```
# Make sure jsonlite package is installed
# Load jsonlite Library
install.packages("jsonlite")
install.packages("plyr")
install.packages("scales")
install.packages("plyr")
install.packages("stringr")
library(jsonlite)
library(scales)
library (plyr)
library(stringr)
#Stream in json file, convert NULL to NA
json_file <- stream_in(file("ida_wrangling_exercise_data.2017-02-13.jsonl.gz"))
json_file$address[json_file$address=="NULL"] <- "NA"
json_file$name[json_file$name=="NULL"] <- "NA"



# Start by making a list of all of the nested named fields that appear in any record.
# Concatenate nested field names using a period '.' to defind named fields for nested records.
# Present the list in alphabetical order. For example, if our data file contained the following


# ------------Now Operating on the complete data set-------------------------------
large_data <- json_file

# 1. Concatenate fields-----------------------------------

# Conatenate name fields
# This will unpack nested name lists and return strings for names
library (plyr)
nested_names_large <- ldply (large_data$name, data.frame)
nested_names_large <- data.frame(lapply(nested_names_large, as.character), stringsAsFactors=FALSE)
# Renames the columns to desired format
colnames(nested_names_large) <- c("name.first","name.last", "name.middle", "name")

# Delete old names column
large_data$name <- NULL
# Combine jsonl data with new string name columns
large_data <- cbind(large_data, nested_names_large)


# Concatenate address fields
# This will unpack nested lists and return strings for names
nested_address_large <- ldply (large_data$address, data.frame)
nested_address_large <- data.frame(lapply(nested_address_large, as.character), stringsAsFactors=FALSE)
# Renames the columns to desired format
colnames(nested_address_large) <- c("address.street","address.city", "address.state", "address.zip","address")

# Delete old address column
large_data$address <- NULL
# Combine new string name columns
large_data <- cbind(large_data, nested_address_large)

# Arrange columns in alphabetical order
large_data <- large_data[ , order(names(large_data)) ]

# List Column (field names) names
names_list <- colnames(large_data)
print(names_list)


# 2. What percentage of the records contain the field?---------------------------
# Determine number of instances for field, divide by number of rows, then print


# -------Field: address-------
a <- percent(sum(complete.cases(large_data$address))/nrow(large_data))

# -------Field: address.city-------
a.city <- percent(sum(complete.cases(large_data$address.city))/nrow(large_data))

# -------Field: address.state-------
a.state <- percent(sum(complete.cases(large_data$address.state))/nrow(large_data))

# -------Field: address.street-------
a.street <- percent(sum(complete.cases(large_data$address.street))/nrow(large_data))

# -------Field: address.zip-------
a.zip <- percent(sum(complete.cases(large_data$address.zip))/nrow(large_data))
```

```
# -------Field: name-------
n <- percent(sum(complete.cases(large_data$name))/nrow(large_data))

# -------Field: name.first-------
n.first <- percent(sum(complete.cases(large_data$name.first))/nrow(large_data))

# -------Field: name.last-------
n.last <- percent(sum(complete.cases(large_data$name.last))/nrow(large_data))

# -------Field: name.middle-------
n.middle <- percent(sum(complete.cases(large_data$name.middle))/nrow(large_data))


field_percentage <- c(a, a.city,a.state,a.street,a.zip,n,n.first,n.last,n.middle)
names(field_percentage) <-
c("address","address.city","address.state","address.street","address.zip","name","name.fist","name.last","name.middle")

field_percentage <- data.frame(field_percentage)
print(field_percentage)

# 2. What are the five most common values of the field?-------
# Plot table of first five common elements

# -------Field: address.city-------
ac <- sort(table(large_data$address.city),decreasing=TRUE)[1:5]
barplot(ac, main = "Common address city", space=3,col=heat.colors(5))

# -------Field: address.state-------
as <- sort(table(large_data$address.state),decreasing=TRUE)[1:5]
barplot(as, main = "Common address state", space=3,col=heat.colors(5))

# -------Field: address.zip-------
az <- sort(table(large_data$address.zip),decreasing=TRUE)[1:5]
barplot(az, main = "Common address zip codes", space=3,col=heat.colors(5))

# -------Field: name-------
n0 <- sort(table(large_data$name), decreasing=TRUE)[1:6]
barplot(n0[2:6], main = "Common names", space=3,col=heat.colors(5))

# -------Field: name.first-------
nf <- sort(table(large_data$name.first),decreasing=TRUE)[1:5]
barplot(nf, main = "Common first names", space=3,col=heat.colors(5))

# -------Field: name.last-------
nl <- sort(table(large_data$name.last),decreasing=TRUE)[1:5]
barplot(nl, main = "Common last names", space=3,col=heat.colors(5))

# -------Field: name.middle-------
nm <- sort(table(large_data$name.middle),decreasing=TRUE)[1:5]
barplot(nm, main = "Common middle names", space=3,col=heat.colors(5))

# 3. How many distinct first names appear in this data set? Explain your procedure for identifying distinct first
names.--
# Examine the first few names in small set
head(large_data$name.first)

# How many unique first names in small set, excluding NA values
paste("Number of unique first names:", length(unique(na.exclude(large_data$name.first))))

# 4. How many unique street names?
# Strips numbers from address.street
street_names <- lapply(strsplit(large_data$address.street, "(?<=\\d)\\b ", perl=T), function(x) if (length(x)<2)
c("", x) else x)
street_names <- do.call(rbind, street_names)
colnames(street_names) <- c("Street Number", "Street Name")

# Find Unique names in street_names, excluding NA
street_names<- data.frame(street_names)
street_names <- data.frame(lapply(street_names, as.character), stringsAsFactors=FALSE)
paste("Unique street names:",length(unique(na.exclude(street_names$Street.Name))))
# With more time, I would strip the street names from address column too
# Five most common street names
barplot(sort(table(street_names$Street.Name),decreasing=TRUE)[1:5], main = "Five most common street names", space=3,
col=heat.colors(5) )

# 5. What are the 5 most common US area codes in the phone number field?
# Explain your approach to identify the US area codes in this data set.

area_codes <- large_data$phone
# Stripped all special characters.
area_codes <- gsub("[[:punct:]]", " ", area_codes)
```

```
# Strip "1 "
area_codes <- gsub("1 ", " ", area_codes)
# Remove white space
area_codes <- gsub(" ", "", area_codes, fixed = TRUE)
# The step below strips the first three numbers
area_codes <- substring(area_codes, 1, 3)
barplot(sort(table(area_codes),decreasing=TRUE)[1:5], main = "Five most common area codes", space=3,
col=heat.colors(5) )
```