columns
    for (column in col_names) {

      # From the Admin data, we can check the number of participants
      # of certain age navigated by the row number.
      # This row number can be identified by which() function which
      # returns the position of the value: the age
      row_num <- which (num_children_by_age$Var1 == column)
      num_children <- num_children_by_age[row_num, 2]

      # Count each value of the column (i.e., understands, produces)
      count <- table(df[column])

      # Convert it into dataframe for convenience of the analysis
      df_count <- as.data.frame(count)
      colnames(df_count)[1] <- "value"

      if (nrow(df_count) == 1) { # meaning only empty values
        next
      }

      else if (nrow(df_count) == 2) { # meaning either understands or produces

        if ("understands" %in% df_count$value) {
          understand_produce <- count[["understands"]][1]
        }
        else {
          understand_produce <- count[["produces"]][1]
        }
```

```
      }

      else { # meaning both understands and produces
        understand_produce <- count[["understands"]][1] + count[["produces"]][1]
      }

      if (column == col_names[length(col_names)]) { # last age group

        # If last age group does not fulfil the condition, assign NA
        if (understand_produce < num_children/2) {
          AoA[i, ] <- NA
        }

        else {
          aoa <- as.numeric(column)
          AoA[i, ] <- aoa
        }

      }

      else{

        # Go to the next column
        # if the count is smaller than half of the number of children
        if (understand_produce < num_children/2) {
          next
          }

        # If the condition is already fulfilled,
        # go to the next item
        else {
          aoa <- as.numeric(column)
          AoA[i, ] <- aoa
          break
        }
      }
    }
  }
  return(AoA)
}
```

1.4.3. Apply the function to the datasets

WG:

```
num_children_by_age_WG <- as.data.frame(table(Admin_WG$age))
num_items_WG <- 1:284

AoA_WG <- determine_AoA(analysis_WG, num_children_by_age_WG, num_items_WG) %>%
  cbind(lexical_items_WG, .) # append the item info
```

First parts of the dataframe

definition	lexical_category	aoa
꼬끼오	other	17
꽝꽝	other	NA
꽥꽥	other	16
꿀꿀	other	16
똑똑	other	16
멍멍	other	12

WS:

```
num_children_by_age_WS <- as.data.frame(table(Admin_WS$age))
num_items_WS <- 1:641

AoA_WS <- determine_AoA(analysis_WS, num_children_by_age_WS, num_items_WS) %>%
  cbind(lexical_items_WS, .) # append the item info
```

First parts of the dataframe

definition	lexical_category	aoa
꽥꽥	other	21
꿀꿀	other	20
똑똑	other	21
멍멍	other	18
빵빵	other	18
야옹	other	18

1.5. Inspecting the results

1.5.1. Handling missing values

Check items of which the age of acquisition were not determined for WG and WS datasets, respectively:

```
sum(is.na(AoA_WG$aoa))
```

```
## [1] 136
```

```
sum(is.na(AoA_WS$aoa))
```

```
## [1] 29
```

Age of acquisition of 136 out of 284 items could not be determined in WG dataset. This means that the half of the oldest age group, which is 17 months, could not produce these words. Likewise, age of acquisition of 29 out of 641 items could not be determined in WG dataset. The big difference of the number of undetermined age of acquisition is attributed to the age difference: WG dataset involves children aged from 8 months

to 17 months whose vocabulary size is smaller than that of children in WS dataset. The age of each age groups are indicated below for WG and WS datsets, respectively:

```
sort(unique(Admin_WG$age))
```

```
##  [1]  8  9 10 11 12 13 14 15 16 17
```

```
sort(unique(Admin_WS$age))
```

```
##  [1] 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
```

For those items of which the age of acquisition is undetermined, the oldest age + 1 was assigned considering the age gap between each age group (1 month) in both datasets. Therefore, 18 for WG and 37 for WS datasets, respectively will be assigned to the missing values. This is to indicate that those words are acquired in later age:

```
AoA_WG$aoa[is.na(AoA_WG$aoa)] <- 18
AoA_WS$aoa[is.na(AoA_WS$aoa)] <- 37
```

1.5.2. Data cleansing

To detect the lexical items in the corpus in later analysis, the data needs to be cleansed. For instance, some lexical items have parentheses, dashes, an alphabet, and/or punctuation for the purpose of providing contexts which will be removed in this section. In addition, synonyms will be divided into two rows so that they can be recognized separately in the corpora.

Generate a function to clean the data:

```
clean_items <- function(df_AoA){
  # create an empty dataframe to which cleansed items will be appended
  items_cleaned <- data.frame(definition = character(0))

  # extract translation, lexical category, and aoa
  # from previous df to be combined with the cleaned items

  other_info <- df_AoA %>%
    select(lexical_category, aoa)

  # remove all words in parentheses along with the parentheses
  for(i in 1:nrow(df_AoA)){
    new_input <- gsub("\\s*\\([^\\)]+\\)","", df_AoA$definition[i])
    items_cleaned[i, ] <- new_input
  }

  # combine the dataframe with other info and separate synonyms
  input_items <-  cbind(items_cleaned, other_info) %>%
    separate_rows(., 1, sep = "/")

  # remove punctuation and an alphabet that stands alone for providing contexts
  input_items$definition <- input_items$definition %>%
    gsub("[[:punct:]]", "", .) %>%
    gsub('�??', '', .)

  return(input_items)
}
```

Apply:

```
AoA_WG <- clean_items(AoA_WG)
AoA_WS <- clean_items(AoA_WS)
```

1.5.3. Removing homonyms

The meaning of homonyms are not distinguishable in this analysis since it requires context. Therefore, they will be removed:

```
# extract rows with duplicate definition
homonyms_WG <- AoA_WG[duplicated(AoA_WG$definition) | duplicated(AoA_WG$definition, fromLast = TRUE), ]
homonyms_WS <- AoA_WS[duplicated(AoA_WS$definition) | duplicated(AoA_WS$definition, fromLast = TRUE), ]

# remove those rows
AoA_WG <- AoA_WG[!(AoA_WG$definition %in% homonyms_WG$definition), ]
AoA_WS <- AoA_WS[!(AoA_WS$definition %in% homonyms_WS$definition), ]
```

Numbers of items after removing the homonyms for WG and WS datesets, respectively:

```
nrow(AoA_WG)
```

```
## [1] 285
```

```
nrow(AoA_WS)
```

```
## [1] 636
```

1.5.4. Merging two datasets

```
all_CDI <- rbind(AoA_WG, AoA_WS)
```

1.5.5. Handling duplicate items

When duplicate items are detected, item with older age of acquisition will be removed:

```
all_CDI <- all_CDI[order(all_CDI$aoa), ]
all_CDI <- all_CDI[!duplicated(all_CDI$definition), ]
```

1.5.6. Handling multi-word forms

Multi-word forms were removed (e.g., sing a song) since it is out of the scope of this study:

```
# find the words that contains spaces indicating multiword form
find_multiword <- grepl(' ', all_CDI$definition)

# extract the row numbers that contain multiword forms
row_multiword<- which (find_multiword == TRUE)

# remove those rows
all_CDI <- all_CDI[-row_multiword, ]
```

1.6. Result

Dataset:

First parts of the dataframe

definition	lexical_category	aoa
맘마	nouns	8
아빠	other	8
엄마	other	8
까꿍	other	8
지지	other	8
만세	other	9

2. Determining parental input frequency in child-directed speech dataset

For this analysis, we will interact with Python to extract mother and father tiers from the corpus and counting the lexical items. All the corpus from different studies was merged into one text file, and then was used for the analysis.

Preliminaries:

```
library(reticulate)
```

2.1. Python code chunks

Extract the utterances of mother and father:

```
import string

# import the corpus
file_dir = "/Users/mijin/Desktop/Thesis/"
with open(file_dir + "all_corpus.txt", "r") as f:
  corpus = f.read()

# split by lines
corpus_split = corpus.split("\n")

# create an empty list to which the extracted sentences will be appended
parental_input_split = []

for line in corpus_split:

  # maternal input
  if "*MOT" in line:
    new_line = line.replace("*MOT:\t", "")

    # remove punctuation and replace two spaces to one
    new_line = new_line.translate(str.maketrans('', '', string.punctuation))\
    .replace("  ", " ")

    # append it to the list
    parental_input_split.append(new_line)

  # paternal input
  elif "*FAT" in line:
    new_line = line.replace("*FAT:\t", "")
    new_line = new_line.translate(str.maketrans('', '', string.punctuation))\
    .replace("  ", " ")
    parental_input_split.append(new_line)

  else:
    pass
```

Total number of word tokens:

```
# Break down the lis of sentences into list of word tokens
word_tokens = [word for line in parental_input_split for word in line.split()]

# Count the word tokens
len(word_tokens)
```

```
## 401359
```

Count the frequency of occurrence of the lexical items:

```
# merge the lines into one string
parental_input = " ".join(parental_input_split)

# extract the lexical items from all_CDI
lexical_items = r.all_CDI['definition']

# create an empty list to which the frequency will be appended
freq_list = []

# count the items in the corpus and append the frequency to the list
for item in lexical_items:
  freq = parental_input.count(item)
  freq_list.append(freq)
```

2.2. Result

Append the frequency information to the existing dataframe:

```
all_CDI$freq_CHILDES <- py$freq_list
```

First parts of the dataframe

definition	lexical_category	aoa	freq_CHILDES

definition	lexical_category	aoa	freq_CHILDES
맘마	nouns	8	127
아빠	other	8	1691
엄마	other	8	4853
까꿍	other	8	251
지지	other	8	182
만세	other	9	31

3. Determining word frequency in adults dataset

A survey on the word frequency of modern Korean (National Institute of Korean Language, 2002) will be employed. The survey includes the word frequency in spoken language for adults. Thus, its correlation with the order of acquisition can be presented as a baseline against which the correlation between parental input frequency and the order of acquisition will be compared. As the previous analysis, Python will be used.

```
definitions <- as.character(all_CDI$definition)
```

3.1. Python code chunks

Preliminary:

```
import pandas as pd
```

Import data:

```
# import excel file
adult_freq_xls = pd.ExcelFile('frequency.xls')
adult_freq = pd.concat([
  pd.read_excel(adult_freq_xls, '고유명사'),
  pd.read_excel(adult_freq_xls, '조사'),
  pd.read_excel(adult_freq_xls, '어미'),
  pd.read_excel(adult_freq_xls, '단어')
])

# extract relevant columns: definition and frquency
adult_freq = adult_freq[['항목', '빈도']]

# lexical items in this analysis
definitions = r.definitions
```

Extract frequency data according to the lexical items:

```
# Create empty lists for definitions and frequency list
# to which information from the adults dataset will be appended
adult_definitions_list = []
adult_freq_list = []

# Extract frequency data according to the lexical items
for row in adult_freq['항목']:
  for definition in definitions:

    if row in definition:

      row_num = int(adult_freq[adult_freq['항목'] == row].index[0])
      adult_definitions_list.append(definition)
      adult_freq_list.append(adult_freq.iloc[row_num]['빈도'])

    else:
      pass

# Merge the lists into a dataframe
frequency_adults = pd.DataFrame(
  {
  'definition': adult_definitions_list,
  'freq_adults':adult_freq_list
  }
)

# add the frequencies of the items with the same meaning
frequency_adults = frequency_adults.groupby(['definition'], as_index = False)['freq_adults'].sum()
```

Append the frequency information to the existing dataframe:

```
all_CDI <- all_CDI %>%
  full_join(., py$frequency_adults, by = "definition")

# Replace NA with 0
all_CDI$freq_adults[is.na(all_CDI$freq_adults)] <- 0
```

3.2. Final dataset

definition	lexical_category	aoa	freq_CHILDES	freq_adults
맘마	nouns	8	127	90
아빠	other	8	1691	6
엄마	other	8	4853	98
까꿍	other	8	251	26
지지	other	8	182	15
만세	other	9	31	55
할머니	other	10	668	268
곤지곤지	other	10	28	102
잼잼	other	10	14	2
짝짜꿍	other	10	64	5
목욕	other	10	91	3
쉬	other	10	304	0
아니	other	10	1692	268
치카	other	10	204	1
양치	other	10	8	55
하지마	other	10	22	117
공	nouns	11	1086	18
과자	nouns	11	141	856

definition	lexical_category	aoa	freq_CHILDES	freq_adults
까까	nouns	11	87	25
물	nouns	11	2461	0
응가	other	11	63	2195
똥	other	11	247	1
주세요	predicates	11	627	236
멍멍	other	12	575	0
자동차	nouns	12	409	318
개	nouns	12	1911	0
멍멍이	nouns	12	274	1644
기저귀	nouns	12	84	353
책	nouns	12	1669	0
숟가락	nouns	12	11	2213
전화	nouns	12	345	98
컵	nouns	12	122	2
텔레비전	nouns	12	19	124
밖	other	12	281	2
바깥	other	12	52	1
응	other	12	7993	0
안돼	other	12	18	30
누워	predicates	12	72	19

definition	lexical_category	aoa	freq_CHILDES	freq_adults
박수쳐	predicates	12	0	300
사랑해	predicates	12	116	193
안아	predicates	12	128	34
뜨거워	predicates	12	24	0
붕	other	13	415	0
얌얌	other	13	20	0
손	nouns	13	838	12
코	nouns	13	1324	1
모자	nouns	13	412	308
신	nouns	13	1083	74
신발	nouns	13	138	76
양말	nouns	13	88	55
인형	nouns	13	280	1
밥	nouns	13	326	1
빗	nouns	13	71	0
아가	other	13	423	2220
할아버지	other	13	510	25
도리도리	other	13	2	20
빠이빠이	other	13	13	1649
앉아	predicates	13	396	4

definition	lexical_category	aoa	freq_CHILDES	freq_adults
어흥	other	14	89	1162
배꼽	nouns	14	49	7
입	nouns	14	790	2
옷	nouns	14	304	0
꼬까	nouns	14	52	72
빵	nouns	14	546	0
우유	nouns	14	243	207
치즈	nouns	14	37	2
베개	nouns	14	16	1
꽃	nouns	14	513	0
세수	other	14	38	69
닫아	predicates	14	33	4
먹어	predicates	14	632	1162
써	predicates	14	226	0
일어나	predicates	14	138	1286
고마워	predicates	14	104	190
없어	predicates	14	574	1162
없다	predicates	14	124	376
예뻐	predicates	14	188	12
빵빵	other	15	129	0

definition	lexical_category	aoa	freq_CHILDES	freq_adults
머리	nouns	15	386	20
발	nouns	15	983	0
연필	nouns	15	92	4
귤	nouns	15	57	1
딸기	nouns	15	263	353
안경	nouns	15	57	34
칫솔	nouns	15	18	2
문	nouns	15	648	11
의자	nouns	15	98	1954
이불	nouns	15	63	1646
집	other	15	1435	0
뽀뽀	other	15	143	1
안녕	other	15	580	32
신어	predicates	15	37	1236
와	predicates	15	3597	540
싫어	predicates	15	245	1162
꽥꽥	other	16	59	5
꿀꿀	other	16	212	1
똑똑	other	16	200	0
야옹	other	16	466	436

definition	lexical_category	aoa	freq_CHILDES	freq_adults
음매	other	16	7	261
고양이	nouns	16	473	1799
물고기	nouns	16	228	479
사자	nouns	16	331	311
토끼	nouns	16	883	1
호랑이	nouns	16	244	1828
귀	nouns	16	512	0
얼굴	nouns	16	255	52
가방	nouns	16	96	2197
바지	nouns	16	112	15
블록	nouns	16	64	0
풍선	nouns	16	73	0
김	nouns	16	121	1025
바나나	nouns	16	112	35
사과	nouns	16	302	551
사탕	nouns	16	82	3
요구르트	nouns	16	10	217
비누	nouns	16	26	42
수건	nouns	16	19	157
시계	nouns	16	148	62

definition	lexical_category	aoa	freq_CHILDES	freq_adults
약	nouns	16	365	0
방	nouns	16	749	2
이모	other	16	192	1644
또	function_words	16	2326	2
나가	predicates	16	330	2225
내려가	predicates	16	45	2413
내려	predicates	16	276	218
넣어	predicates	16	259	1162
닦아	predicates	16	71	4
때려	predicates	16	7	218
벗어	predicates	16	64	1162
빗어	predicates	16	2	1162
씻어	predicates	16	15	1162
열어	predicates	16	130	1162
올라가	predicates	16	159	2203
웃어	predicates	16	21	1162
입어	predicates	16	38	1164
매워	predicates	16	25	25
아파	predicates	16	164	4
좋아	predicates	16	1219	5

definition	lexical_category	aoa	freq_CHILDES	freq_adults
꼬끼오	other	17	19	253
버스	nouns	17	301	0
돼지	nouns	17	420	16
고추	nouns	17	40	116
고기	nouns	17	426	452
열쇠	nouns	17	14	1
냉장고	nouns	17	32	161
이거	function_words	17	5378	1644
요거	function_words	17	408	48
꺼	predicates	17	662	0
들어	predicates	17	1249	1264
타	predicates	17	1405	0
무서워	predicates	17	181	102
꽝꽝	other	18	8	2
윙	other	18	102	0
칙칙폭폭	other	18	222	4
기차	nouns	18	260	358
불자동차	nouns	18	1	320
소방차	nouns	18	100	32
비행기	nouns	18	297	376

definition	lexical_category	aoa	freq_CHILDES	freq_adults
택시	nouns	18	59	64
개구리	nouns	18	327	58
곰	nouns	18	1071	2
기린	nouns	18	292	352
나비	nouns	18	177	52
닭	nouns	18	92	2
말	nouns	18	2087	4
벌	nouns	18	547	0
새	nouns	18	1066	0
소	nouns	18	1697	24
송아지	nouns	18	37	87
악어	nouns	18	143	1162
오리	nouns	18	121	222
쥐	nouns	18	635	0
코끼리	nouns	18	383	20
하마	nouns	18	86	99
다리	nouns	18	479	394
등	nouns	18	223	0
똥꼬	nouns	18	9	48
목	nouns	18	414	2

definition	lexical_category	aoa	freq_CHILDES	freq_adults
엉덩이	nouns	18	53	1646
이	nouns	18	45114	1644
이빨	nouns	18	224	1645
젖	nouns	18	79	1
팔	nouns	18	381	1
피	nouns	18	807	23
단추	nouns	18	21	37
장갑	nouns	18	27	60
팬티	nouns	18	30	14
반지	nouns	18	13	18
사진	nouns	18	174	25
종이	nouns	18	113	1644
종	nouns	18	2910	0
감	nouns	18	496	1
계란	nouns	18	43	108
달걀	nouns	18	22	1
고구마	nouns	18	12	239
국	nouns	18	278	0
김치	nouns	18	21	1025
껌	nouns	18	51	1

definition	lexical_category	aoa	freq_CHILDES	freq_adults
꿀	nouns	18	608	1
떡	nouns	18	211	0
라면	nouns	18	35	448
무	nouns	18	3277	0
무우	nouns	18	4	30
밤	nouns	18	244	0
수박	nouns	18	145	300
쌀	nouns	18	34	2
아이스크림	nouns	18	133	1844
오렌지	nouns	18	30	220
주스	nouns	18	57	45
초콜렛	nouns	18	29	16
콩	nouns	18	249	0
포도	nouns	18	79	2
가위	nouns	18	129	2200
돈	nouns	18	119	0
우산	nouns	18	103	32
치약	nouns	18	9	1
칼	nouns	18	52	11
침대	nouns	18	66	36

definition	lexical_category	aoa	freq_CHILDES	freq_adults
피아노	nouns	18	68	280
놀이터	other	18	52	1646
병원	other	18	102	45
학교	other	18	73	4
안	other	18	3935	30
나무	nouns	18	483	30
달	nouns	18	941	0
돌	nouns	18	1126	0
땅	nouns	18	253	2
별	nouns	18	242	2
불	nouns	18	802	2
비	nouns	18	1764	22
시소	nouns	18	15	86
하늘	nouns	18	220	12
고모	other	18	17	100
누나	other	18	50	75
삼촌	other	18	62	1
선생님	other	18	154	3
아저씨	other	18	496	7
아줌마	other	18	56	95

definition	lexical_category	aoa	freq_CHILDES	freq_adults
언니	other	18	367	266
오빠	other	18	296	206
친구	other	18	959	37
형	other	18	454	1
거기	function_words	18	725	352
나	function_words	18	8603	30
내	function_words	18	1573	0
누구	function_words	18	1574	55
뭐	function_words	18	6327	2
무엇	function_words	18	158	4
여기	function_words	18	4008	568
저거	function_words	18	112	0
다시	function_words	18	431	438
더	function_words	18	1564	61
많이	function_words	18	532	1647
빨리	function_words	18	350	21
공부해	predicates	18	8	26
그려	predicates	18	121	348
깎아	predicates	18	15	4
노래해	predicates	18	11	383

definition	lexical_category	aoa	freq_CHILDES	freq_adults
놀아	predicates	18	235	4
맞아	predicates	18	138	4
봐	predicates	18	4548	0
빼	predicates	18	237	0
안해	predicates	18	68	38
울어	predicates	18	131	1162
읽어	predicates	18	585	1162
간지러워	predicates	18	14	55
같아	predicates	18	288	4
똑같아	predicates	18	32	4
깜깜해	predicates	18	11	8
더워	predicates	18	24	61
많아	predicates	18	188	4
맛있어	predicates	18	116	1186
미워	predicates	18	22	185
아니야	predicates	18	767	705
있어	predicates	18	3221	1186
차가워	predicates	18	19	2200
추워	predicates	18	47	16
랑	function_words	18	2604	180

definition	lexical_category	aoa	freq_CHILDES	freq_adults
야	function_words	18	12870	436
말구	function_words	18	5	39
고	function_words	18	15925	100
게	function_words	18	10441	147
요	function_words	18	16592	48
지	function_words	18	17728	15
아기	other	18	700	356
짹짹	other	21	58	2
뽀뽀해	predicates	21	16	9
자전거	nouns	22	131	407
휴지	nouns	22	27	16
하나	function_words	22	1024	42
오토바이	nouns	24	49	1849
유모차	nouns	24	14	205
트럭	nouns	24	153	1
장난감	nouns	24	243	63
크레용	nouns	24	1	6
크레파스	nouns	24	34	1
거북이	nouns	24	120	1720
뱀	nouns	24	164	1

definition	lexical_category	aoa	freq_CHILDES	freq_adults
병아리	nouns	24	163	23
비둘기	nouns	24	9	375
양	nouns	24	1067	51
원숭이	nouns	24	327	1690
목걸이	nouns	24	13	1646
잠바	nouns	24	2	1
주머니	nouns	24	75	311
계단	nouns	24	23	21
소파	nouns	24	22	31
화장실	nouns	24	85	63
감자	nouns	24	18	309
당근	nouns	24	199	75
커피	nouns	24	84	25
케이크	nouns	24	81	2226
토마토	nouns	24	87	91
손가락	nouns	24	73	2226
걸레	nouns	24	19	0
그릇	nouns	24	31	0
쓰레기	nouns	24	73	354
쓰레기통	nouns	24	19	355

definition	lexical_category	aoa	freq_CHILDES	freq_adults
휴지통	nouns	24	2	17
젓가락	nouns	24	26	2213
청소기	nouns	24	20	379
포크	nouns	24	120	0
그네	nouns	24	69	90
미끄럼틀	nouns	24	140	188
햇빛	nouns	24	29	3
파이팅	other	24	5	1645
어디	function_words	24	2876	1168
걸어	predicates	24	87	1162
꺼내	predicates	24	109	0
다쳐	predicates	24	19	376
던져	predicates	24	46	484
들어가	predicates	24	294	3480
떄려	predicates	24	0	218
뛰어	predicates	24	169	1162
마셔	predicates	24	42	90
몰라	predicates	24	95	8
발라	predicates	24	24	12
불어	predicates	24	34	1164

definition	lexical_category	aoa	freq_CHILDES	freq_adults
일어서	predicates	24	34	1546
좋아해	predicates	24	138	13
줘	predicates	24	842	0
더러워	predicates	24	13	152
조용해	predicates	24	12	92
졸려	predicates	24	83	218
뒤	function_words	24	391	0
개미	nouns	25	107	185
다람쥐	nouns	25	163	408
벌레	nouns	25	244	2
여우	nouns	25	246	246
참새	nouns	25	42	0
펭귄	nouns	25	87	51
구두	nouns	25	103	37
운동화	nouns	25	9	5
목욕탕	nouns	25	26	5
세탁기	nouns	25	17	395
창문	nouns	25	54	14
국수	nouns	25	6	28
옥수수	nouns	25	48	32

definition	lexical_category	aoa	freq_CHILDES	freq_adults
콜라	nouns	25	6	22
피자	nouns	25	23	332
머리카락	nouns	25	41	39
발톱	nouns	25	52	1
입술	nouns	25	15	4
카메라	nouns	25	25	10
눈사람	nouns	25	20	36
동생	other	25	237	3
왜	function_words	25	1032	3
가져	predicates	25	229	2195
기다려	predicates	25	38	946
나와	predicates	25	247	570
넘어져	predicates	25	15	1162
미워해	predicates	25	0	193
비켜	predicates	25	51	28
숨어	predicates	25	72	1162
괜찮아	predicates	25	126	4
무거워	predicates	25	24	0
배고파	predicates	25	21	106
헬리콥터	nouns	26	41	20

definition	lexical_category	aoa	freq_CHILDES	freq_adults
선물	nouns	26	188	0
공룡	nouns	26	310	19
염소	nouns	26	54	24
치마	nouns	26	31	90
부엌	nouns	26	39	2
과일	nouns	26	86	620
배추	nouns	26	47	22
생선	nouns	26	68	1
얼음	nouns	26	57	273
호박	nouns	26	43	277
무릎	nouns	26	72	2
그림	nouns	26	336	7
지갑	nouns	26	8	15
슈퍼마켓	other	26	23	133
아파트	other	26	21	7
회사	other	26	27	3
의사	other	26	58	1649
떨어져	predicates	26	70	1162
말해	predicates	26	118	12
부어	predicates	26	8	1162

definition	lexical_category	aoa	freq_CHILDES	freq_adults
춤춰	predicates	26	1	0
풀어	predicates	26	37	1162
혼나	predicates	26	9	30
귀여워	predicates	26	29	216
시끄러워	predicates	26	12	68
재미있어	predicates	26	53	1371
어	function_words	26	32857	1162
앞	function_words	26	318	2
위	function_words	26	1110	5
이따가	other	26	15	3842
저기	function_words	26	390	352
로봇	nouns	27	109	353
비누방울	nouns	27	0	44
총	nouns	27	306	0
부엉이	nouns	27	49	1649
사슴	nouns	27	83	34
어깨	nouns	27	16	1162
구름	nouns	27	156	36
모래	nouns	27	102	122
바람	nouns	27	106	31

definition	lexical_category	aoa	freq_CHILDES	freq_adults
돼	other	27	1587	0
가게	other	27	110	2344
바다	other	27	176	378
산	other	27	478	1
시장	other	27	59	122
조금	function_words	27	217	98
둬	predicates	27	32	0
버려	predicates	27	37	218
맛없어	predicates	27	1	1162
힘들어	predicates	27	51	1264
까	function_words	27	5707	25
는	other	27	17664	1882
한테	other	27	661	250
에게	other	27	291	388
나중에	other	27	127	41
동물	nouns	28	363	5
목도리	nouns	28	14	49
서랍	nouns	28	24	103
닭고기	nouns	28	10	455
땅콩	nouns	28	8	3

definition	lexical_category	aoa	freq_CHILDES	freq_adults
복숭아	nouns	28	10	5
햄버거	nouns	28	12	2
몸	nouns	28	330	0
뺨	nouns	28	0	1
턱	nouns	28	13	0
혀	nouns	28	142	0
옷걸이	nouns	28	0	1728
접시	nouns	28	27	63
길	nouns	28	509	1
흙	nouns	28	96	2
동물원	other	28	69	52
유치원	other	28	213	223
너무	function_words	28	595	0
잘	function_words	28	2856	0
간지럽혀	predicates	28	2	31
꽂아	predicates	28	50	4
끼워	predicates	28	27	0
놓아	predicates	28	29	4
눌러	predicates	28	41	6
달려	predicates	28	182	218

definition	lexical_category	aoa	freq_CHILDES	freq_adults
당겨	predicates	28	20	80
덮어	predicates	28	18	1162
도와	predicates	28	91	542
두드려	predicates	28	4	244
떨어뜨려	predicates	28	3	1380
만들어	predicates	28	252	1278
먹여	predicates	28	17	216
멈춰	predicates	28	16	0
밀어	predicates	28	35	1163
보여	predicates	28	247	216
부딪쳐	predicates	28	4	0
붙여	predicates	28	441	216
빨아	predicates	28	26	4
사	predicates	28	3915	3
시작해	predicates	28	20	70
싸워	predicates	28	11	0
알아	predicates	28	171	4
운전해	predicates	28	19	105
잘라	predicates	28	79	8
잡아	predicates	28	325	4

definition	lexical_category	aoa	freq_CHILDES	freq_adults
찾아	predicates	28	455	4
태워	predicates	28	68	0
나빠	predicates	28	4	30
동그래	predicates	28	10	265
목말라	predicates	28	4	14
심심해	predicates	28	17	16
작아	predicates	28	49	4
조그맣다	predicates	28	0	517
커	predicates	28	667	0
거야	function_words	28	2493	436
테야	function_words	28	4	436
도	other	28	7167	2
하고	other	28	1843	329
밑	function_words	28	160	0
아래	function_words	28	66	126
옆	function_words	28	161	2
내일	other	28	75	78
아침	other	28	152	6
지금	other	28	477	29
우리	function_words	28	1902	48

definition	lexical_category	aoa	freq_CHILDES	freq_adults
싶다	function_words	28	24	378
옷장	nouns	29	6	60
유리	nouns	29	48	195
전자레인지	nouns	29	0	422
설탕	nouns	29	12	2
음식	nouns	29	61	224
허리	nouns	29	22	64
라디오	nouns	29	14	220
병	nouns	29	430	0
빗자루	nouns	29	12	366
상자	nouns	29	65	311
공원	other	29	24	63
은행	other	29	13	15
진짜	function_words	29	149	27
경찰	other	29	57	4
남자	other	29	34	374
사람	other	29	584	36
여자	other	29	68	524
어떻게	function_words	29	1914	1309
건너	predicates	29	52	128

definition	lexical_category	aoa	freq_CHILDES	freq_adults
고쳐	predicates	29	24	100
깨	predicates	29	373	0
끊어져	predicates	29	0	1162
다녀	predicates	29	116	376
돌려	predicates	29	73	218
물어	predicates	29	84	1162
불러	predicates	29	96	8
뺘앗아	predicates	29	0	4
쏟아	predicates	29	24	4
올려	predicates	29	77	218
이야기해	predicates	29	9	2527
치워	predicates	29	25	0
칠해	predicates	29	46	8
길어	predicates	29	63	1163
높아	predicates	29	16	4
빨개	predicates	29	7	0
빨라	predicates	29	34	8
어두워	predicates	29	3	1164
께	function_words	29	335	97
래	function_words	29	4372	122

definition	lexical_category	aoa	freq_CHILDES	freq_adults
금방	other	29	34	16
그거	function_words	29	762	1
너	function_words	29	1536	0
주다	function_words	29	2	420
공주	other	30	180	84
몇	function_words	30	508	6
쓰러져	predicates	30	9	6
잠옷	nouns	31	4	1
사이다	nouns	31	3	4073
소금	nouns	31	31	38
귀신	other	31	7	74
쏴	predicates	31	13	0
달라	predicates	31	83	8
멋있어	predicates	31	14	1186
니	function_words	31	7301	264
오늘	other	31	365	213
사다리	nouns	32	83	400
수영장	other	32	20	101
간호사	other	32	10	23
언제	function_words	32	112	84

definition	lexical_category	aoa	freq_CHILDES	freq_adults
자꾸	function_words	33	113	308
무슨	function_words	33	840	1
나눠가져	predicates	33	0	2225
떠	predicates	33	172	0
이겨	predicates	33	14	1650
일해	predicates	33	4	80
죽어	predicates	33	11	1162
찍어	predicates	33	58	1162
가벼워	predicates	33	5	2195
까매	predicates	33	3	50
이상해	predicates	33	28	1667
에서	other	33	1291	892
그런데	function_words	33	344	78
는데	function_words	33	1280	2340
낮	other	33	52	2
썰매	nouns	34	17	26
코트	nouns	34	4	2
외투	nouns	34	0	2
허리띠	nouns	34	6	65
도너츠	nouns	34	14	2

definition	lexical_category	aoa	freq_CHILDES	freq_adults
메론	nouns	34	7	0
핫도그	nouns	34	0	3
망치	nouns	34	32	0
못	nouns	34	515	0
쟁반	nouns	34	0	4
바위	nouns	34	107	5
지붕	nouns	34	42	16
일	other	34	2378	72
주유소	other	34	11	246
딸	other	34	410	0
소방수	other	34	1	58
어른	other	34	58	1162
왕자	other	34	110	309
어디서	function_words	34	138	1270
어떤	function_words	34	445	1163
끌어	predicates	34	37	1162
날라가	predicates	34	29	2203
돌아	predicates	34	373	4
부숴	predicates	34	5	0
빠져	predicates	34	42	0

definition	lexical_category	aoa	freq_CHILDES	freq_adults
뽑아	predicates	34	14	4
생각해	predicates	34	28	8
수영해	predicates	34	1	50
실어	predicates	34	19	1162
요리해	predicates	34	6	74
터져	predicates	34	9	1
틀어	predicates	34	21	1162
피어	predicates	34	18	1185
노래	predicates	34	214	375
멀어	predicates	34	10	1162
짜	predicates	34	643	0
착해	predicates	34	37	8
파래	predicates	34	0	122
구나	function_words	34	482	142
를	other	34	4021	1572
만	other	34	2991	14
에	other	34	10611	2
그래서	function_words	34	620	372
어저께	other	34	31	1263
어제	other	34	75	1246

definition	lexical_category	aoa	freq_CHILDES	freq_adults
저녁	other	34	42	6
보다	function_words	34	309	567
강	other	35	466	83
아주	function_words	35	399	50
전부	function_words	35	11	97
맛봐	predicates	35	1	0
쫓아	predicates	35	44	4
하얘	predicates	35	0	12
는다	function_words	35	73	2419
대	function_words	35	1649	36
로	other	35	5075	352
그리고	function_words	35	476	122
매일	other	35	28	103
맨날	other	35	76	0
당나귀	nouns	36	9	112
스파게티	nouns	36	5	161
군인	other	36	19	80
아들	other	36	156	108
우체부	other	36	2	62
살아	predicates	36	72	4

definition	lexical_category	aoa	freq_CHILDES	freq_adults
시간	other	36	145	78
점심	other	36	26	9
거위	nouns	37	42	5
지하실	nouns	37	0	3467
호떡	nouns	37	1	58
발목	nouns	37	1	2
깃발	nouns	37	3	0
잔디	nouns	37	4	7
교회	other	37	26	0
마당	other	37	29	172
숲	other	37	132	0
시골	other	37	19	64
절	other	37	98	0
도둑	other	37	34	2
어느	function_words	37	269	1162
가리켜	predicates	37	6	2219
끝내	predicates	37	3	3
날라	predicates	37	70	8
눌려	predicates	37	1	218
떼	predicates	37	147	0

definition	lexical_category	aoa	freq_CHILDES	freq_adults
말려	predicates	37	5	222
서둘러	predicates	37	15	108
손대	predicates	37	0	48
원해	predicates	37	34	53
치여	predicates	37	1	216
튀겨	predicates	37	1	6
날	other	37	1161	0
전	other	37	1072	97
후	other	37	292	0
놓다	function_words	37	0	376
말다	function_words	37	0	380

4. Assessing correlation between parental input frequency and age of acquisition

Preliminaries:

```
library("ggpubr") # for scatter plots
```

4.1. Organizing data

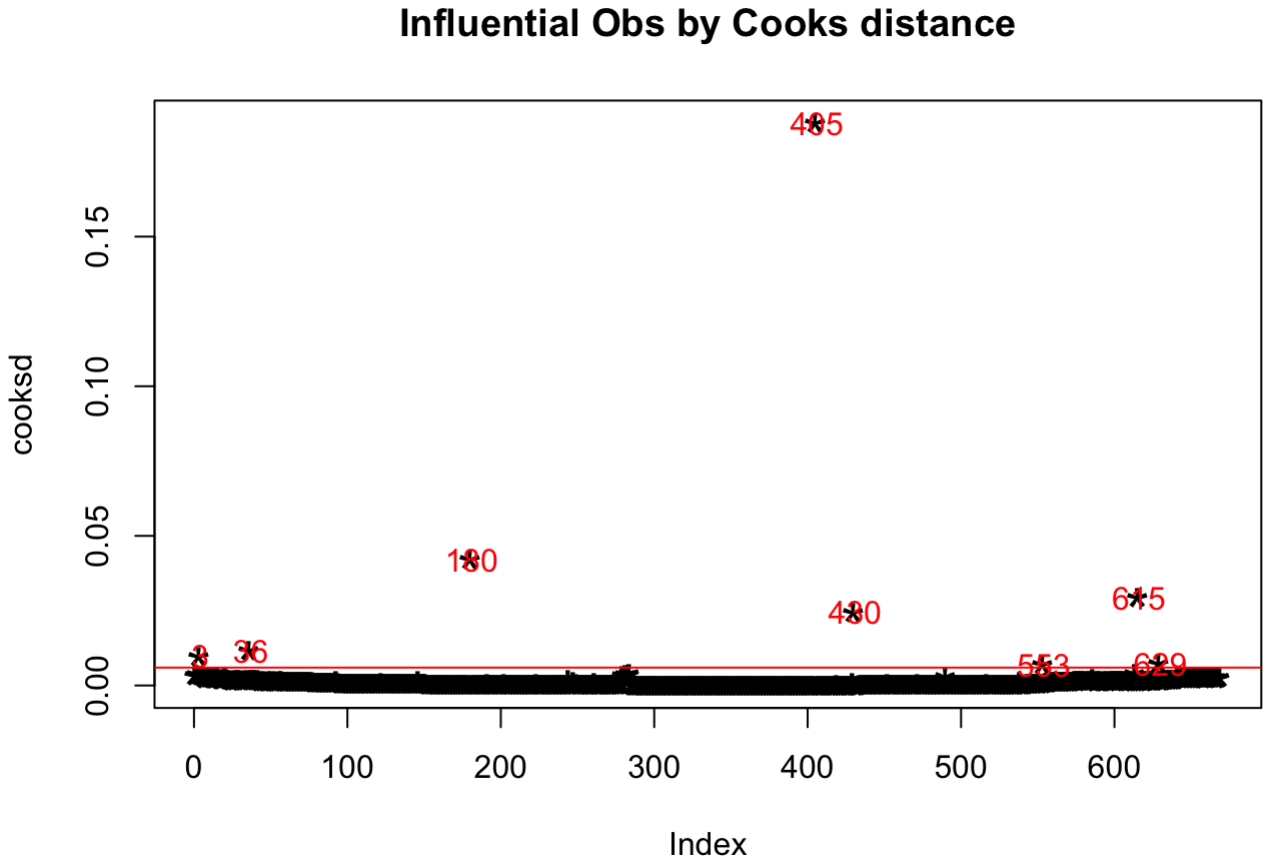
Change variable names:

4.2. Dealing with influential cases

Calculating cook's distance:

```
mod <- lm(aoa ~ freq_CHILDES, data = all_CDI)
cooksd <- cooks.distance(mod)

# Plot the Cook's Distance using the traditional 4/n criterion
sample_size <- nrow(all_CDI)
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4/sample_size, col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4/sample_size, names(cooksd),""), col="red")  # add lab
els
```



Inspecting influential cases:

```
# influential row numbers
influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])
all_CDI[influential, ]
```

```
## # A tibble: 8 × 5
##   definition lexical_category   aoa freq_CHILDES freq_adults
##   <chr>      <fct>            <dbl>        <int>       <dbl>
## 1 엄마       Non-object nouns     8         4853          98
## 2 응         Non-object nouns    12         7993           0
## 3 이         Object nouns        18        45114        1644
## 4 어         Function words      26        32857        1162
## 5 는         Non-object nouns    27        17664        1882
## 6 니         Function words      31         7301         264
## 7 에         Non-object nouns    34        10611           2
## 8 로         Non-object nouns    35         5075         352
```

Remove influential cases:

```
all_CDI <- all_CDI[-influential, ]
```

Make a table to be filled later:

```
results_table <- as.data.frame(table(all_CDI$lexical_category))
names(results_table)[names(results_table) == 'Var1'] <- 'category'
names(results_table)[names(results_table) == 'Freq'] <- 'n_words'
results_table$r_parental_input <- ""
results_table$r_adults_dataset <- ""
```

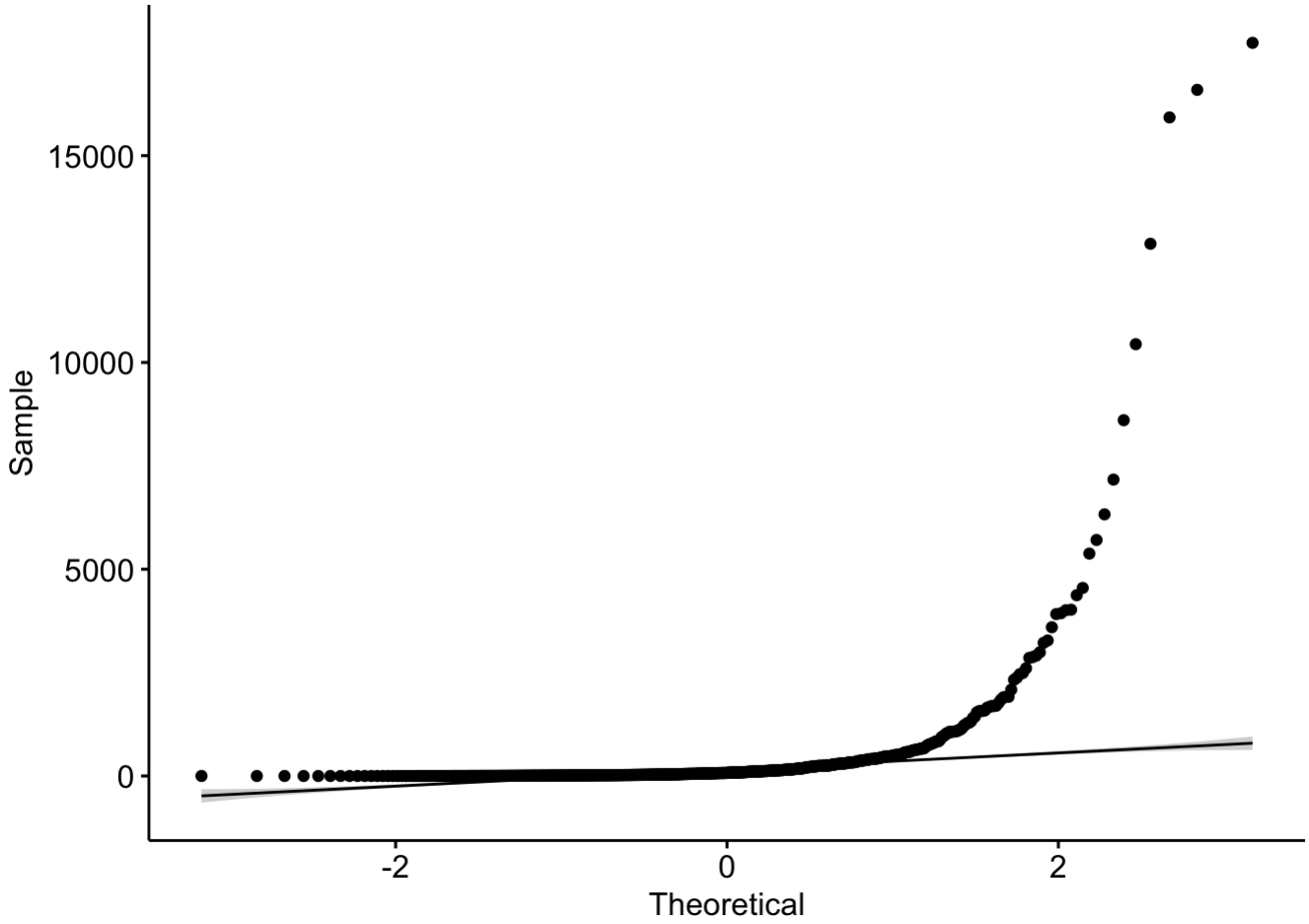
category	n_words	r_parental_input	r_adults_dataset
Object nouns	271		
Non-object nouns	122		
Predicates	202		
Function words	66		

4.3. Checking normality

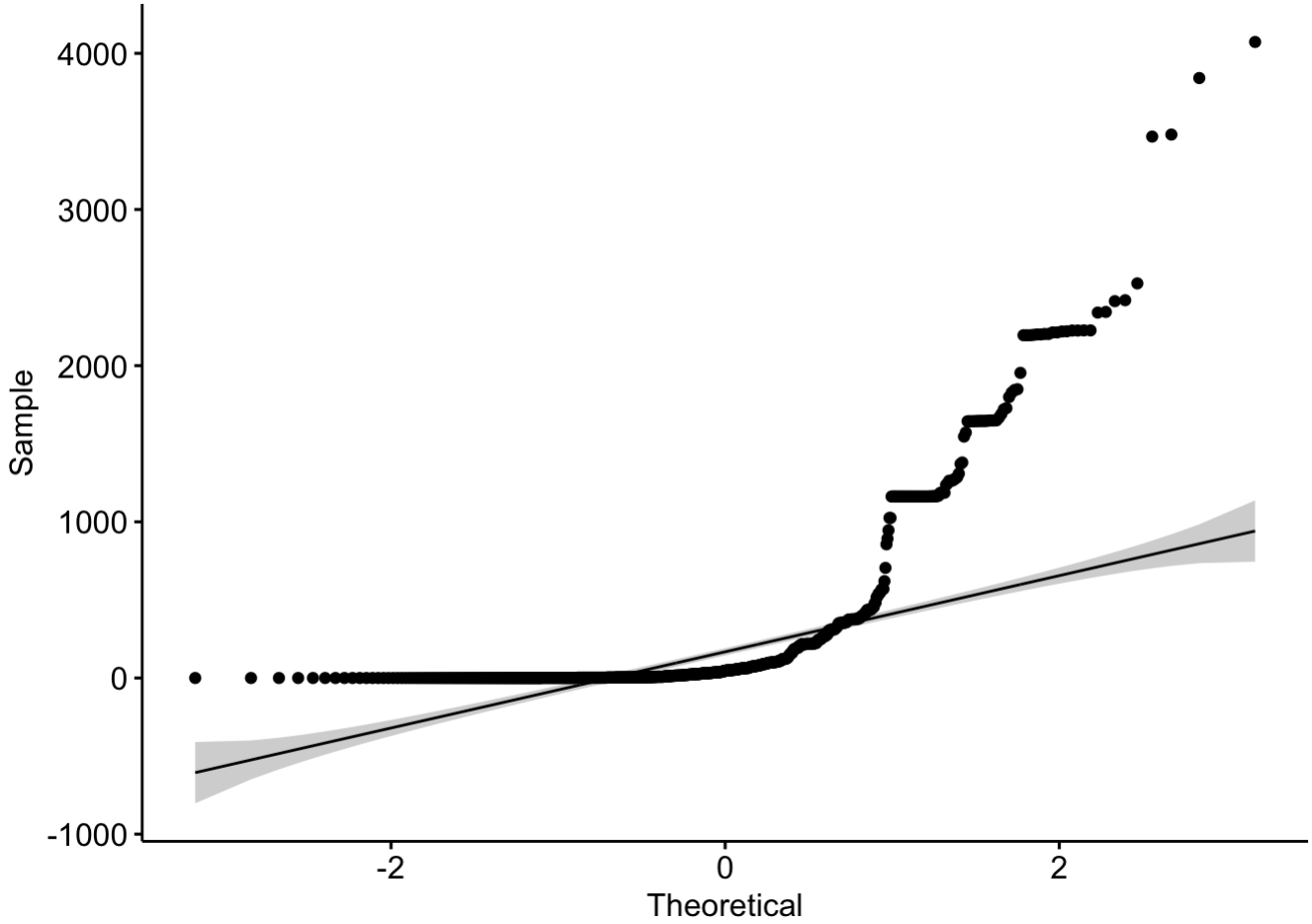
Visual inspection using Q-Q plots

Data normality for child-directed speech frequency data, adults frequency data, and age of word acquisition, respectively:

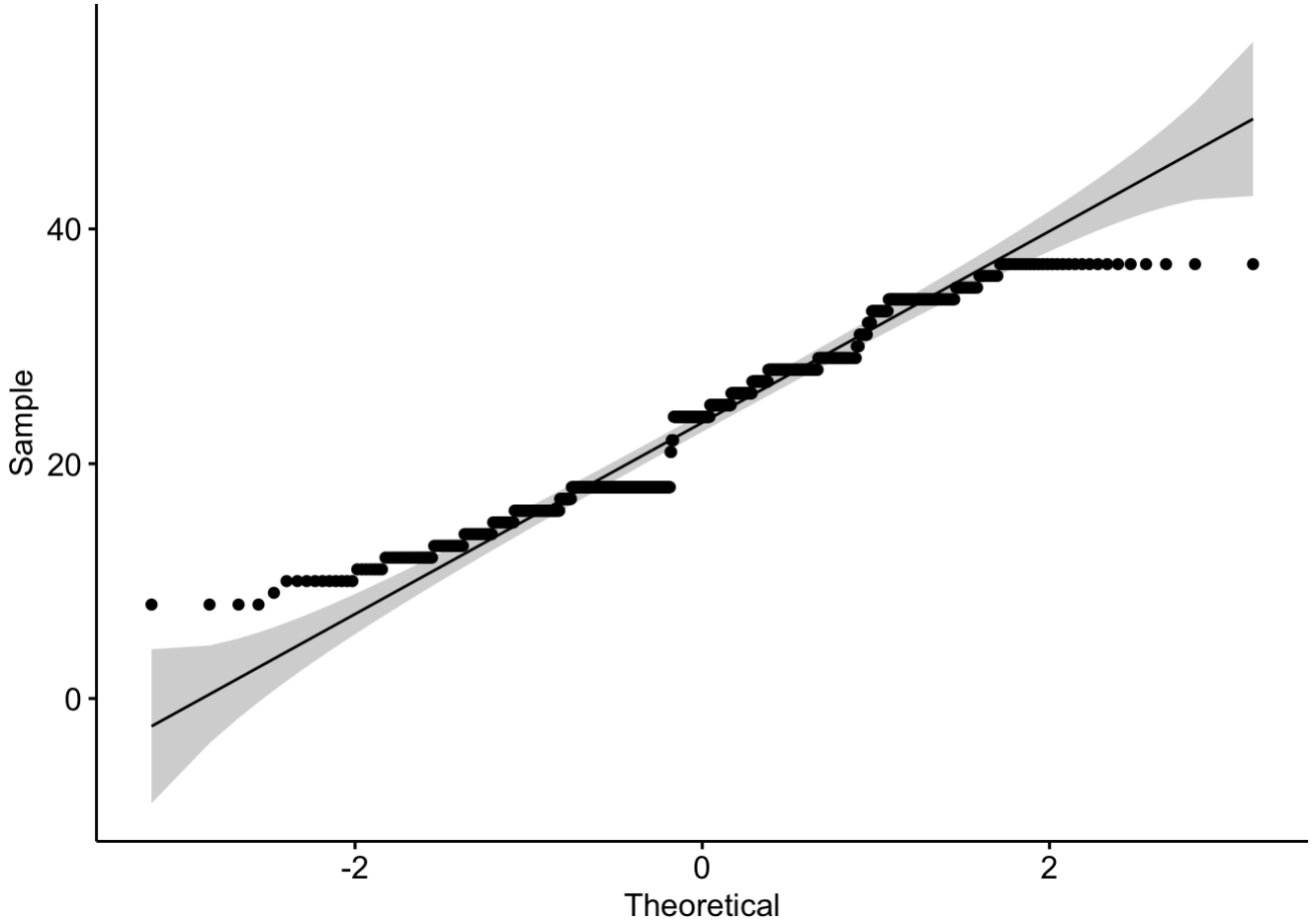
```
ggqqplot(all_CDI$freq_CHILDES)
```




```
ggqqplot(all_CDI$freq_adults)
```




```
ggqqplot(all_CDI$aoa)
```



-> The frequency data are not normally distributed. Therefore, Spearman rank correlation coefficient will be used for correlation analysis.

4.4. Spearman rank correlation coefficient

4.4.1. Parental input frequency and age of acquisition

4.4.1.1. All categories

Calculate Spearman rank correlation coefficient and append the results to the table:

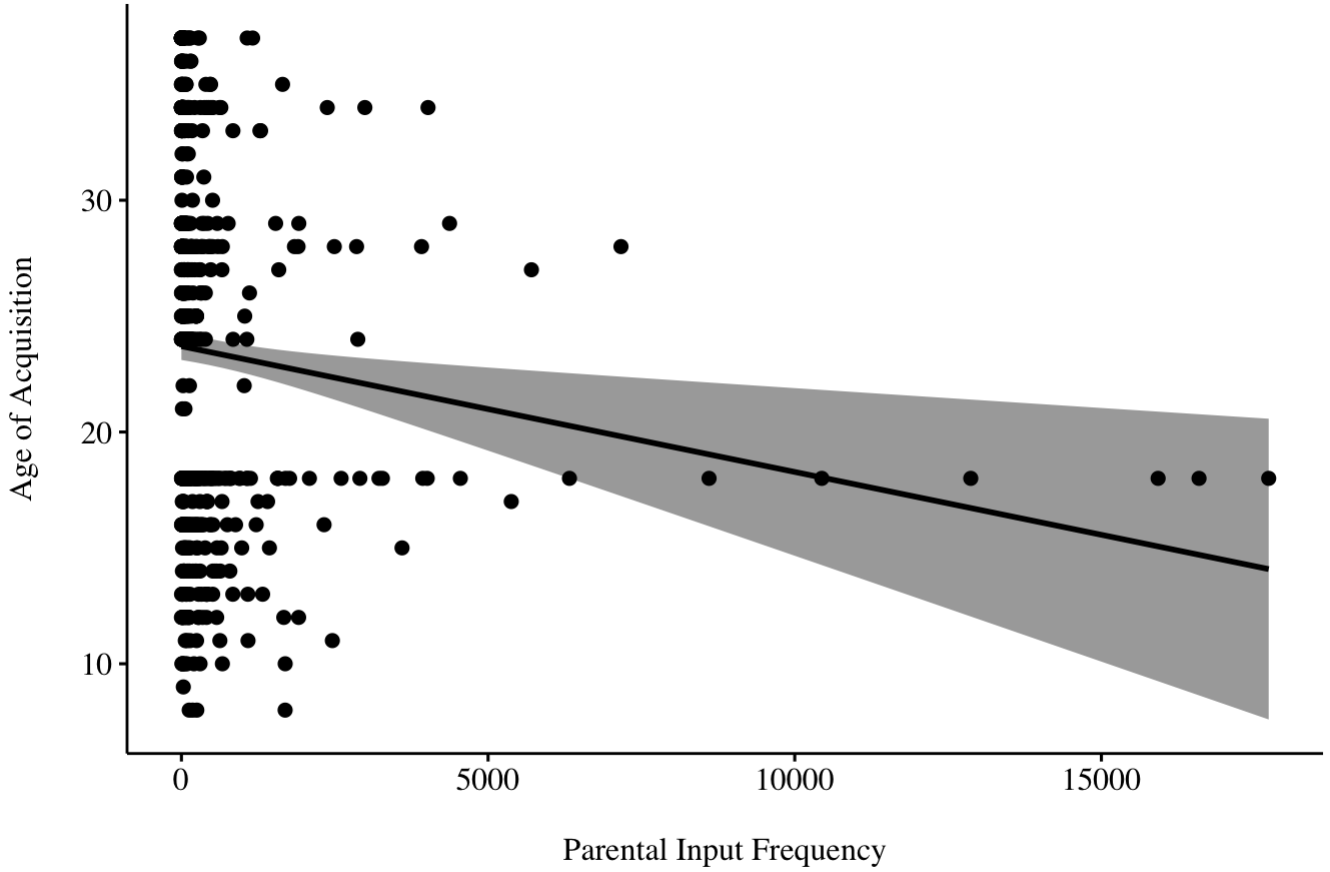
```
res_all_CHILDES <- cor.test(all_CDI$freq_CHILDES, all_CDI$aoa, method = "spearman", exact=FALSE)
res_all_CHILDES
```

```
##
##  Spearman's rank correlation rho
##
## data:  all_CDI$freq_CHILDES and all_CDI$aoa
## S = 63341588, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.3159422
```

```
results_table <- rbind(data.frame(category = "All categories", n_words = nrow(all_CDI),
                 r_parental_input = paste(res_all_CHILDES$estimate, "(", "p = ",
                                          res_all_CHILDES$p.value, ")"),
                 r_adults_dataset = ""), results_table)
```

Visual representation:

```
ggscatter(all_CDI, x = "freq_CHILDES", y = "aoa",
          add = "reg.line", conf.int = TRUE,
          cor.method = "spearman",
          xlab = paste("","Parental Input Frequency", sep = "\n"), ylab = paste("Age of Acquisition", "", sep =
"\n")) +
  labs(caption = paste("", "* Shaded region indicates confidence interval", sep = "\n")) +
  theme(text = element_text(family = "serif"))
```



* Shaded region indicates confidence interval

4.4.1.2. Function words

```
data_func <- all_CDI %>%
  filter(lexical_category == "Function words")
```

Calculate Spearman rank correlation coefficient and append the results to the table:

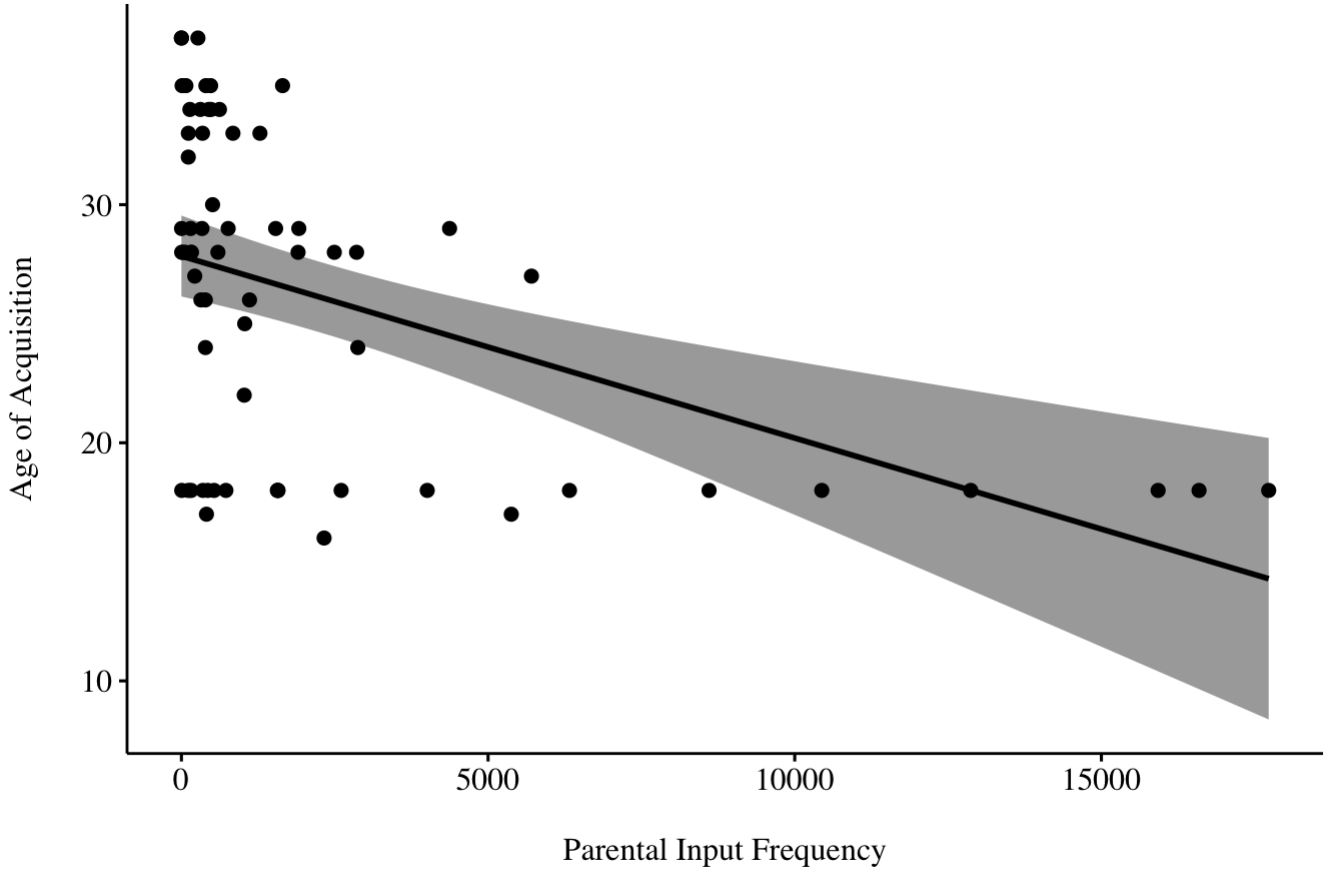
```
res_func_CHILDES <- cor.test(data_func$freq_CHILDES, data_func$aoa, method = "spearman", exact=FALSE)
res_func_CHILDES
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_func$freq_CHILDES and data_func$aoa
## S = 70334, p-value = 7.359e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.468201
```

```
results_table[5,3] = paste(res_func_CHILDES$estimate, "(", "p = ",
                           res_func_CHILDES$p.value, ")")
```

Visual representation:

```
ggscatter(data_func, x = "freq_CHILDES", y = "aoa",
          add = "reg.line", conf.int = TRUE,
          cor.method = "spearman",
          xlab = paste("","Parental Input Frequency", sep = "\n"), ylab = paste("Age of Acquisition", "", sep =
"\n")) +
  labs(caption = paste("", "* Shaded region indicates confidence interval", sep = "\n")) +
  theme(text = element_text(family = "serif"))
```



* Shaded region indicates confidence interval

4.4.1.3. Nouns

```
data_noun <- all_CDI %>%
  filter(lexical_category == "Object nouns")
```

Calculate Spearman rank correlation coefficient and append the results to the table:

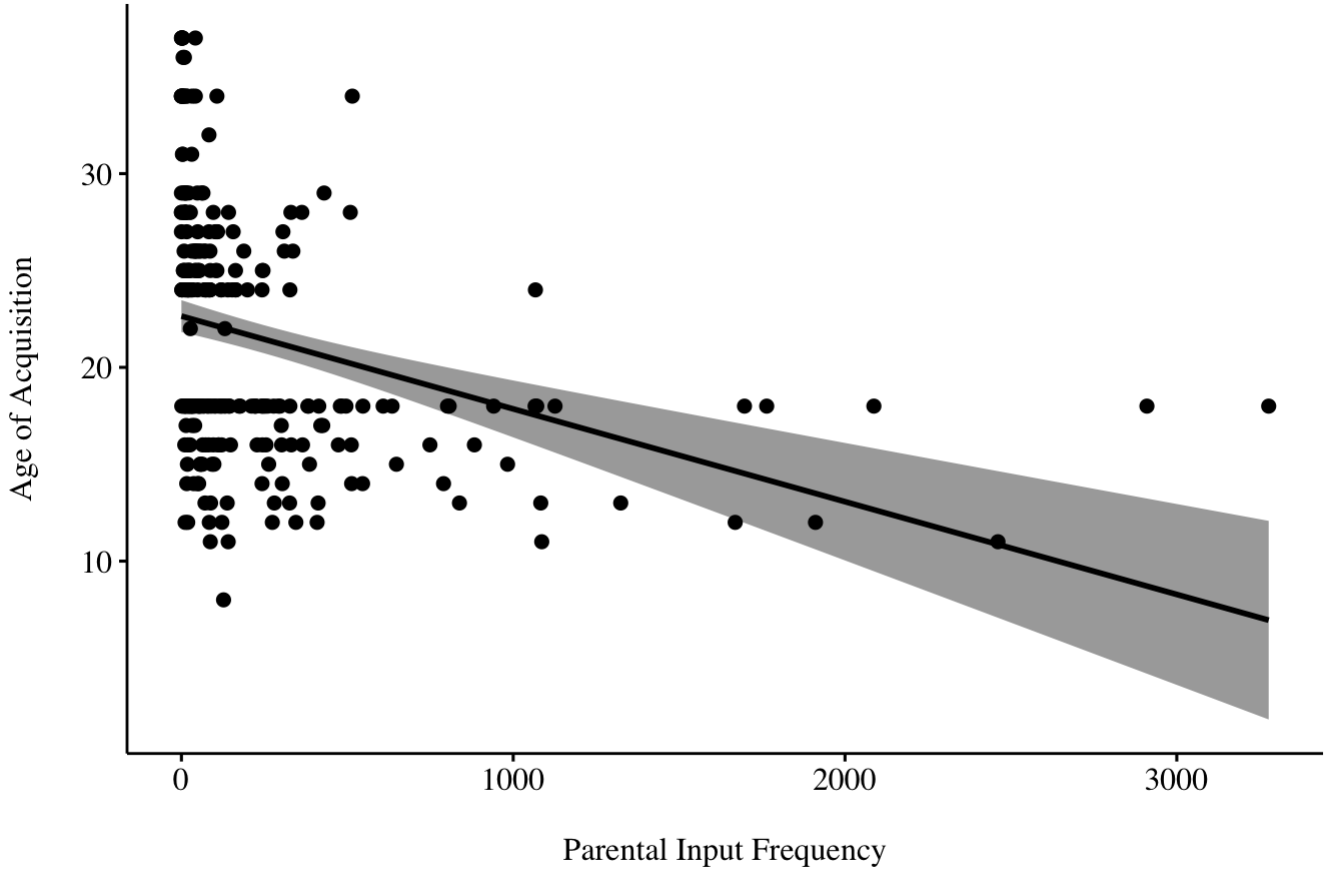
```
res_noun_CHILDES <- cor.test(data_noun$freq_CHILDES, data_noun$aoa, method = "spearman", exact=FALSE)
res_noun_CHILDES
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_noun$freq_CHILDES and data_noun$aoa
## S = 4950474, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4924371
```

```
results_table[2,3] = paste(res_noun_CHILDES$estimate, "(", "p = ",
                           res_noun_CHILDES$p.value, ")")
```

Visual representation:

```
ggscatter(data_noun, x = "freq_CHILDES", y = "aoa",
          add = "reg.line", conf.int = TRUE,
          cor.method = "spearman",
          xlab = paste("","Parental Input Frequency", sep = "\n"), ylab = paste("Age of Acquisition", "", sep =
"\n")) +
  labs(caption = paste("", "* Shaded region indicates confidence interval", sep = "\n")) +
  theme(text = element_text(family = "serif"))
```



* Shaded region indicates confidence interval

4.4.1.4. Predicates

```
data_pred <- all_CDI %>%
  filter(lexical_category == "Predicates")
```

Calculate Spearman rank correlation coefficient and append the results to the table:

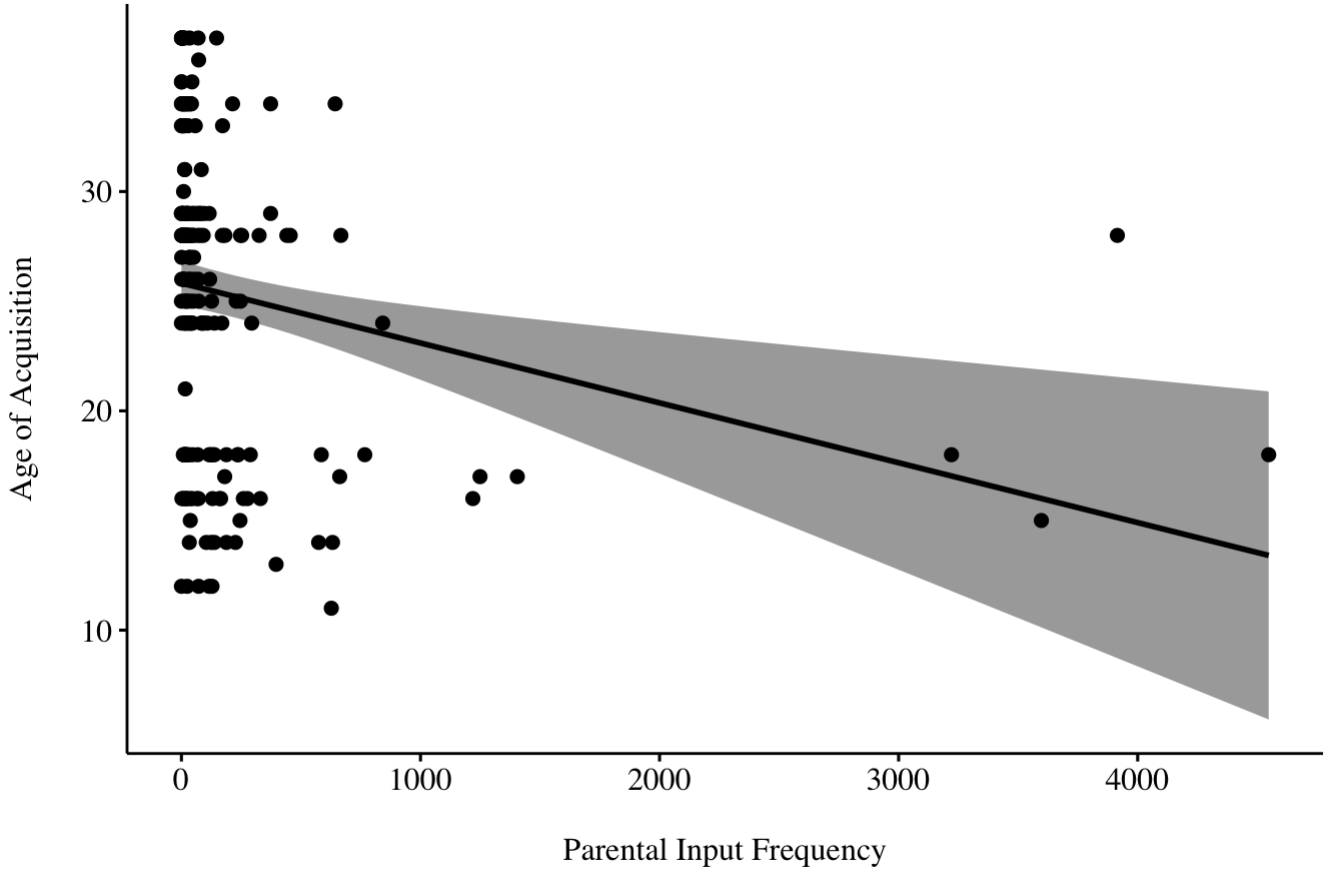
```
res_pred_CHILDES <- cor.test(data_pred$freq_CHILDES, data_pred$aoa, method = "spearman", exact=FALSE)
res_pred_CHILDES
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_pred$freq_CHILDES and data_pred$aoa
## S = 1940909, p-value = 1.017e-09
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4129049
```

```
results_table[4,3] = paste(res_pred_CHILDES$estimate, "(", "p = ",
                           res_pred_CHILDES$p.value, ")")
```

Visual representation:

```
ggscatter(data_pred, x = "freq_CHILDES", y = "aoa",
          add = "reg.line", conf.int = TRUE,
          cor.method = "spearman",
          xlab = paste("","Parental Input Frequency", sep = "\n"), ylab = paste("Age of Acquisition", "", sep =
"\n")) +
  labs(caption = paste("", "* Shaded region indicates confidence interval", sep = "\n")) +
  theme(text = element_text(family = "serif"))
```



* Shaded region indicates confidence interval

4.4.1.5. Non-object nouns

```
data_other <- all_CDI %>%
  filter(lexical_category == "Non-object nouns")
```

Calculate Spearman rank correlation coefficient and append the results to the table:

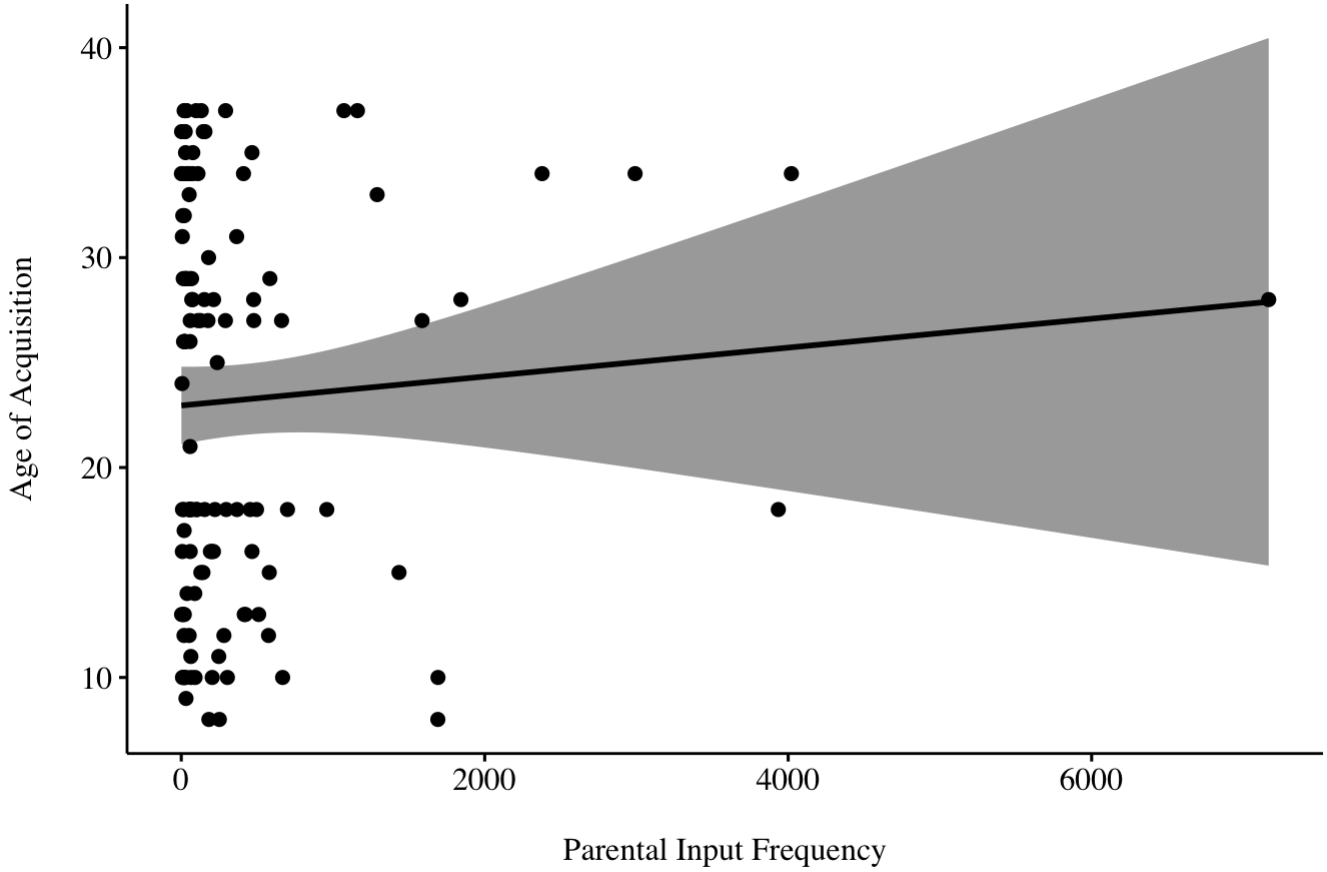
```
res_other_CHILDES <- cor.test(data_other$freq_CHILDES, data_other$aoa, method = "spearman", exact=FALSE)
res_other_CHILDES
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_other$freq_CHILDES and data_other$aoa
## S = 323972, p-value = 0.44
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##         rho
## -0.07055218
```

```
results_table[3,3] = paste(res_other_CHILDES$estimate, "(", "p = ",
                           res_other_CHILDES$p.value, ")")
```

Visual representation:

```
ggscatter(data_other, x = "freq_CHILDES", y = "aoa",
          add = "reg.line", conf.int = TRUE,
          cor.method = "spearman",
          xlab = paste("","Parental Input Frequency", sep = "\n"), ylab = paste("Age of Acquisition", "", sep =
"\n")) +
  labs(caption = paste("", "* Shaded region indicates confidence interval", sep = "\n")) +
  theme(text = element_text(family = "serif"))
```



* Shaded region indicates confidence interval

4.4.2. Adults language use dataset and age of acquisition

4.4.2.1. All categories

Calculate Spearman rank correlation coefficient and append the results to the table:

```
res_all_adults <- cor.test(all_CDI$freq_adults, all_CDI$aoa, method = "spearman", exact=FALSE)
res_all_adults
```

```
##
##  Spearman's rank correlation rho
##
## data:  all_CDI$freq_adults and all_CDI$aoa
## S = 46322104, p-value = 0.3339
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.03764314
```

```
results_table[1,4] = paste(res_all_adults$estimate, "(", "p = ",
                           res_all_adults$p.value, ")")
```

4.4.2.2. Function words

Calculate Spearman rank correlation coefficient and append the results to the table:

```
res_func_adults <- cor.test(data_func$freq_adults, data_func$aoa, method = "spearman", exact=FALSE)
res_func_adults
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_func$freq_adults and data_func$aoa
## S = 36823, p-value = 0.06164
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.2313292
```

```
results_table[5,4] = paste(res_func_adults$estimate, "(", "p = ",
                           res_func_adults$p.value, ")")
```

4.4.2.3. Nouns

Calculate Spearman rank correlation coefficient and append the results to the table:

```
res_noun_adults <- cor.test(data_noun$freq_adults, data_noun$aoa, method = "spearman", exact=FALSE)
res_noun_adults
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_noun$freq_adults and data_noun$aoa
## S = 3262679, p-value = 0.7883
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.01638832
```

```
results_table[2,4] = paste(res_noun_adults$estimate, "(", "p = ",
                           res_noun_adults$p.value, ")")
```

4.4.2.4. Predicates

Calculate Spearman rank correlation coefficient and append the results to the table:

```
res_pred_adults <- cor.test(data_pred$freq_adults, data_pred$aoa, method = "spearman", exact=FALSE)
res_pred_adults
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_pred$freq_adults and data_pred$aoa
## S = 1543584, p-value = 0.07952
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.1236682
```

```
results_table[4,4] = paste(res_pred_adults$estimate, "(", "p = ",
                           res_pred_adults$p.value, ")")
```

4.4.2.5. Non-object nouns

Calculate Spearman rank correlation coefficient and append the results to the table:

```
res_other_adults <- cor.test(data_other$freq_adults, data_other$aoa, method = "spearman", exact=FALSE)
res_other_adults
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_other$freq_adults and data_other$aoa
## S = 277491, p-value = 0.3632
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.08304109
```

```
results_table[3,4] = paste(res_other_adults$estimate, "(", "p = ",
                           res_other_adults$p.value, ")")
```

4.5. Results

```
results_table %>%
  kable(row.names = FALSE) %>%
  kable_styling()
```

category	n_words	r_parental_input	r_adults_dataset
All categories	661	-0.315942195196391 (p = 8.68154561692353e-17)	0.0376431442215373 (p = 0.333887603488449)
Object nouns	271	-0.492437113578119 (p = 5.84577923572889e-18)	0.0163883245638062 (p = 0.788270713611233)
Non-object nouns	122	-0.0705521810829852 (p = 0.439987095236933)	0.0830410870198309 (p = 0.363166293798427)
Predicates	202	-0.412904908687539 (p = 1.01688080966675e-09)	-0.123668246602311 (p = 0.0795186772357193)
Function words	66	-0.468200985151425 (p = 7.35901390403811e-05)	0.231329220393135 (p = 0.0616428056652297)

4. Inspecting the differences of age of acquisition and parental input frequency among categories

4.2. Age of acquisition

4.1.1. Kruskal-Wallis test

```
kruskal.test(aoa ~ lexical_category, data = all_CDI)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  aoa by lexical_category
## Kruskal-Wallis chi-squared = 42.499, df = 3, p-value = 3.145e-09
```

4.1.2. Pairwise Comparison

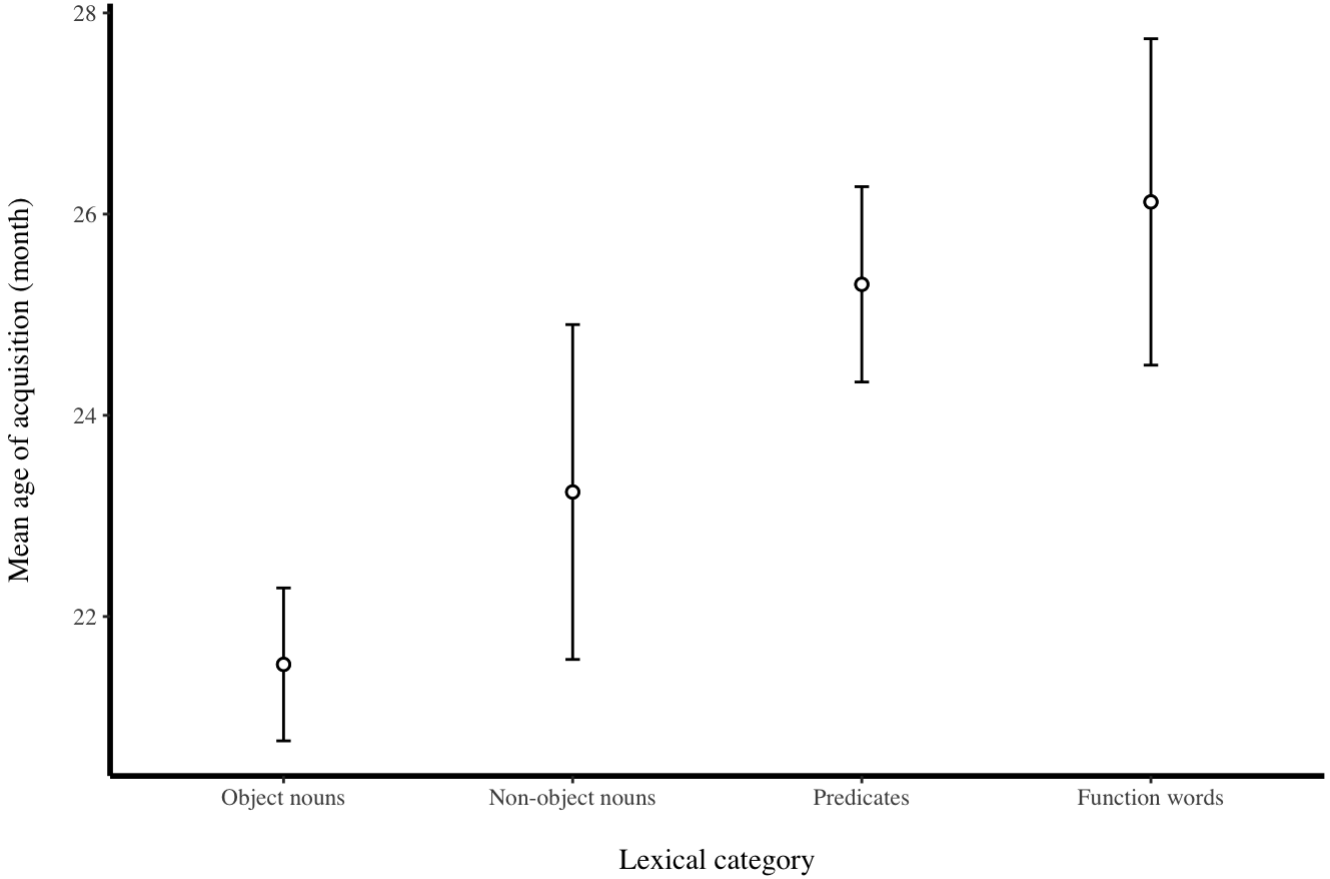
```
pairwise.wilcox.test(all_CDI$aoa, all_CDI$lexical_category,
                 p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  all_CDI$aoa and all_CDI$lexical_category
##
##                  Object nouns Non-object nouns Predicates
## Non-object nouns 0.107        -                -
## Predicates       2.7e-08      0.107            -
## Function words   7.5e-07      0.062            0.241
##
## P value adjustment method: BH
```

4.1.3. Visual representation

```
sum_aoa_cat <- all_CDI %>%
  group_by(lexical_category) %>%
  summarise(
    aoa_mean = mean(aoa),
    aoa_ci = 1.96 * sd(aoa)/sqrt(n()))

sum_aoa_cat %>%
  ggplot(aes(x = lexical_category, y = aoa_mean)) +
  geom_point() +
  geom_point(size = 2, color = "black") +
  geom_errorbar(aes(ymin = aoa_mean - aoa_ci, ymax = aoa_mean + aoa_ci),
                width = .05, linetype = 1) +
  geom_point(size = 2) +
  geom_point(size = 1, color = "white") +
  labs(x = paste("", 'Lexical category', sep = "\n"),
       y = paste('Mean age of acquisition (month)', "", sep = "\n"),
       caption = paste(" ", "Error bar: 95% CI", sep = "\n")) +
  theme(text=element_text(family="serif"),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(colour = "black", size = 1),
        axis.line.y = element_line(colour = "black", size = 1))
```



Error bar: 95% CI

4.2. Parental input frequency

4.2.1. Kruskal-Wallis test

```
kruskal.test(freq_CHILDES ~ lexical_category, data = all_CDI)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  freq_CHILDES by lexical_category
## Kruskal-Wallis chi-squared = 85.208, df = 3, p-value < 2.2e-16
```

4.2.2. Pairwise Comparison

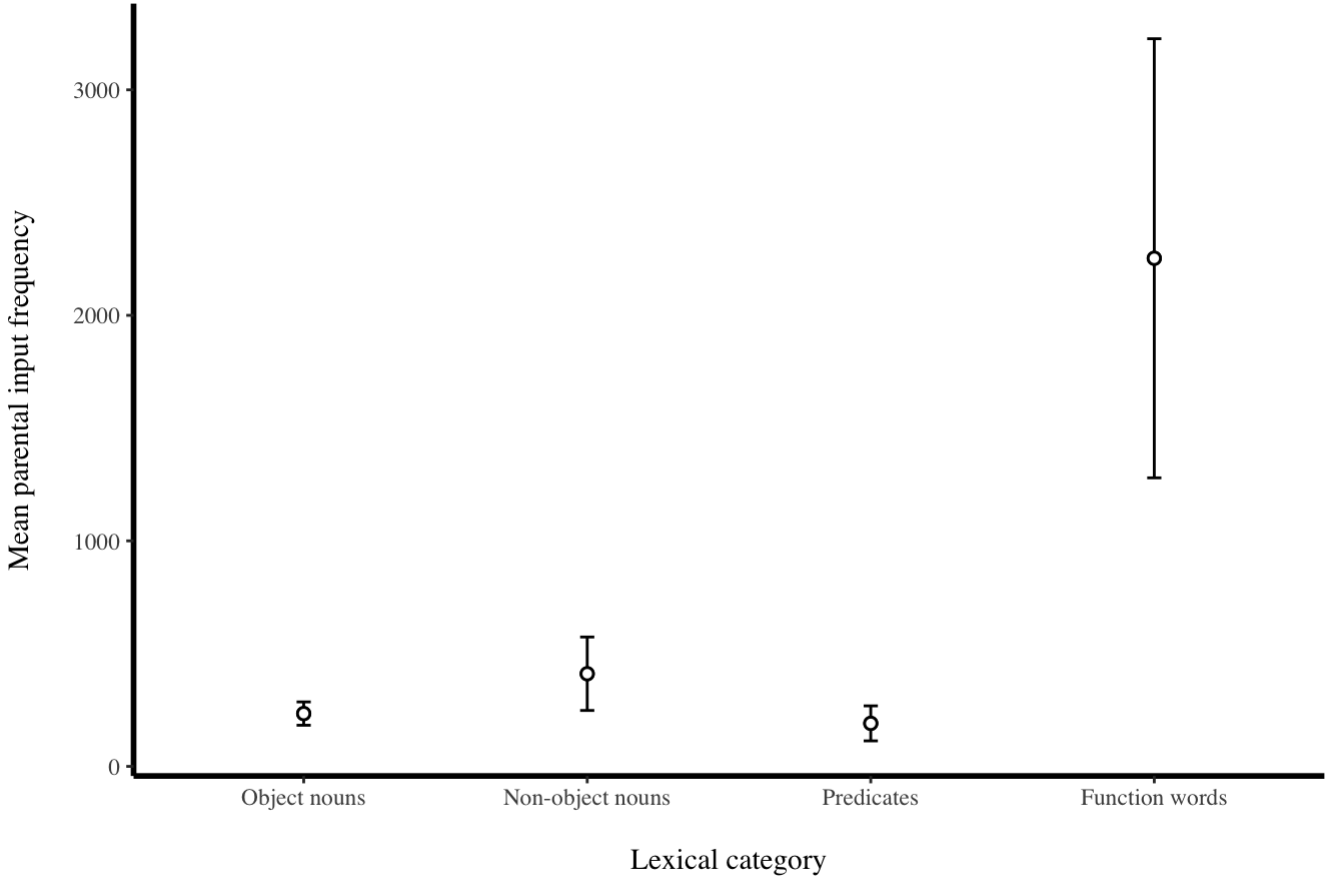
```
pairwise.wilcox.test(all_CDI$freq_CHILDES, all_CDI$lexical_category,
                 p.adjust.method = "BH")
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  all_CDI$freq_CHILDES and all_CDI$lexical_category
##
##                  Object nouns Non-object nouns Predicates
## Non-object nouns 0.06952      -                -
## Predicates       0.00015      3.3e-06          -
## Function words   1.1e-12      9.1e-08          3.0e-15
##
## P value adjustment method: BH
```

4.2.3. Visual representation

```
sum_freq_cat <- all_CDI %>%
  group_by(lexical_category) %>%
  summarise(
    freq_mean = mean(freq_CHILDES),
    freq_ci = 1.96 * sd(freq_CHILDES)/sqrt(n()))

sum_freq_cat %>%
  ggplot(aes(x = lexical_category, y = freq_mean)) +
  geom_point() +
  geom_point(size = 2, color = "black") +
  geom_errorbar(aes(ymin = freq_mean - freq_ci, ymax = freq_mean + freq_ci),
                width = .05, linetype = 1) +
  geom_point(size = 2) +
  geom_point(size = 1, color = "white") +
  labs(x = paste("", 'Lexical category', sep = "\n"),
       y = paste('Mean parental input frequency', "", sep = "\n"),
       caption = paste(" ", "Error bar: 95% CI", sep = "\n")) +
  theme(text=element_text(family="serif"),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(colour = "black", size = 1),
        axis.line.y = element_line(colour = "black", size = 1))
```



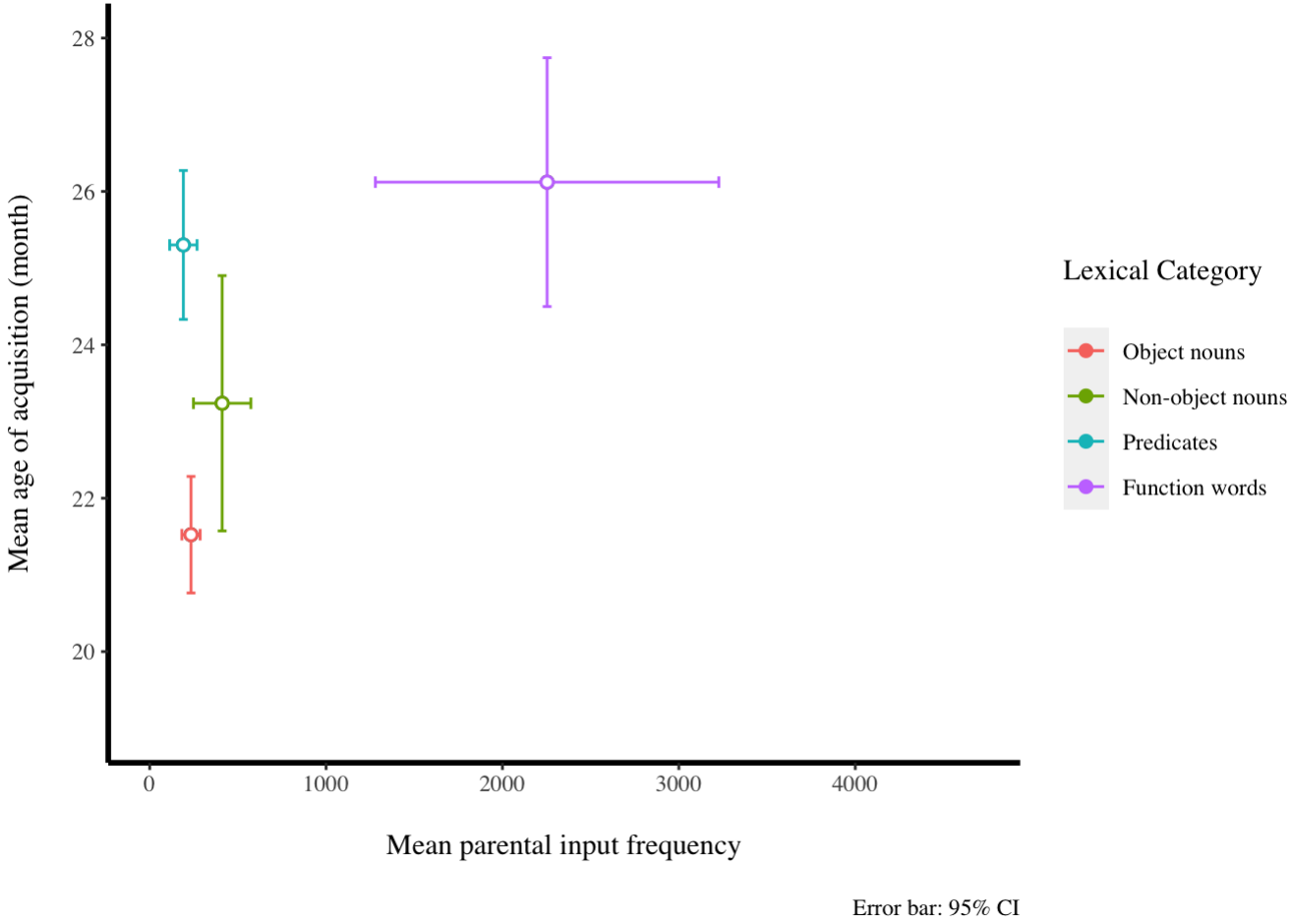
Error bar: 95% CI

4.3. Visual representation of mean age of acquisition as a function of input frequency and lexical categories

Mean age of acquisition as a function of mean parental frequency for different lexical categories:

```
data_summary <- all_CDI %>%
  group_by(lexical_category) %>%
  summarise(
    aoa_mean = mean(aoa),
    freq_mean = mean(freq_CHILDES),
    aoa_ci = 1.96 * sd(aoa)/sqrt(n()),
    freq_ci = 1.96 * sd(freq_CHILDES)/sqrt(n()))

data_summary %>%
  ggplot(aes(x = freq_mean, y = aoa_mean, color = lexical_category)) +
  geom_point() +
  xlim(0, 4700) + ylim(19, 28) +
  geom_point(size = 2, color = "black") +
  geom_errorbar(aes(ymin = aoa_mean - aoa_ci, ymax = aoa_mean + aoa_ci),
                width = 50, linetype = 1) +
  geom_errorbar(aes(xmin = freq_mean - freq_ci, xmax = freq_mean + freq_ci),
                width = .15, size = 0.5) +
  geom_point(size = 2) +
  geom_point(size = 1, color = "white") +
  labs(x = paste("", 'Mean parental input frequency', sep = "\n"),
       y = paste('Mean age of acquisition (month)', "", sep = "\n"),
       color = paste('Lexical Category', "", sep = "\n"),
       caption = paste(" ", "Error bar: 95% CI", sep = "\n")) +
  theme(text=element_text(family="serif"),
        panel.background = element_rect(fill = "white"),
        axis.line.x = element_line(colour = "black", size = 1),
        axis.line.y = element_line(colour = "black", size = 1))
```


Mean age of acquisition (month)
28
26
24
22
20
0
1000
2000
3000
4000
Mean parental input frequency
Lexical Category
Object nouns
Non-object nouns
Predicates
Function words
Error bar: 95% CI