Manipulation and Visualization of Autism Severity and Demographic Data

Gayane Hovsepyan

March 24, 2017

1 Introduction

The objective of this project was to see if there are any links between Autism Spectrum Disorder (ASD) severity in diagnosed children and parental ages and other demographic information. The disorder was examined from both a biological and sociological perspective.

According to the 5th edition of the American Psychiatric Association's Diagnostic and Statistical Manual (DSM-V), the criteria for ASD diagnoses are impairments in social communication and interaction as well as restricted, repetitive patterns of behavior. The datasets reflect this.

The datasets were in 2 parts: diagnostic scores of the children, and demographic information provided by parents. The data was manipulated and visualized in order to determine which factors correlated the most to autism severity in a diagnosed population. Linear regression analyses and plotting as well as histogram plotting were done with the datasets.

The data was manipulated using basic shell commands and Python, then visualized in R.

1.1 Data Source and Caveat

The data for this project was provided by the Early Childhood Partial Hospitalization Program (ECPHP) at the Resnick Psychiatric Institute at UCLA. ECPHP is a 10 week behavioral and psychiatric intervention program for preschool aged children who have autism and other developmental disorders which make it difficult for them to attend school with neurotypical children.

Because of the nature of the datasets, the statistics and figures presented in this report do not reflect a typical population. The datasets were comprised of a group of individuals who had presented enough symptoms to undergo a psychiatric evaluation, as well as be admitted into a program intended for children with somewhat severe ASD and other developmental disorders.

There is no identifiable patient information present in the datasets aside from ID numbers (this is only so that multiple datasets may be combined).

1.2 Data Collection

The data collection at ECPHP occurs both before and after preschool-aged children are admitted into the program.

1.2.1 Diagnostic Data

The children undergo a diagnostic test called the Autism Diagnostic Observation Schedule (ADOS) which is a semi-structured observational assessment for autism and more generally, ASD. The ADOS dataset includes the four subtotals of A, B, C and D, which are as follows:

A = communication total
B = communication & social interaction total
C = play total
D = stereotyped behaviors and restricted interests total

The higher the ADOS totals, the greater likelihood of ASD and the greater the severity of autism. The ADOS dataset also includes the date of administration of the test which was used as a proxy for the date of diagnosis.

1.2.2 Demographic Data

Once admitted into ECPHP, parents fill out several questionnaires, one of which is a demographic information sheet. The demographic information includes mother's and father's date of birth, child's date of birth, and family's socioeconomic status (measured via a proxy of mother's highest level of education (MHLE)).

1.3 Areas of Interest

Previous research has indicated that paternal age may be a significant factor in the development of autism (Sandin). Paternal age during birth of child was calculated and compared to all four ADOS totals. The same process was repeated with maternal age, as a great deal of literature also suggests a correlation between maternal age and a higher risk of autism (Sandin).

Higher socioeconomic status has been linked with an earlier age of diagnosis, possibly due to better resources and more contact with health professionals (Thomas). MHLE (mother's highest level of education) was used as a proxy for socioeconomic status. MHLE ranges from 1-8, with 8 being post-graduate education. This was compared with the age of diagnosis (date of the ADOS administration was used as a proxy for date of diagnosis).

The age of diagnosis was then compared with all four ADOS totals to determine if more severe cases are diagnosed earlier. This may be significant as research indicates that earlier diagnoses result in better prognoses (Fernell).

And finally, the age of diagnoses of males and females were compared, as it previous research has indicated that females are diagnosed at later ages because of an apparent lack of severity of symptoms (Kemp).

2 Methods

2.1 Cleaning Datasets in the Shell

The two files were joined together based on a common field (RID#). The join command was used in the shell. Executed code:

```
join --nocheck-order -t"," -a 1 -a 2 -e "NULL" -o "0,1.1,1.2,1.3,
1.4,1.5,1.6,1.7,1.8,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,2.10,2.11,
2.12,2.13,2.14" ADOS.csv demo.csv > ADOSandDemo.csv
```

- join merges two files based on single column
- -nocheck-order prevents need for sorting on only one column (files were already sorted)
- -t "," sets delimiter as comma
- -a 1 and -a 2 indicate that if no value present in field of first and second file, still print
- -e says to replace empty value with "NULL"
- -o says to replace empty value with "NULL" if it is present in 0 (key), 1.1 (first field of first file), 2.2 (second field of second file) etc.
- first two file names simply reference the files
- > writes to new file

Afterwards, fields which were not used for data analysis were deleted for the sake of simplicity. Executed code:

```
cut -f 2,3,5,10,13,14,17,18,21,22 -d "," --complement
ADOSandDemo.csv > ADOSandDemo2.csv
#complement does the inverse selection
```

2.2 Manipulating Data in Python

Using the father's date of birth coupled with the child's date of birth, paternal age at the time of birth was calculated as follows:

```
#need datetime to deal with ages
from datetime import datetime
#define function
```

```
#read file but do not write
def ageoffather(filename): #defining function
    fobj = open(filename, "r") #opening file
    data = fobj.readlines() #reading file
    #close file
    fobj.close()
    #open output file and write everything in last file to it + new column with ages
    fobj = open("output1.csv", "w") #opening and writing to new file with additional column of father's age
    fobj.write(data[0].strip()+ ",Father_birth_age\n") #column name
    for line in data[1:]: #skipping header
        item = line.split(",") #delimiter is comma
        fatherDOB = item [11] #specifiying which column is father's DOB
        childDOB = item [6] #specifying which column is child's DOB
        if item[11] != "NULL" and item[6] != "NULL": # so that code ignores NULL items
            F = datetime.strptime(fatherDOB, "%Y/%m/%d") #read date
            C = datetime.strptime(childDOB, "%Y/%m/%d") #read date
            delta = C-F #difference in date
            age=delta.days/365.0 #dividing by 365 to get age in years
            fobj.write(line.strip() + "," + str(age)  + "\n")
            print(age)
        else:
            fobj.write(line.strip() + "," + "NULL" + "\n") # if 6 and 11 are NULL, write line and add null in place of AOF
            print("NULL")
    fobj.close()
#decided not to round since I will be plotting data and would prefer accuracy
```

The code essentially reads through the input file, closes it, opens a new output file, and writes each lines to the new file with the addition of a new column containing the father's age at the time of birth (calculated by subtracting the father's date of birth from the child's date of birth.)

The same process was repeated with the mother's age when she gave birth to the child. This time, the output file from the father's age function was used as the input file for the mother's age function. The code is as follows:

```
#need datetime to deal with ages
```

```
from datetime import datetime
#define function
#read file but do not write
def ageofmother(filename): #defining function
    fobj = open(filename, "r") #opening file
    data = fobj.readlines() #reading file
    #close file
    fobj.close()
    #open output file and write everything in last file to it + new column with ages
    fobj = open("output2.csv", "w") #opening and writing to new file with additional column of mother's age
    fobj.write(data[0].strip()+ ",Mother_birth_age\n") #column name
    for line in data[1:]: #skipping header
        item = line.split(",") #delimiter is comma
        motherDOB = item [9] #specifiying which column is father's DOB
        childDOB = item [6] #specifying which column is child's DOB
        if item[9] != "NULL" and item[6] != "NULL": # so that code ignores NULL items
            M = datetime.strptime(motherDOB, "%Y/%m/%d") #read date
            C = datetime.strptime(childDOB, "%Y/%m/%") #read date
            delta = C-M #difference in date
            age=delta.days/365.0 #dividing by 365 to get age in years
            fobj.write(line.strip() + "," + str(age)  + "\n")
            print(age)
        else:
            fobj.write(line.strip() + "," + "NULL" + "\n") # if 6 and 9 are NULL, write line and add null in place of AOM
            print("NULL")
    fobj.close()
#decided not to round since I will be plotting data and would prefer accuracy
#will use ouput of ageoffather function as argument for age of mother function
```

Following this, it was necessary to also have a column in the datasets for the age of the child during the time of diagnosis. The date of the ADOS administration was used as a proxy for the date of the diagnosis. The child's date of birth was subtracted from the date of the diagnosis to determine the age

at the time of diagnosis. Similar to the age of parent functions, this function reads through the input file (in this case, the output file from the age of mother function), opens a new output file and rewrites each line, but with the addition of a column for the age of diagnosis. The code is as follows:

```
#need datetime to deal with ages
from datetime import datetime
#define function
#read file but do not write
def diagnosisage(filename): #defining function
    fobj = open(filename, "r") #opening file
    data = fobj.readlines() #reading file
    #close file
    fobj.close()
    #open output file and write everything in last file to it + new column with ages
    fobj = open("output3.csv", "w") #opening and writing to new file with additional column of age of diagnosis
    fobj.write(data[0].strip()+ ",Diagnosis_age\n") #column name
    for line in data[1:]: #skipping header
        item = line.split(",") #delimiter is comma
        ADOSdate = item [1] #specifiying which column is father's DOB
        childDOB = item [6] #specifying which column is child's DOB
        if item[1] != "NULL" and item[6] != "NULL": # so that code ignores NULL items
            A = datetime.strptime(ADOSdate, "%Y/%m/%d") #read date
            C = datetime.strptime(childDOB, "%Y/%m/%d") #read date
            delta = A-C #difference in date
            age=delta.days/365.0 #dividing by 365 to get age in years
            fobj.write(line.strip() + "," + str(age)  + "\n")
            print(age)
        else:
            fobj.write(line.strip() + "," + "NULL" + "\n") # if 6 and 11 are NULL, write line and add null in place of AOF
            print("NULL")
    fobj.close()
#decided not to round since I will be plotting data and would prefer accuracy
#using output of mother's age function as argument for diagnosis age
```

The file "output3.csv" was the final one used for data visualization in R.

It was later necessary to separate the values above and below the median of all four ADOS subtotal in order to visualize the data in R as overlain histograms. This was somewhat difficult as the nature of the dataset did not allow use of bulit-in median functions in Python or R. The following was a manually built method of determining the median:

```
def findmedian(filename):
    #ADOS Total A Median
    fobj = open(filename, "r") #opening file
    data = fobj.readlines()[1:] #reading file
    listA = [] #empty list
    for line in data:
        item = line.split(",") #defining items as everything in
        ↪  between commas
        if item [2] != "NULL": #skip null elements
            item2 = int(item[2]) #making values in item 2
            ↪  integers so that 1, 10, 2, 11 are not --> 1, 10,
            ↪  11, 2
            listA.append(item2) #add numbers to list
    listA.sort() #sort that in numerical order
    midA = int(len(listA)//2) #takes length of list, divides by 2
    ↪  to get middle, // == integer division
    print("Median of ADOS A total is " + str(listA[midA]))

    #ADOS Total B Median
    fobj2 = open(filename, "r") #opening file
    data3 = fobj2.readlines()[1:] #reading file
    listB = [] #empty list
    for line in data3:
        item = line.split(",") #defining items as everything in
        ↪  between commas
        if item [3] != "NULL": #skip null elements
            item3 = int(item[3]) #making values in item 3
            ↪  integers so that 1, 10, 2, 11 are not --> 1, 10,
            ↪  11, 2
            listB.append(item3) #add numbers to list
    listB.sort() #sort that in numerical order
    midB = len(listB)//2 #takes length of list, divides by 2 to
    ↪  get middle, // == integer division
    print("Median of ADOS B total is " + str(listB[midB]))

    #ADOS Total C Median
    fobj3 = open(filename, "r") #opening file
    data3 = fobj3.readlines()[1:] #reading file
    listC = [] #empty list
```

```
    for line in data3:
        item = line.split(",") #defining items as everything in between commas
        if item [4] != "NULL": #skip null elements
            item4 = int(item[4]) #making values in item 4 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listC.append(item4) #add numbers to list
    listC.sort() #sort that in numerical order
    midC = len(listC)//2 #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS C total is " + str(listC[midC]))

    #ADOS Total D median
    fobj4 = open(filename, "r") #opening file
    data4 = fobj4.readlines()[1:] #reading file
    listD = [] #empt list
    for line in data4:
        item = line.split(",") #defining items as everything in between commas
        if item [5] != "NULL": #skip null elements
            item5 = int(item[5]) #making values in item 5 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listD.append(item5) #add numbers to list
    listD.sort() #sort that in numerical order
    midD = len(listD)//2 #takes length of list, divides by 2 to get middle index, // == int division
    print("Median of ADOS D total is " + str(listD[midD]))
findmedian("output3.csv")
```

Median of ADOS A total is 5
Median of ADOS B total is 8
Median of ADOS C total is 3
Median of ADOS D total is 2

This function essentially works by taking the values of the columns each ADOS total, inserting them into a list, sorting the list, determining the length of the list (number of items in the list), dividing the length by 2 to get the middle index, then calling the value of that index in the list.

Though this function worked, it was later determined that when making the histograms using the parental ages and ADOS scores, many of the scores had null values for the parental ages. This meant that the median values for the scores which *also* had corresponding parental ages were different from the median values of the scores irrespective of the parental ages.

New functions were written which took this into account. The first function

calculated the median values of all four ADOS totals *if* the scores also had matching paternal age. This was done by skipping the lines which had null values for the father's age, as seen below:

```
def findmedianF(filename):
    #ADOS Total A Median
    fobj = open(filename, "r") #opening file
    data = fobj.readlines()[1:] #reading file
    listA = [] #empty list
    for line in data:
        item = line.split(",") #defining items as everything in between commas
        if item [2] != "NULL" and item[13]!= "NULL": #skip null elements, and skip if father's age is null as well
            item2 = int(item[2]) #making values in item 2 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listA.append(item2) #add numbers to list
    listA.sort() #sort that in numerical order
    midA = int(len(listA)//2) #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS A total is " + str(listA[midA]))

    #ADOS Total B Median
    fobj2 = open(filename, "r") #opening file
    data3 = fobj2.readlines()[1:] #reading file
    listB = [] #empty list
    for line in data3:
        item = line.split(",") #defining items as everything in between commas
        if item [3] != "NULL" and item[13]!="NULL": #skip null elements, and skip if father's age is null as well
            item3 = int(item[3]) #making values in item 3 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listB.append(item3) #add numbers to list
    listB.sort() #sort that in numerical order
    midB = len(listB)//2 #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS B total is " + str(listB[midB]))

    #ADOS Total C Median
    fobj3 = open(filename, "r") #opening file
    data3 = fobj3.readlines()[1:] #reading file
    listC = [] #empty list
    for line in data3:
```

```
        item = line.split(",") #defining items as everything in between commas
        if item [4] != "NULL" and item[13]!= "NULL": #skip null elements, and skip if father's age is null as well
            item4 = int(item[4]) #making values in item 4 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listC.append(item4) #add numbers to list
    listC.sort() #sort that in numerical order
    midC = len(listC)//2 #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS C total is " + str(listC[midC]))

    #ADOS Total D median
    fobj4 = open(filename, "r") #opening file
    data4 = fobj4.readlines()[1:] #reading file
    listD = [] #empt list
    for line in data4:
        item = line.split(",") #defining items as everything in between commas
        if item [5] != "NULL" and item[13]!="NULL": #skip null elements, and skip if father's age is null as well
            item5 = int(item[5]) #making values in item 5 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listD.append(item5) #add numbers to list
    listD.sort() #sort that in numerical order
    midD = len(listD)//2 #takes length of list, divides by 2 to get middle index, // == int division
    print("Median of ADOS D total is " + str(listD[midD]))
findmedianF("output3.csv")
```

Median of ADOS A total is 5
Median of ADOS B total is 9
Median of ADOS C total is 3
Median of ADOS D total is 3

A similar function was written to only calculate the median values of all four ADOS totals *if* the scores also had matching maternal age, as seen below:

```
def findmedianM(filename):
    #ADOS Total A Median
    fobj = open(filename, "r") #opening file
    data = fobj.readlines()[1:] #reading file
    listA = [] #empty list
    for line in data:
```

```
        item = line.split(",") #defining items as everything in between commas
        if item [2] != "NULL" and item[14]!= "NULL": #skip null elements, and skip if mother's age is null as well
            item2 = int(item[2]) #making values in item 2 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listA.append(item2) #add numbers to list
    listA.sort() #sort that in numerical order
    midA = int(len(listA)//2) #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS A total is " + str(listA[midA]))

    #ADOS Total B Median
    fobj2 = open(filename, "r") #opening file
    data3 = fobj2.readlines()[1:] #reading file
    listB = [] #empty list
    for line in data3:
        item = line.split(",") #defining items as everything in between commas
        if item [3] != "NULL" and item[14]!="NULL": #skip null elements, and skip if mother's age is null as well
            item3 = int(item[3]) #making values in item 3 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listB.append(item3) #add numbers to list
    listB.sort() #sort that in numerical order
    midB = len(listB)//2 #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS B total is " + str(listB[midB]))

    #ADOS Total C Median
    fobj3 = open(filename, "r") #opening file
    data3 = fobj3.readlines()[1:] #reading file
    listC = [] #empty list
    for line in data3:
        item = line.split(",") #defining items as everything in between commas
        if item [4] != "NULL" and item[14]!= "NULL": #skip null elements, and skip if mother's age is null as well
            item4 = int(item[4]) #making values in item 4 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listC.append(item4) #add numbers to list
    listC.sort() #sort that in numerical order
```

```
    midC = len(listC)//2 #takes length of list, divides by 2 to get middle, // == integer division
    print("Median of ADOS C total is " + str(listC[midC]))

    #ADOS Total D median
    fobj4 = open(filename, "r") #opening file
    data4 = fobj4.readlines()[1:] #reading file
    listD = [] #empt list
    for line in data4:
        item = line.split(",") #defining items as everything in between commas
        if item [5] != "NULL" and item[14]!="NULL": #skip null elements, and skip if mother's age is null as well
            item5 = int(item[5]) #making values in item 5 integers so that 1, 10, 2, 11 are not --> 1, 10, 11, 2
            listD.append(item5) #add numbers to list
    listD.sort() #sort that in numerical order
    midD = len(listD)//2 #takes length of list, divides by 2 to get middle index, // == int division
    print("Median of ADOS D total is " + str(listD[midD]))
findmedianM("output3.csv")
```

Median of ADOS A total is 5
Median of ADOS B total is 9
Median of ADOS C total is 3
Median of ADOS D total is 3

2.3 Data Visualization in R

Linear regression tests and plots were run on the variables of paternal age, maternal age, MHLE, and age of diagnosis against all four ADOS totals. The following was the code used for a linear regression analysis of father's age vs. ADOS A total:

```
mydata <- read.csv("/home/eeb177-student/Desktop/eeb-177-final-project/output3.csv", na.strings = "NULL")
lm (mydata$ATotal ~ mydata$Father_birth_age, mydata = 'output3.csv') #linear regression of Age of father field vs ADOS total A field)
lm.out = lm (mydata$ATotal ~ mydata$Father_birth_age, mydata= "output3.csv", na.strings="NULL") #naming output
summary(lm.out) #gives LOTS of info about output, including coefficients, R values, R^2 etc
options(show.signif.stars=F) #turning off significance stars
```

```
anova(lm.out) #gives ANOVA table
plot(mydata$ATotal ~ mydata$Father_birth_age,
  ↪  mydata='output3.csv', ylab = "ADOS Total A", xlab= "Age of
  ↪  Father at Birth", main="Age of Father at Birth vs. ADOS Total
  ↪  A")#plotting linear regression and naming plot
abline(lm.out, col="red") #linear regression line red
```

This was repeated with the other three ADOS totals, as well as maternal age and all four ADOS totals, MHLE and age of diagnosis, as well as age of diagnosis and all four ADOS totals. The following figures were obtained:

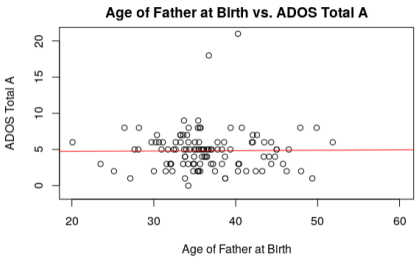

Figure 1: Linear regression analysis of paternal age during birth of child vs. ADOS total A.

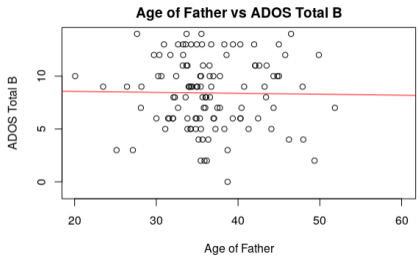

Figure 2: Linear regression analysis of paternal age during birth of child vs. ADOS total B.

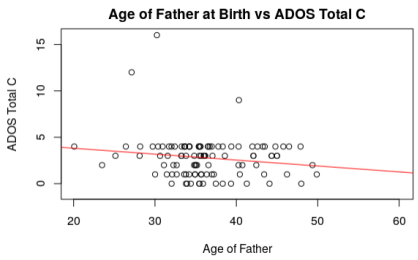

Figure 3: Linear regression analysis of paternal age during birth of child vs. ADOS total C.

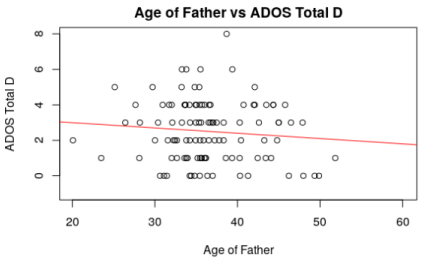

Figure 4: Linear regression analysis of paternal age during birth of child vs. ADOS total D.

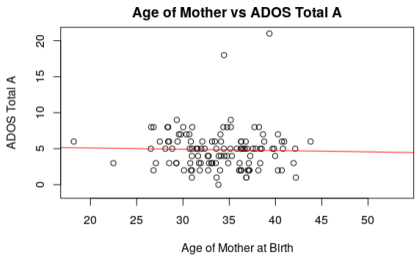

Figure 5: Linear regression analysis of maternal age during birth of child vs. ADOS total A.

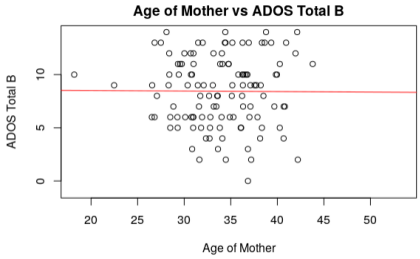

Figure 6: Linear regression analysis of maternal age during birth of child vs. ADOS total B.

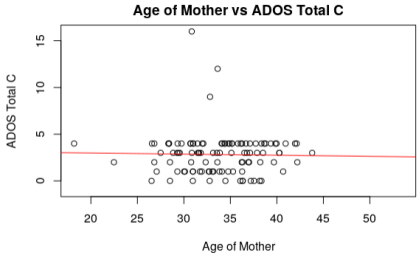

Figure 7: Linear regression analysis of maternal age during birth of child vs. ADOS total C.

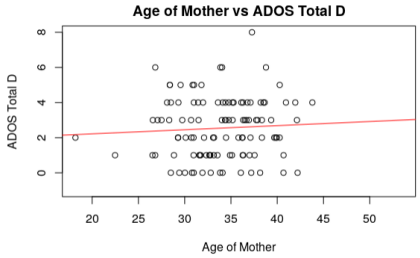

Figure 8: Linear regression analysis of maternal age during birth of child vs. ADOS total D.

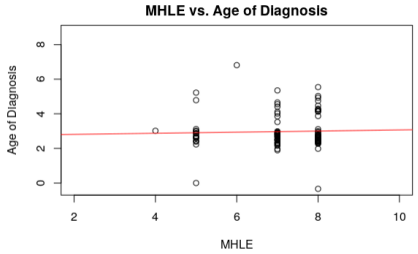

Figure 9: Linear regression analysis of MHLE vs Age od diagnosis

None of the above figures demonstrated any strong correlation or significance, however, three of the four age of diagnosis vs ADOS total regressions resulted in significant correlations:

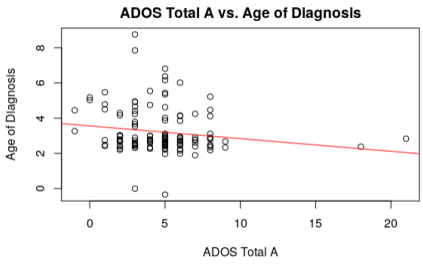

Figure 10: Linear regression analysis of age of diagnosis vs. ADOS total A.

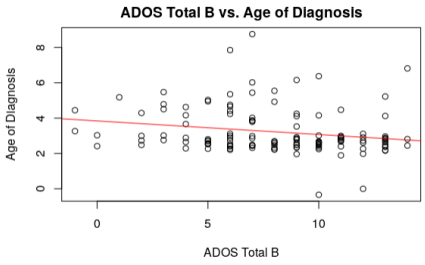

Figure 11: Linear regression analysis of age of diagnosis vs. ADOS total B.

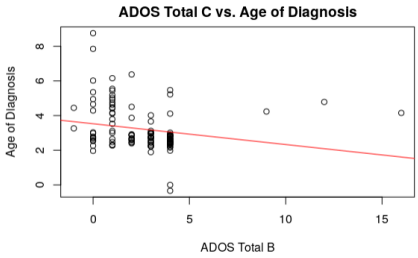

Figure 12: Linear regression analysis of age of diagnosis vs. ADOS total C.

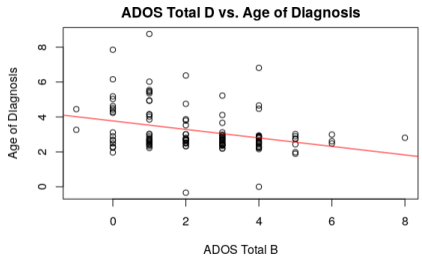

Figure 13: Linear regression analysis of age of diagnosis vs. ADOS total D.

Of figures 10-13, figures 11, 12, and 13 demonstrated significant correlation ($p < 0.05$), with R^2 values of 0.04292, 0.04225, 0.1027 respectively.

The following table is a summary of the linear regression statistics. The last three rows were the only ones with $p < 0.05$.

Statistics				
variables Tested	R^2	Adjusted R^2	Type of Correlation	p-value
Age of Father vs ADOS Total A	0.0001151	-0.008733	positive	0.9094
Age of Father vs ADOS Total B	0.0002288	-0.008698	negative	0.8731
Age of Father vs ADOS Total C	0.0252	0.01574	negative	0.1058
Age of Father vs ADOS Total D	0.009568	0.0002241	negative	0.3139
Age of Mother vs ADOS Total A	0.0007846	-0.00798	negative	0.7653
Age of Mother vs ADOS Total B	4.308E-05	-0.008806	negative	0.9445
Age of Mother vs ADOS Total C	0.0005609	-0.009049	negative	0.8096
Age of Mother vs ADOS Total D	0.003713	-0.005598	positive	0.5291
ADOS Total A vs Age of Diagnosis	0.02479	0.01705	negative	0.07594
ADOS Total B vs Age of Diagnosis	0.04292	0.03526	negative	0.01944
ADOS Total C vs Age of Diagnosis	0.04225	0.034	negative	0.02554
ADOS Total D vs Age of Diagnosis	0.1027	0.09512	negative	0.0003399

After the linear regressions, multiple histograms were made in R. First, the the ADOS scores of males and females were compared. The following code was executed:

```
newdata <-
 ↪  read.csv("/home/eeb177-student/Desktop/eeb-177-final-project/output3.csv",
 ↪  na.strings = "NULL")
hist((newdata$ATotal)[newdata$Gender == 1], freq = FALSE,
 ↪  col=rgb(1,0,0,0.5), ylim = c(0,0.3), main="Gender and ADOS A
 ↪  total", xlab="ADOS A Total") #male ados scores
hist(newdata$ATotal[newdata$Gender == 2], freq = FALSE,
 ↪  col=rgb(0,0,1,0.5), add=T) #female ados scores
box()
legend(x = "topright", # location of legend within plot area
 c("Male", "Female"), #labels
 col = c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)), #colors
```

```
8    lwd = c(10, 10)) #size of lines in legend
```

This was repeated with totals B, C, and D. The following histograms were the output:

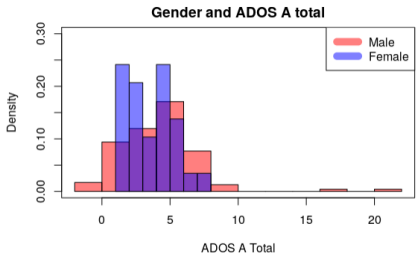

Figure 14: Histogram of ADOS A scores for males and females.

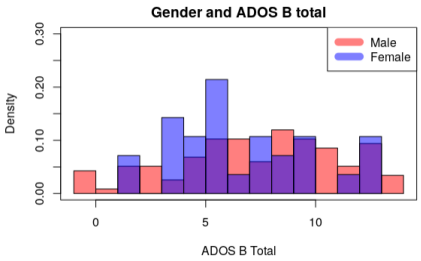

Figure 15: Histogram of ADOS B scores for males and females.

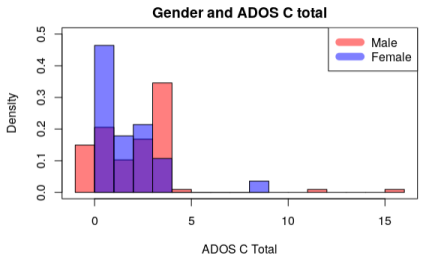

Figure 16: Histogram of ADOS C scores for males and females.

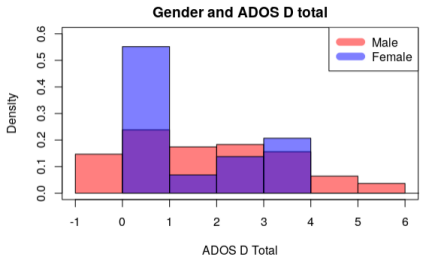

Figure 17: Histogram of ADOS D scores for males and females.

The following R code was executed in order to produce Figure 18, which displays age of diagnosis instead of ADOS scores:

```
newdata <-
 ↪  read.csv("/home/eeb177-student/Desktop/eeb-177-final-project/output3.csv",
 ↪  na.strings = "NULL")
hist((newdata$Diagnosis_age)[newdata$Gender == 1], freq = FALSE,
 ↪  breaks = 30, col=rgb(1,0,0,0.5), ylim = c(0,1.3),
 ↪  main="Gender and Age of Diagnosis", xlab="Age of
 ↪  Diagnosis")#male scores
hist(newdata$Diagnosis_age[newdata$Gender == 2], freq = FALSE,
 ↪  breaks = 30, col=rgb(0,0,1,0.5), add=T)#female scores
legend(x = "topright", # location of legend within plot area
 c("Male", "Female"),#labels
 col = c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)),#colors
 lwd = c(10, 10))#thickness of lines in legend
box()
```

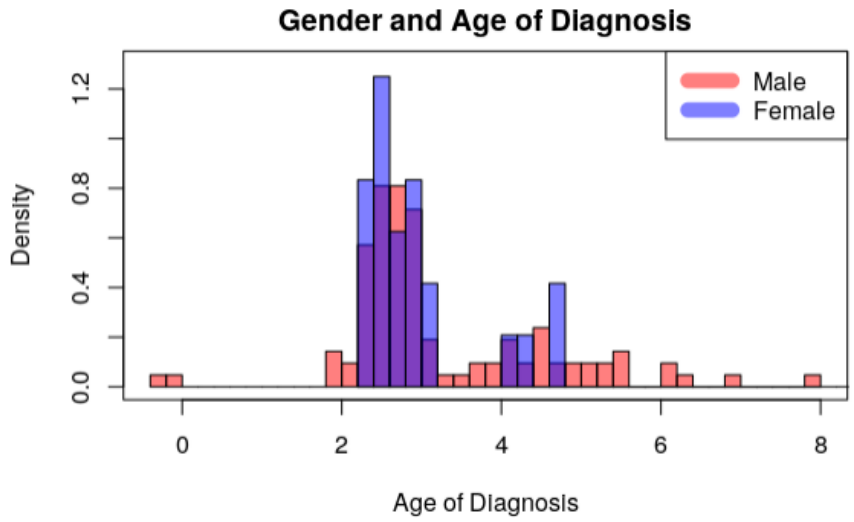

Figure 18: Age of diagnosis of males vs. females.

Following this, histograms were created which displayed parental ages in terms of corresponding ADOS scores (below and above the median). The median values were determined by the Python functions "findmedianF" and "findmedianM" detailed earlier. The following is the code that was used to obtain a histogram of the paternal ages of scores less than and greater than or equal to the median:

```r
hist(newdata$Father_birth_age[newdata$ATotal >= 5], freq = FALSE, col=rgb(1,0,0,0.5), breaks = 25, ylim = c(0,0.25), main="Father's Age and ADOS A Totals", xlab="Father's age during birth of child") #scores above median
hist(newdata$Father_birth_age[newdata$ATotal < 5], breaks = 30, freq = FALSE, col=rgb(0,0,1,0.5), add=T) #scores below median
box()
legend(x = "topright", # location of legend within plot area
 c("Greater than or equal to median score", "Less than median score"), #labels
 col = c(rgb(1,0,0,0.5),rgb(0,0,1,0.5)),#colors
 lwd = c(10, 10)) #thickness of lines in legend
```

This was repeated with totals B, C, and D, then again with maternal age and all four totals. The following figures were obtained:

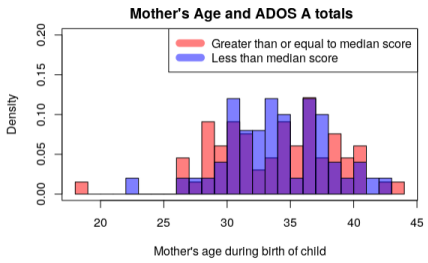

Figure 19: Histogram of mother's age in ADOS A scores below and above the median.

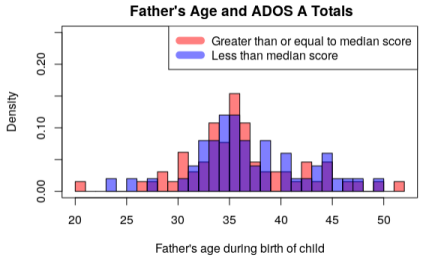

Figure 20: Histogram of father's age in ADOS A scores below and above the median.

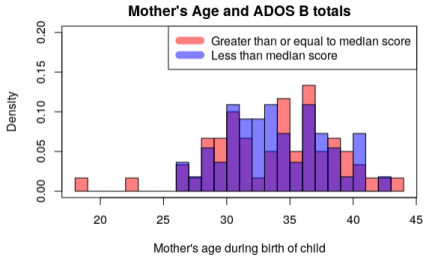

Figure 21: Histogram of mother's age in ADOS B scores below and above the median.

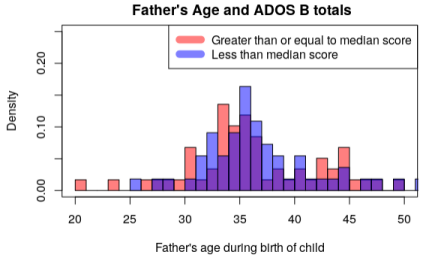

Figure 22: Histogram of father's age in ADOS B scores below and above the median.

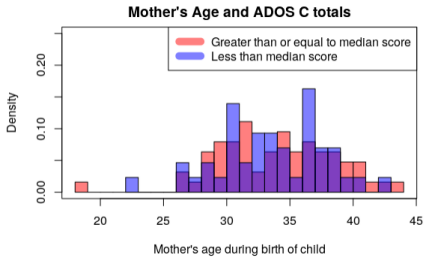

Figure 23: Histogram of mother's age in ADOS C scores below and above the median.

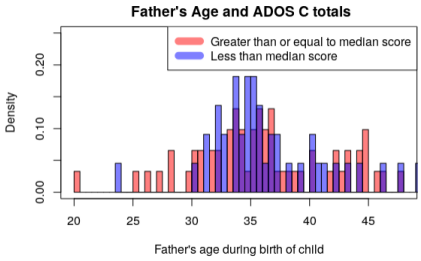

Figure 24: HHistogram of father's age in ADOS C scores below and above the median.

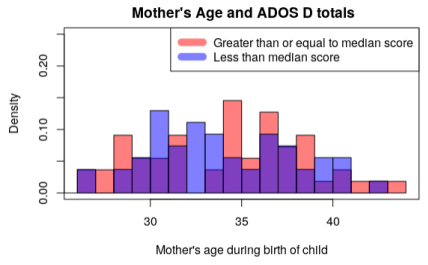

Figure 25: Histogram of mother's age in ADOS D scores below and above the median.

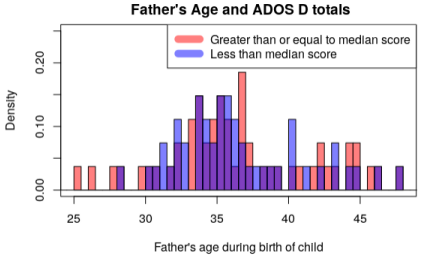

Figure 26: Histogram of father's age in ADOS D scores below and above the median.

3 Discussion

Of figures 10-13, figures 11, 12, and 13 demonstrated significant correlation ($p < 0.05$), with R^2 values of 0.04292, 0.04225, 0.1027 respectively. This trend was expected as more severe cases of autism tend to be diagnosed earlier. Figure 10, which was not significant, showed the lowest correlation, possibly because total A reflects communication abilities, the lack of which is hard to detect at an early age (Fernell).

As it is apparent from Figures 14-17, females had scores concentrated in the mid-ranges, while male scores were more normally distributed. The female sample size was smaller, so it could have affected these figures. Females are thought to exhibit milder symptoms of ASD, but this could simply be due to a test-bias for male symptoms (Kemp).

Figure 18 depicts the age of diagnosis of males and females. While there does not appear to be much of a difference in the distribution, an interesting aspect is that there is a spike in diagnoses at ages 4-5 after a steep drop from age 3 when diagnoses occur. This may be significant because 4-5 is the age

when children enter school for the first time, and are exposed to teachers and professionals who may be aware of the signs and symptoms of ASD. Prior to this, the spike around age 3 could be because of frequent visits to the doctor and constant parental care. The lack of diagnoses occurring in the window between age 3 and age 4 could be due to parents returning to work, infrequent visits doctor visits, and lack of contact with other health and education professionals.

It is apparent from figures 21-26, the paternal and maternal ages at the time of birth have different affects on ADOS scores. The maternal ages in Figures 21, 23, and 25 are somewhat normally distributed, whereas the paternal ages are not. There is a spike in the frequency of ADOS scores above the median when the father's age is around 40-45 years. This is apparent in Figures 22, 24, and 26.

4 References

Fernell, E., Eriksson, M. A., Gillberg, C. (2013). Early diagnosis of autism and impact on prognosis: a narrative review. *Clinical Epidemiology*, 5, 33–43. http://doi.org/10.2147/CLEP.S41714

Kemp, C. (2015, April 28). Age at autism diagnosis differs between boys, girls. Retrieved from http://www.aappublications.org/content/early/2015/04/28/aapnews.20150428-3?sso=1sso$_r$$edirect_c$$ount$ = $1nfstatus$ = $401nftoken$ = 00000000 − 0000 − 0000 − 0000 − 000000000000$nfstatusdescription$ = $ERROR$

Sandin, S., Schendel, D., Magnusson, P., Hultman, C., SurÃ©n, P., Susser, E., . . . Reichenberg, A. (2015). Autism risk associated with parental age and with increasing difference in age between the parents. *Molecular Psychiatry*, 21(5), 693-700. doi:10.1038/mp.2015.70

Thomas, P., Zahorodny, W., Peng, B., Kim, S., Jani, N., Halperin, W., Brimacombe, M. (2012). The association of autism diagnosis with socioeconomic status. *Autism*, 16(2), 201-213. doi:10.1177/1362361311413397