# Final Project

Ellie Bi, Justin Nguyen, Terrie Kim

2023-03-14

```
library('ggplot2')
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.4
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

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

## Problem 1

*a)*

```
Cases <- as.data.frame(read.csv("C:/Users/ellie/OneDrive/Documents/cases.csv"))
Children <- as.data.frame(read.csv("C:/Users/ellie/OneDrive/Documents/children.csv"))
Parents <- as.data.frame(read.csv("C:/Users/ellie/OneDrive/Documents/parents.csv"))
Payments <- as.data.frame(read.csv("C:/Users/ellie/OneDrive/Documents/payments.csv"))
```

```
dim(Cases)
```

```
## [1] 172422      6
```

```
dim(Children)
```

```
## [1] 257253      9
```

```
dim(Parents)
```

```
## [1] 128317     10
```

```
dim(Payments)
```

```
## [1] 1510216       6
```

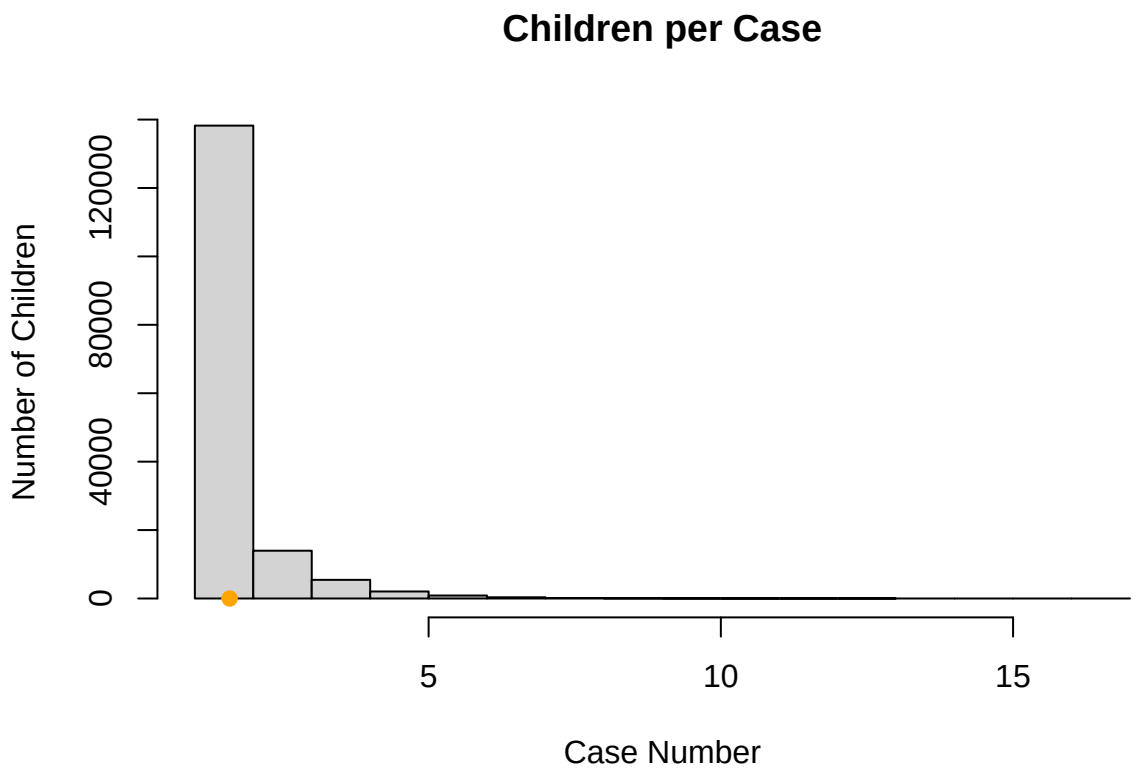
*b)*

```
hist(table(Children$CASE_NUM), main = 'Children per Case', xlab = 'Case Number', ylab = 'Number of Chil
```

```
case_distrb <- mean(table(Children$CASE_NUM)); case_distrb
```

```
## [1] 1.595991
```

```
points(x = case_distrb, y = 0, col = 'orange', pch = 19)
```




The distribution is skewed right with the location of average number being around 1.595991.

*c)*

```
max(table(Children$ID))
```

```
## [1] 12
```

The most number of cases per child is 12. It is the same child because each child has a unique ID, and since the table() function returns a contingency table, the max() function will find the largest number of cases associated with one child.

*d)*

```
pool.pay.par <- Payments %>%
  left_join(Parents, by = 'AP_ID')
sum(is.na(pool.pay.par$AP_ID))
```

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

Every absent parent does have an identifying record because there are no missing rows when the payments and parents data frames are joined.

# Problem 2

```
pool_categories <- function(var, threshold){
  if ("Other" %in% levels(var)){  # checks if Other already exists
    stop("'Other' Already Exists")
  }
  tab <- table(var)
  cat_list <- vector(mode = "list")
  count = 0
  for (i in tab){
    count = count + 1
    if (i < threshold){
      cat_list <- append(cat_list, tab[count])
    }
  }

  change <- names(cat_list)

  for (i in change) {
    var[var == i] <- 'Other'
  }
  return(var)
}
```

```
# testing the function
table(pool_categories(Payments$PYMNT_SRC, 8))
```

```
##
##       A       C       F       G       I       L   Other       S       U       W
##   69144    2092    6690     513   19762     120       9    4305   50574 1356858
##       X       Z
##      70      79
```

```
table(pool_categories(Payments$PYMNT_SRC, 150))
```

```
##
##       A       C       F       G       I   Other       S       U       W
##   69144    2092    6690     513   19762     278    4305   50574 1356858
```

```
table(pool_categories(Payments$PYMNT_SRC, 20000))
```

```
##
##       A   Other       U       W
##   69144   33640   50574 1356858
```

# Problem 3

*a)*

```
# changes the dates to the correct format
Payments$DATE <- as.Date(Payments$COLLECTION_DT, format =  "%m/%d/%Y"); head(Payments)
```

```
##    CASE_NUM PYMNT_AMT     COLLECTION_DT PYMNT_SRC PYMNT_TYPE   AP_ID       DATE
## 1 871449518     80.77 9/10/2015 0:00:00         W          E 1784827 2015-09-10
## 2 871449518     15.00 9/10/2015 0:00:00         W          E 1784827 2015-09-10
## 3 871449518     80.77 9/17/2015 0:00:00         W          E 1784827 2015-09-17
## 4 871449518     15.00 9/17/2015 0:00:00         W          E 1784827 2015-09-17
## 5 871449518     80.77 9/24/2015 0:00:00         W          E 1784827 2015-09-24
## 6 871449518     15.00 9/24/2015 0:00:00         W          E 1784827 2015-09-24
```

```
#(i)
min(Payments$DATE)
```

```
## [1] "2002-07-06"
```

```
max(Payments$DATE)
```

```
## [1] "2016-11-04"
```

```
max(Payments$DATE) - min(Payments$DATE)  # range of dates of all payments
```

```
## Time difference of 5235 days
```

```
#(ii)
payments.before <- Payments %>% filter(Payments$DATE < '2015-05-01') %>% nrow(); payments.before
```

```
## [1] 5763
```

```
total.payments <- nrow(Payments); total.payments
```

```
## [1] 1510216
```

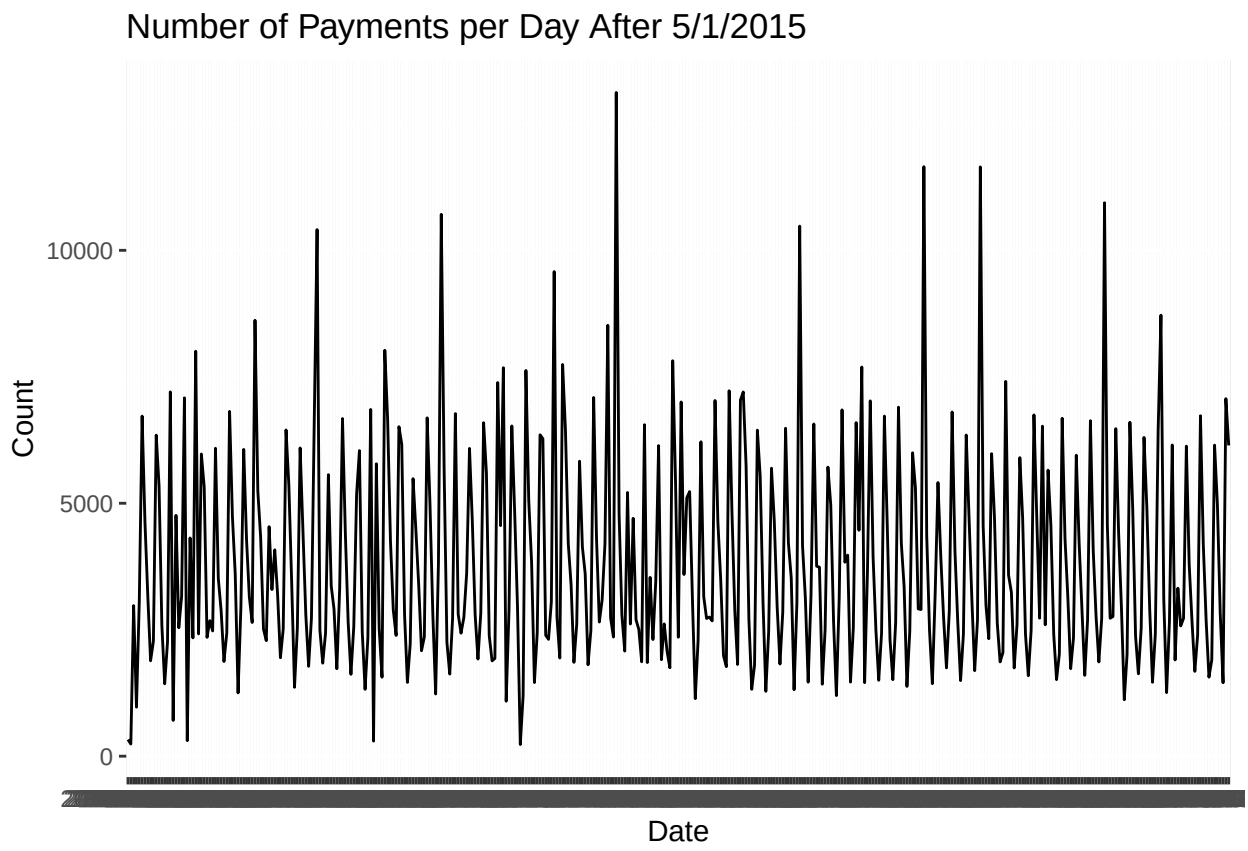
```
payments.before/total.payments  # percentage of payments made before 5/1/2015
```

```
## [1] 0.00381601
```

*b)*

```
# data
Payments.af.may <- Payments %>%
  filter(Payments$DATE >= '2015-05-01') %>%  # filtering payments after 5/1/2015
  arrange()
counts <- table(Payments.af.may$DATE)  # creating a frequency table
# setting table into df
new.payments <- setNames(data.frame(counts), c('Date', 'Count'))
```

```
ggplot(data = new.payments, mapping = aes(x = Date, y = Count, group = 1)) +
  geom_line() +  # plotting line plot
  labs(title = 'Number of Payments per Day After 5/1/2015')
```



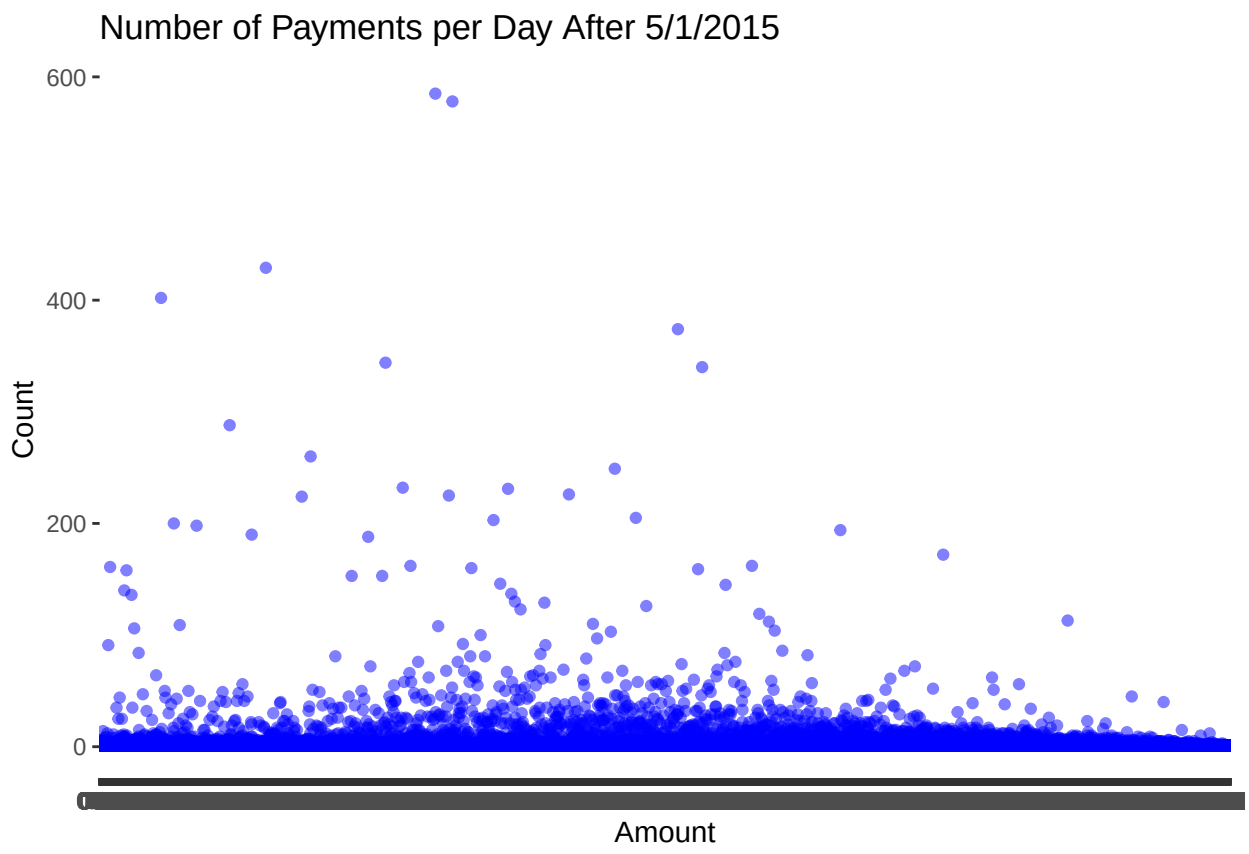

*c)*

The bimodal shape we see has its peaks around the summer months whereas the lows typically are around May, March, and October. These fluctuations depend on the payment dates. This happens because more people are paying on similar days due to child support payments being due. Depending on money consuming events, such as tax filings in the beginning of the year and holiday season at the end, payment dates will fluctuate throughout the month. The peaks in the data typically occur around the end of the month or the beginning of the month, which indicate timely payments as people tend to pay at the very start or end of each payment period.

*d)*

```
#sample
sample <- Payments[sample(nrow(Payments), 50000), ]
# creating df of counts
```

```
amnt <- setNames(data.frame(table(sample$PYMNT_AMT)), c('Amount', 'Count'))

ggplot(data = amnt, mapping = aes(x = Amount, y = Count, group = 1)) +
  geom_point(alpha = 0.5, color = 'blue') +
  labs(title = 'Number of Payments per Day After 5/1/2015')
```




The distribution is right skewed, which indicate that less people pay a larger amount. Because the distribution is right skewed, the data shows a median that is smaller than the mean, meaning that larger payments have an affect on the mean. The reason why the distribution could be right skewed can depend on the number of children that the parent has to account for. Since the average parent accounts for 1-3 children, the median data would gather around a smaller payment amount.

## Problem 4

```
# creating subset of payment + parents dataframe,
# selecting only the ID and the payment amount they made
id.payment <- subset(pool.pay.par, select = c("AP_ID", "PYMNT_AMT"))

# creating a frequency table indicating how many times the absent parent made a payment
# by tallying how many times AP_ID appears
ap_id.pymntct <- setNames(data.frame(table(id.payment$AP_ID)), c('AP_ID', 'Payment_Count'))
ap_id.pymntct$AP_ID <- as.integer(as.character(ap_id.pymntct$AP_ID))
```

```
# creating a dataframe with the total payment amount made by AP_ID
pool <- pool.pay.par %>%
  group_by(AP_ID) %>%
  summarise(Payment_Amount = sum(PYMNT_AMT)) %>%
  ungroup()

# creating a dataframe that represents the number of children an absent parent has
a <- Parents %>% left_join(Cases, by = "AP_ID")
b <- a %>% left_join(Children, by = "CASE_NUM")
child_per <- b %>%
  group_by(AP_ID) %>%
  summarise(Num_Children = n_distinct(ID), CASE_NUM = CASE_NUM) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'AP_ID'. You can override using the
## `.groups` argument.
```

```
# joining all the dataframes together and removing duplicate AP_IDs
absent_parent <- pool %>%
  left_join(ap_id.pymntct, by = 'AP_ID') %>%
  inner_join(child_per, by = 'AP_ID') %>%
  inner_join(Parents, by = 'AP_ID') %>%
  distinct(AP_ID, .keep_all =  TRUE); head(absent_parent)
```

```
## # A tibble: 6 x 14
##    AP_ID Payme~1 Payme~2 Num_C~3 CASE_~4 AP_AD~5 AP_DE~6 AP_CU~7 AP_AP~8 MARIT~9
##    <int>   <dbl>   <int>   <int>   <int> <chr>   <chr>   <lgl>     <int> <chr>
## 1 1.72e6   5175.      10       1  5.41e8 04      " "     NA            0 "N"
## 2 1.72e6   9360       36       1  2.12e8 04      " "     NA           30 "M"
## 3 1.72e6   2756.       5       7  9.71e8 01      " "     NA           32 "D"
## 4 1.72e6   2186.      15       1  4.11e8 03      " "     NA           22 "N"
## 5 1.72e6   1195.      18       1  2.91e8 03      " "     NA            0 " "
## 6 1.72e6   7254       36       1  9.11e8 03      " "     NA            0 " "
## # ... with 4 more variables: SEX_CD <chr>, RACE_CD <chr>, PRIM_LANG_CD <chr>,
## #   CITIZENSHIP_CD <chr>, and abbreviated variable names 1: Payment_Amount,
## #   2: Payment_Count, 3: Num_Children, 4: CASE_NUM, 5: AP_ADDR_ZIP,
## #   6: AP_DECEASED_IND, 7: AP_CUR_INCAR_IND, 8: AP_APPROX_AGE,
## #   9: MARITAL_STS_CD
```

*a)*

Null Hypothesis: There is no relationship between the number of children a parent is responsible for and the number of payments they make.

Alternative Hypothesis: There is a positive relationship between the number of children a parent is responsible for and the number of payments they make.

Assumptions: Since we are measuring a linear relationship, and the data is random and normally distributed, it is possible to use a correlation coefficient test.

Let alpha be a conservative estimate of 0.05.

```
# using cor.test to test association
cor.test(absent_parent$Num_Children, absent_parent$Payment_Count)
```

```
##
##  Pearson's product-moment correlation
##
## data:  absent_parent$Num_Children and absent_parent$Payment_Count
## t = 54.145, df = 28577, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2944775 0.3155080
## sample estimates:
##     cor
## 0.30503
```

Since the p-value of 2.2e-16 is less than the alpha of 0.05, we can reject the null hypothesis.

It is possible to say that there is a slightly positive correlation between the number of children a parent is responsible for and the number of payments they make. A correlation coefficient of 0.31 indicates there is a positive relationship between the number of children and the total number of payments they make, but that the linear association is weak.

*___*

Null Hypothesis: There is no relationship between the number of children a parent is responsible for and the total payment amount.

Alternative Hypothesis: There is a positive relationship between the number of children a parent is responsible for and the total payment amount.

Assumptions: Since we are measuring a linear relationship, and the data is random and normally distributed, it is possible to use a correlation coefficient test.

Let alpha be a conservative estimate of 0.05.

```
cor.test(absent_parent$Num_Children, absent_parent$Payment_Amount)
```

```
##
##  Pearson's product-moment correlation
##
## data:  absent_parent$Num_Children and absent_parent$Payment_Amount
## t = 42.518, df = 28577, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2329842 0.2547925
## sample estimates:
##       cor
## 0.2439192
```

Since the p-value of 2.2e-16 is less than the alpha of 0.05, we can reject the null hypothesis.

It is possible to say that there is a slightly positive correlation between the number of children a parent is responsible for and the total payment amount. A correlation coefficient of 0.24 indicates there is a positive relationship between the number of children and the total payment amount, but that the linear association is weak.

*b)*

```r
# creating a copy of Children df
ChildrenCopy <- Children
# finding the age of a child by finding the difference between their DOB and 1/1/2017
ChildrenCopy$Age <- as.Date('2017-01-01') - as.Date(ChildrenCopy$DATE_OF_BIRTH_DT, format =  "%m/%d/%Y")
# converting from days to years
ChildrenCopy$Year <- as.numeric(gsub("days", "", ChildrenCopy$Age / 365))

# joining the children df with the Cases df
ChildrenOfAP <- ChildrenCopy %>% left_join(Cases, by = "CASE_NUM")

# creating subset of Case #, children ID, and their age
ChildrenAge <- subset(ChildrenOfAP, select = c('CASE_NUM', 'ID', 'Year'))

# finding which absent parent belongs to which child by joining df based on case #s
# and removing duplicates
child.pay.age <- ChildrenAge %>%
  right_join(absent_parent, by = 'CASE_NUM') %>%
  distinct(ID, .keep_all = TRUE)

# finding the average age of each child by dividing
# the sum of the their ages by the number of children
avg_age <- child.pay.age %>%
  group_by(AP_ID) %>%
  summarise(avg_age = sum(Year) / Num_Children) %>%
  distinct(AP_ID, .keep_all =  TRUE); head(avg_age)
```

```
## `summarise()` has grouped output by 'AP_ID'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 6 x 2
## # Groups:   AP_ID [6]
##     AP_ID avg_age
##     <int>   <dbl>
## 1 1718626    8.07
## 2 1718628   13.5
## 3 1718629   10.5
## 4 1718630   10.3
## 5 1718631   NA
## 6 1718643   15.8
```

```r
#creating a df with AP_ID, the avg age, and the payment amounts made by each parent
age.count <- avg_age %>%
  left_join(absent_parent, by = 'AP_ID') %>%
  distinct(AP_ID, .keep_all =  TRUE)

# creating a subset of just average age and the payment amout
age.count.pymnt <- na.omit(subset(age.count, select = c('avg_age', 'Payment_Amount')))
head(age.count.pymnt)
```

```
## # A tibble: 6 x 2
##   avg_age Payment_Amount
##     <dbl>          <dbl>
```

```
## 1    8.07          5175.
## 2   13.5           9360
## 3   10.5           2756.
## 4   10.3           2186.
## 5   15.8           7254
## 6    8.12          1168.
```

Null Hypothesis: There is no association between the average age of children a parent is responsible for and the total payment amount.

Alternative Hypothesis: There is an association between the average age of children a parent is responsible for and the total payment amount.

Assumptions: Since we are measuring a linear relationship, and the data is random and normally distributed, it is possible to use a correlation coefficient test.

Let alpha be a conservative estimate of 0.05.

```
# using cor.test to test the association
cor.test(age.count.pymnt$avg_age, age.count.pymnt$Payment_Amount)
```

```
##
##  Pearson's product-moment correlation
##
## data:  age.count.pymnt$avg_age and age.count.pymnt$Payment_Amount
## t = -41.588, df = 27259, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2553921 -0.2330668
## sample estimates:
##        cor
## -0.2442618
```

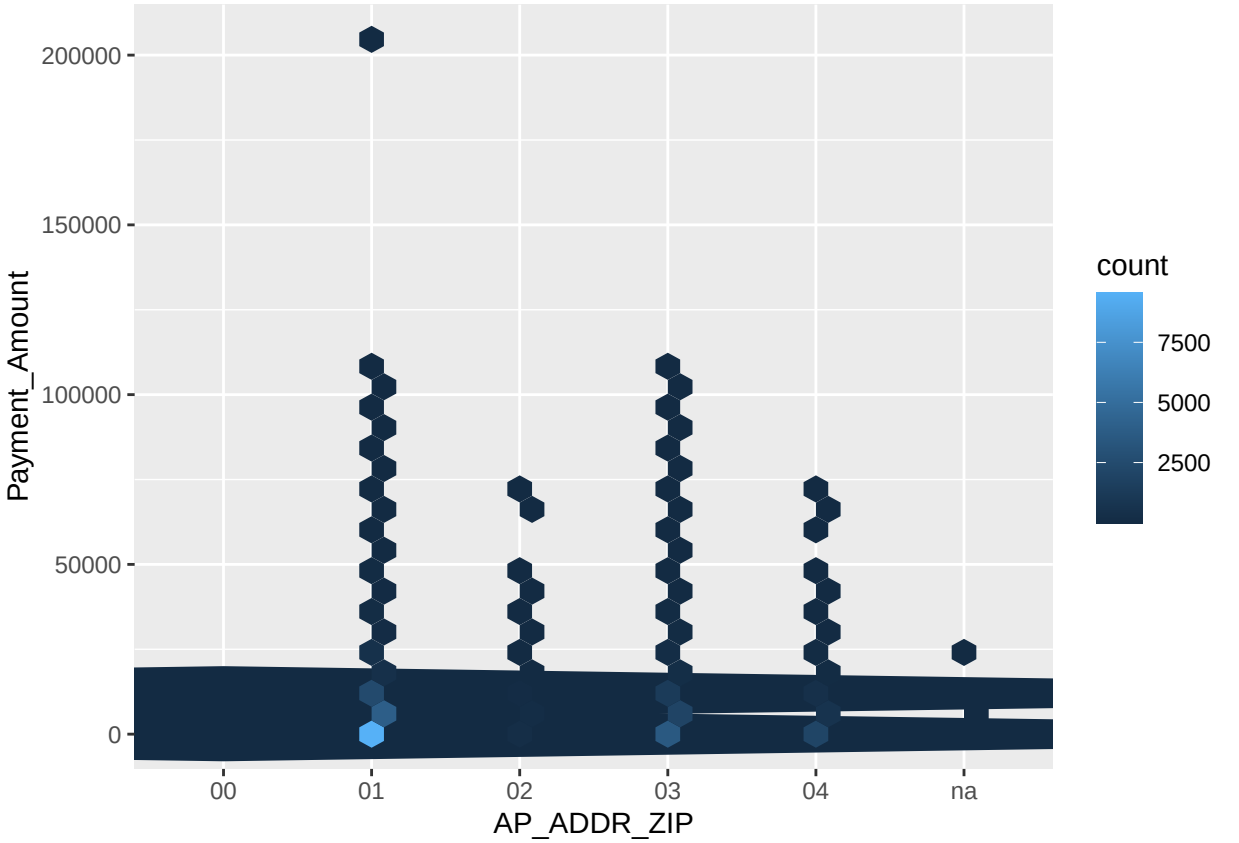
Since the p-value of 2.2e-16 is less than the alpha of 0.05, we can reject the null hypothesis.

It is possible to say that there is a slightly negative correlation between the average age of children a parent is responsible for and the total payment amount. A correlation coefficient of -0.24 indicates that as the average age increases, there is a decrease in total payment amount, but it is a weak linear association.

*c)*

```
# creating a subset of a df of just AP_ID, payment amount, and zip code
abs.zip <- subset(absent_parent, select = c('AP_ID', 'Payment_Amount', 'AP_ADDR_ZIP'))
```

```
# using ggplot to show data
ggplot(data = abs.zip, mapping = aes(x = AP_ADDR_ZIP, y = Payment_Amount)) +
  geom_hex()
```




Looking at the graph, we can conclude that location of the parent can anticipate the total payment amount. Zip codes 01 and 03 are more likely to have a higher total payment amount compared to 00, 02, and 04, suggesting that these are areas that must make more payment amounts.

*d)*

1)

Null Hypothesis: There is no relationship between the total amount of payments and the number of children.

Alternative Hypothesis: There is a relationship between the total amount of payments and the number of children.

2)

Null Hypothesis: There is no relationship between the total amount of payments and the average age of children.

Alternative Hypothesis: There is a relationship between the total amount of payments and the average age of children.

3)

Null Hypothesis: There is no relationship between the total amount of payments and the interaction between the number and average age of children.

Alternative Hypothesis: There is a relationship between the total amount of payments and the interaction between the number and average age of children.

Let alpha be a conservative estimate of 0.05.

```
summary(lm(Payment_Amount ~ Num_Children * avg_age, data = age.count))
```

```
##
## Call:
## lm(formula = Payment_Amount ~ Num_Children * avg_age, data = age.count)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -24959  -4088  -1570   1785 175404
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5759.227    149.936   38.41   <2e-16 ***
## Num_Children          1122.647     40.414   27.78   <2e-16 ***
## avg_age               -96.875      9.594  -10.10   <2e-16 ***
## Num_Children:avg_age  -41.365      3.594  -11.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7879 on 27257 degrees of freedom
##   (58 observations deleted due to missingness)
## Multiple R-squared:  0.09193,    Adjusted R-squared:  0.09183
## F-statistic: 919.8 on 3 and 27257 DF,  p-value: < 2.2e-16
```

Since the p-value of 2.2e-16 is less than the alpha of 0.05, we can reject the null hypotheses. Therefore, all variables are statistically significant.

There is evidence to suggest that there is a relationship between the number of children, average children age, and their interaction with the total payment amount.

Payment_Amount = 5759.227 + 1122.647(Num_Children) - 96.875(avg_age) - 41.365(Num_Children:avg_age)

There is a positive relationship between the number of children and the payment amount, a negative relationship between the average age and the payment amount, and a negative relationship between the interaction of the two variables and the payment amount.

# Problem 5

*a)*

```
# creating df with sum of payment amounts per day
pymnts.date <- Payments %>%
  group_by(DATE, AP_ID) %>%
  summarise(daily_pay = sum(PYMNT_AMT)) %>%
  arrange(AP_ID, DATE) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'DATE'. You can override using the
## `.groups` argument.
```

```r
# creating df with the sd and mean of each daily payment associated with AP_ID
pymnts.daily <- pymnts.date %>%
  group_by(AP_ID) %>%
  summarise(sd(daily_pay), mean(daily_pay)); head(pymnts.daily)
```

```
## # A tibble: 6 x 3
##     AP_ID `sd(daily_pay)` `mean(daily_pay)`
##     <int>           <dbl>             <dbl>
## 1 1718626            303.              518.
## 2 1718628              0               260
## 3 1718629            356.              551.
## 4 1718630            138.              146.
## 5 1718631              0                66.4
## 6 1718643              0               202.
```

Null Hypothesis: There is no association between the SD of total daily payments and the average of total daily payments.

Alternative Hypothesis: There is an association between the SD of total daily payments and the average of total daily payments.

Assumptions: Since we are measuring a linear relationship, and the data is random and normally distributed, it is possible to use a correlation coefficient test.

Let alpha be a conservative estimate of 0.05.

```r
# using cor.test to test association
cor.test(pymnts.daily$`sd(daily_pay)`, pymnts.daily$`mean(daily_pay)`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  pymnts.daily$`sd(daily_pay)` and pymnts.daily$`mean(daily_pay)`
## t = 190.18, df = 26997, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7515515 0.7617489
## sample estimates:
##       cor
## 0.7566962
```

Since the p-value of 2.2e-16 is less than the alpha of 0.05, we can reject the null hypothesis.

There is evidence to suggest that there is an association between the SD of total daily payments and the average of total daily payments. Since the correlation coefficient is 0.76, there is evidence to say that the association is a strong linear association.

*b)*

```r
#finding the CV values and rounding to the ten thousandths place
pymnts.daily$CV <- round(pymnts.daily$`sd(daily_pay)`/ pymnts.daily$`mean(daily_pay)`, 4)

# omitting NA values
pymnts.daily.omit <- na.omit(pymnts.daily)

min(pymnts.daily.omit$CV)  # finding min CV values
```

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

```
median(pymnts.daily.omit$CV) # finding median CV values
```

```
## [1] 0.3037
```

```
max(pymnts.daily.omit$CV) # finding max CV values
```

```
## [1] 7.2285
```

```
low <- pymnts.daily.omit[pymnts.daily.omit$CV == 0,]
med <- pymnts.daily.omit[pymnts.daily.omit$CV == 0.3037,]
```

```
# finding representative low CV parent
low.sample <- sample_n(low, 1); low.sample
```

```
## # A tibble: 1 x 4
##     AP_ID `sd(daily_pay)` `mean(daily_pay)`    CV
##     <int>           <dbl>             <dbl> <dbl>
## 1 1759725               0              473.     0
```

```
# finding representative medium CV parent
med.sample <- sample_n(med, 1); med.sample
```

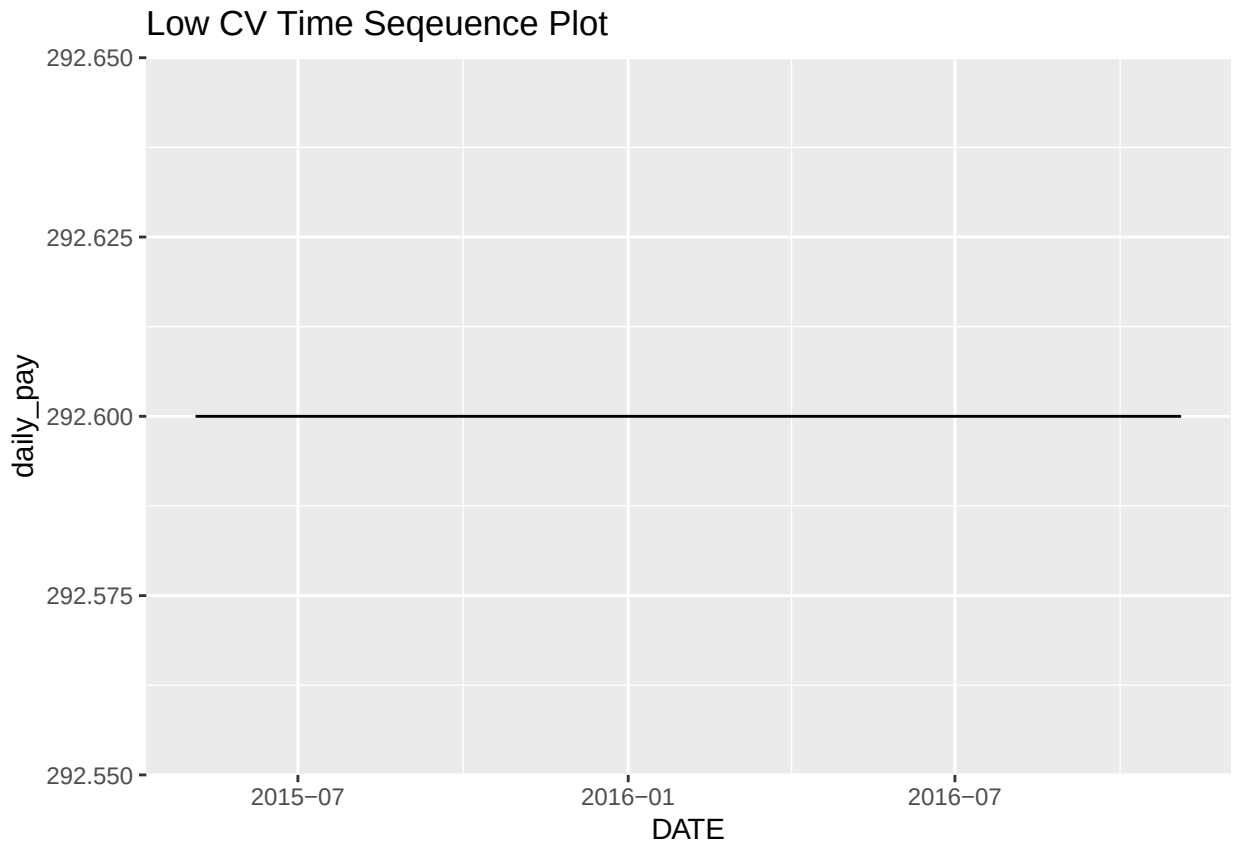
```
## # A tibble: 1 x 4
##     AP_ID `sd(daily_pay)` `mean(daily_pay)`    CV
##     <int>           <dbl>             <dbl> <dbl>
## 1 1767885            18.0              59.4 0.304
```

```
# finding representative high CV parent
high <- pymnts.daily.omit[pymnts.daily.omit$CV == 7.2285,]; high
```

```
## # A tibble: 1 x 4
##     AP_ID `sd(daily_pay)` `mean(daily_pay)`    CV
##     <int>           <dbl>             <dbl> <dbl>
## 1 1725000            390.              53.9  7.23
```

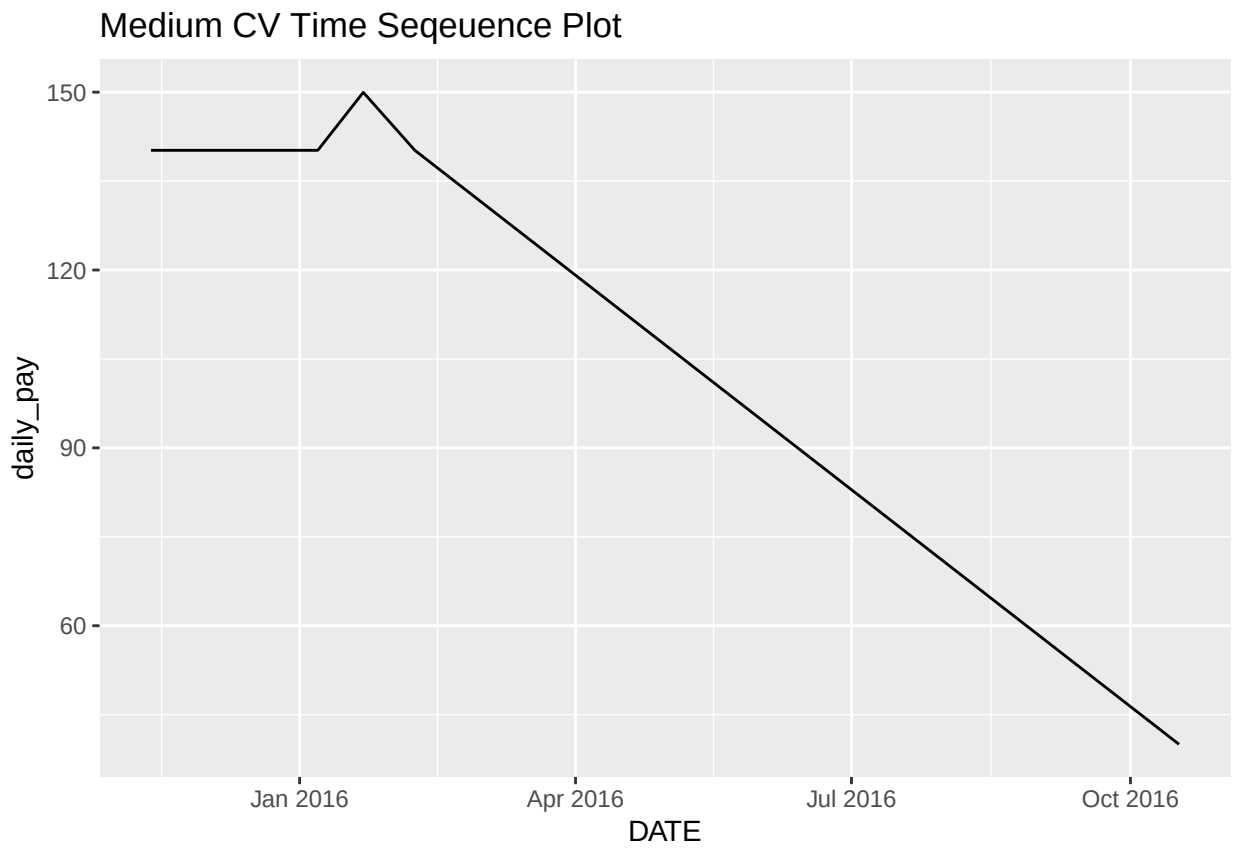
For the lowest CV of 0, a representative parent is 1748799.

```
# time plot for lowest CV
low.cv.data <- pymnts.date %>% filter(AP_ID == 1748799)

ggplot(data = low.cv.data, mapping = aes(x = DATE, y = daily_pay)) +
  geom_line() +
  labs(title = 'Low CV Time Seqeuence Plot')
```



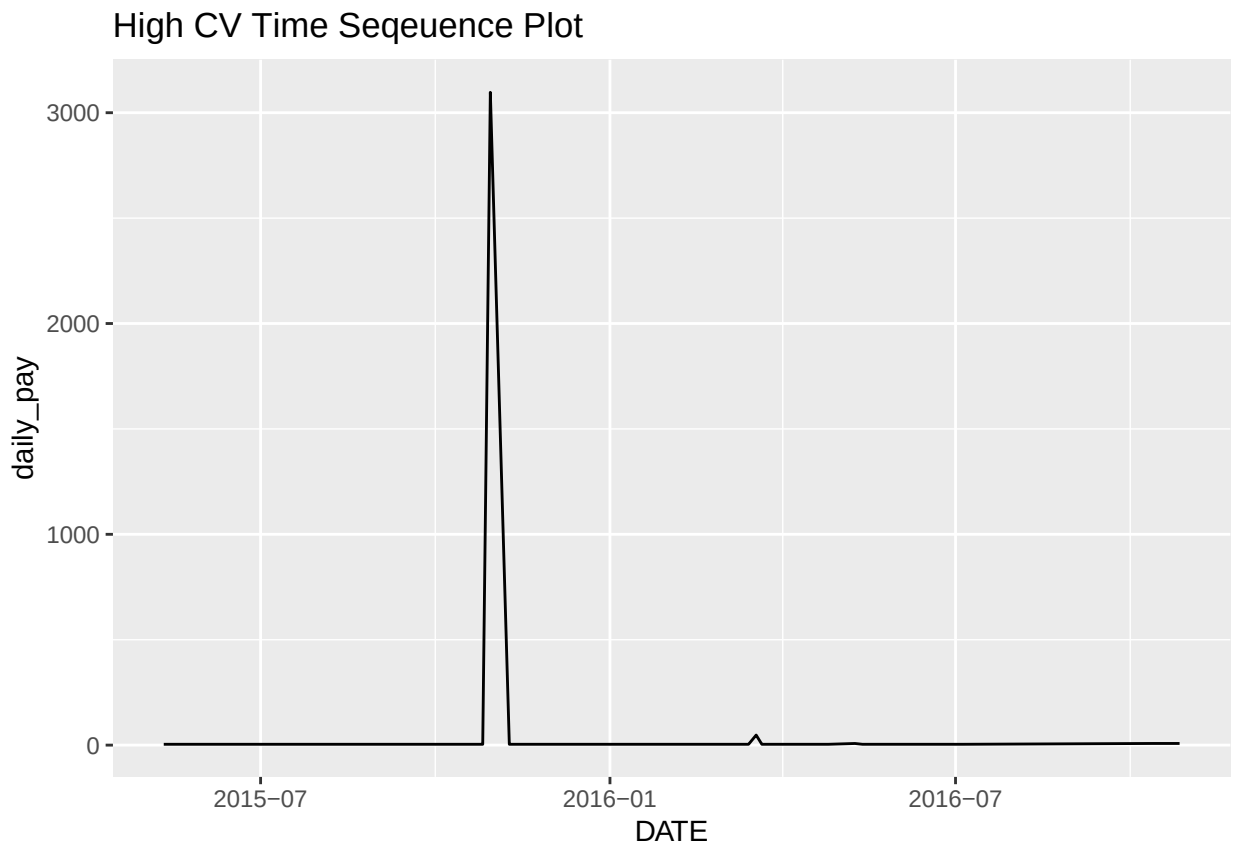

For a median CV of 0.3037, a representative parent is 1801087.

```
# time plot for medium CV
med.cv.data <- pymnts.date %>% filter(AP_ID == 1801087)

ggplot(data = med.cv.data, mapping = aes(x = DATE, y = daily_pay)) +
  geom_line() +
  labs(title = 'Medium CV Time Seqeuence Plot')
```




For a high CV of 7.2285, a representative parent is 1725000.

```
# time plot for highest CV
high.cv.data <- pymnts.date %>% filter(AP_ID == 1725000)

ggplot(data = high.cv.data, mapping = aes(x = DATE, y = daily_pay)) +
  geom_line() +
  labs(title = 'High CV Time Seqeuence Plot')
```




*c)*

```
# creating df by joining the CV df and the absent_parent df
daily.par.pay <- pymnts.daily.omit %>% left_join(absent_parent, by = "AP_ID")

# creating a subset df of AP_IDs, CV, and payment amount
daily.par.pay <- subset(daily.par.pay, select = c("AP_ID", "CV", "Payment_Amount"))
head(daily.par.pay)
```

```
## # A tibble: 6 x 3
##     AP_ID    CV Payment_Amount
##     <int> <dbl>          <dbl>
## 1 1718626 0.585          5175.
## 2 1718628 0              9360
## 3 1718629 0.646          2756.
## 4 1718630 0.946          2186.
## 5 1718631 0              1195.
## 6 1718643 0              7254
```

Null Hypothesis: There is no association between the total amount of payments and the CV of payments.

Alternative Hypothesis: There is an association between the total amount of payments and the CV of payments.

Assumptions: Since we are measuring a linear relationship, and the data is random and normally distributed, it is possible to use a correlation coefficient test.

Let alpha be a conservative estimate of 0.05.

```r
# using cor.test to test association
cor.test(daily.par.pay$CV, daily.par.pay$Payment_Amount)
```

```
##
##  Pearson's product-moment correlation
##
## data:  daily.par.pay$CV and daily.par.pay$Payment_Amount
## t = 6.9466, df = 26997, p-value = 3.828e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.03032707 0.05414107
## sample estimates:
##       cor
## 0.04224007
```

Since the p-value of 3.828e-12 is less than the alpha of 0.05, we can reject the null hypotheses.

There is evidence to suggest that there is an association between the total amount of payments and the CV of payments. With a correlation coefficient of 0.04, there is a weaker linear association between CV and payment amount.

*d)*

1)

Null Hypothesis: There is no relationship between the CV and number of children.

Alternative Hypothesis: There is a relationship between the CV and number of children.

2)

Null Hypothesis: There is no relationship between the CV and average age.

Alternative Hypothesis: There is a relationship between the CV and average age.

3)

Null Hypothesis: There is no relationship between the CV and zip code 01.

Alternative Hypothesis: There is a relationship between the CV and zip code 01.

4)

Null Hypothesis: There is no relationship between the CV and zip code 02.

Alternative Hypothesis: There is a relationship between the CV and zip code 02.

5)

Null Hypothesis: There is no relationship between the CV and zip code 03.

Alternative Hypothesis: There is a relationship between the CV and zip code 03.

6)

Null Hypothesis: There is no relationship between the CV and zip code 04.

Alternative Hypothesis: There is a relationship between the CV and zip code 04.

7)

Null Hypothesis: There is no relationship between the CV and zip code NA.

Alternative Hypothesis: There is a relationship between the CV and zip code NA.

Let alpha be a conservative estimate of 0.05.

```
bonus <- pymnts.daily.omit %>%
  left_join(absent_parent, by = "AP_ID") %>%
  left_join(avg_age, by = "AP_ID")

# creating subset df of AP_IDs, CVs, the number of children, avg_age, and zip
bonus <- na.omit(subset(bonus, select = c("AP_ID", "CV", "Num_Children", "avg_age", "AP_ADDR_ZIP")))
head(bonus)
```

```
## # A tibble: 6 x 5
##     AP_ID    CV Num_Children avg_age AP_ADDR_ZIP
##     <int> <dbl>        <int>   <dbl> <chr>
## 1 1718626 0.585            1    8.07 04
## 2 1718628 0                1   13.5  04
## 3 1718629 0.646            7   10.5  01
## 4 1718630 0.946            1   10.3  03
## 5 1718643 0                1   15.8  03
## 6 1718656 0.314            1    8.12 01
```

```
summary(lm(CV ~ Num_Children + avg_age + AP_ADDR_ZIP , data = bonus))
```

```
##
## Call:
## lm(formula = CV ~ Num_Children + avg_age + AP_ADDR_ZIP, data = bonus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8511 -0.4355 -0.2044  0.1361  6.5603
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0579029  0.6776868   0.085    0.932
## Num_Children   0.0215650  0.0023984   8.991  < 2e-16 ***
## avg_age       -0.0035480  0.0005407  -6.562  5.4e-11 ***
## AP_ADDR_ZIP01  0.4312410  0.6776722   0.636    0.525
## AP_ADDR_ZIP02  0.4180007  0.6782330   0.616    0.538
## AP_ADDR_ZIP03  0.4400901  0.6776954   0.649    0.516
## AP_ADDR_ZIP04  0.4907403  0.6777668   0.724    0.469
## AP_ADDR_ZIPna  0.4534915  0.7187754   0.631    0.528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6776 on 25759 degrees of freedom
## Multiple R-squared:  0.007299,   Adjusted R-squared:  0.007029
## F-statistic: 27.06 on 7 and 25759 DF,  p-value: < 2.2e-16
```

Since the p-value of 2.2e-16 and 5.4e-11 are less than the alpha of 0.05, we can reject the null hypotheses for the number of children and average age of children.

There is evidence to suggest that there is a relationship between the number of children and average children age with the CV.

Alternatively, p-values that are greater than 0.4 are greater than our alpha of 0.05 so we fail to reject the null hypotheses that there are no relationship between the CV and zip codes 01, 02, 03, 04, and NA.

This suggests that the number of children and average children age affect the CV while the zip codes do not.