Challenge 4

Naiyu Jiang

Question 0: Pre-process the dataset.

I assume that the responses from participants are trustworthy and responses with overly large scales are just careless errors made by respondents who wanted to tick the largest value. Therefore, I will impute those wrong values with the upper bound of theirs respective scale.

```
# Load libraries
library(tidyverse)
library(here)
library(tictoc)
library(h2o) # ML engine for fitting the AE
library(bit64) # speeds up some h2o computation

# read in ANES 2016
anes <- read_csv(here("data", "anes_2016.csv"))

# select the fourteen survey questions on salient social issues
anes_short <- anes %>%
  select(vaccine, autism, birthright_b, forceblack, forcewhite,
         stopblack, stopwhite, freetrade, aa3,
         warmdo, finwell, childcare, healthspend,
         minwage, amer_ident, race_ident) %>%
  mutate(strong_amer_ident = case_when(amer_ident == 1 ~ 1,
                                       amer_ident == 2 ~ 1, TRUE ~ 0),
         strong_race_ident = case_when(race_ident == 1 ~ 1,
                                       race_ident == 2 ~ 1, TRUE ~ 0))

# preprocess and clean the 14 variables
anes_short <- anes_short %>%
  mutate(birthright_b = replace(birthright_b, birthright_b > 7, 7),
         stopwhite = replace(stopwhite, stopwhite > 5, 5),
         aa3 = replace(aa3, aa3 > 7, 7),
         warmdo = replace(warmdo, warmdo > 7, 7),
         finwell = replace(finwell, finwell > 7, 7),
         healthspend = replace(healthspend, healthspend > 7, 7),
         minwage = replace(minwage, minwage > 4, 4)) %>%
  as.data.frame()

skimr::skim(anes_short)
```

Table 1: Data summary

Name	anes_short
Number of rows	1200
Number of columns	18

Table 1: Data summary

Column type frequency:	
numeric	18
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
vaccine	0	1	2.30	1.74	1	1	1	3	7	
autism	0	1	4.48	1.56	1	3	5	6	6	
birthright_b	0	1	5.60	2.07	1	4	7	7	7	
forceblack	0	1	3.59	1.05	1	3	3	5	5	
forcewhite	0	1	2.68	0.86	1	2	3	3	5	
stopblack	0	1	3.66	1.03	1	3	4	5	5	
stopwhite	0	1	2.69	0.93	1	2	3	3	5	
freetrade	0	1	3.90	1.62	1	3	4	5	7	
aa3	0	1	5.72	1.90	1	4	7	7	7	
warmdo	0	1	3.20	2.05	1	1	3	4	7	
finwell	0	1	4.59	1.73	1	3	5	6	7	
childcare	0	1	3.38	1.79	1	2	3	4	7	
healthspend	0	1	3.26	1.91	1	2	3	4	7	
minwage	0	1	1.60	0.89	1	1	1	2	4	
amer_ident	0	1	2.14	1.25	1	1	2	3	5	
race_ident	0	1	2.80	1.47	1	1	3	4	5	
strong_amer_ident	0	1	0.66	0.47	0	0	1	1	1	
strong_race_ident	0	1	0.46	0.50	0	0	0	1	1	

The American Identity

Question 1

Here, I set `predictors` by removing the dichotomous American/ race identity features from the `anes_short` dataset. Therefore, the `predictors` only include the 14 questions.

```
# fitting
set.seed(1234)
anes_short$strong_amer_ident <- as.factor(anes_short$strong_amer_ident)

# initializing the h2o cluster; have to do this to work with the h2o engine
my_h2o <- h2o.init()
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         11 hours 39 minutes
##     H2O cluster timezone:       America/New_York
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.32.0.1
##     H2O cluster version age:    6 months and 28 days !!!
##     H2O cluster name:           H2O_started_from_R_nyjiang_edm253
```

```
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   3.53 GB
##     H2O cluster total cores:    8
##     H2O cluster allowed cores:  8
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, XGBoost, Algos, AutoML, Core V3, TargetEncoder, Core V4
##     R Version:                  R version 3.6.1 (2019-07-05)
```

```
## Warning in h2o.clusterInfo():
## Your H2O cluster version is too old (6 months and 28 days)!
## Please download and install the latest version from http://h2o.ai/download/
```

```
# h2o df
anes_h2o <- anes_short %>%
  as.h2o()
```

```
##   |                                                                      |
```

```
# Store response and predictors separately (per h2o syntax)
response <- c("amer_ident", "strong_amer_ident", "race_ident", "strong_race_ident")
predictors <- setdiff(colnames(anes_h2o), response)
```

Note that here we do not need to do the supervised task, so I just use `anes_h2o` as the training frame instead of the train set.

```
{
  tic()
ae1 <- h2o.deeplearning(x = predictors, # input layer
                        training_frame = anes_h2o, # which data frame we are using for this task
                        autoencoder = TRUE,
                        hidden = 2, # number of hidden layers, and how many nodes in each.
                        epochs = 100,  # number of times to see the full input data
                        activation = "Tanh") # recall, non-linear activation
  toc()
}
```

```
##   |                                                                      |
## 1.243 sec elapsed
```

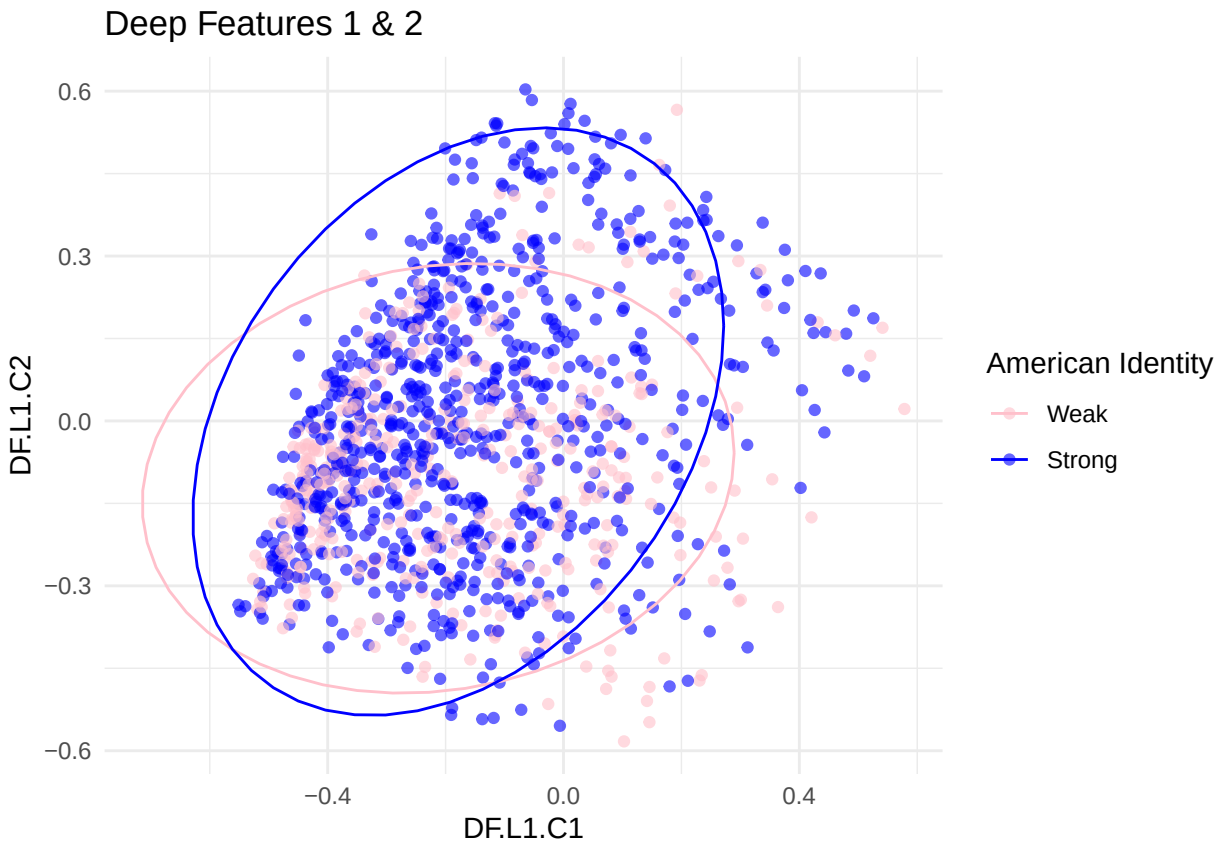
```
# feature extraction
ae1_codings <- h2o.deepfeatures(ae1,
                                data = anes_h2o,
                                layer = 1) %>%
  as.data.frame() %>%
  mutate(strong_amer_ident = as.vector(anes_h2o[ , 17])) # retain the 17th feature
```

```
##   |                                                                      |
```

Question 2

```
p1 <- ggplot(ae1_codings, aes(x = DF.L1.C1,
                              y = DF.L1.C2,
                              color = factor(strong_amer_ident))) +
```

```
  geom_point(alpha = 0.6) +
  stat_ellipse() +
  scale_color_manual(values=c("pink", "blue"),
                     name="American Identity",
                     breaks=c("0", "1"),
                     labels=c("Weak", "Strong")) +
  labs(title = "Deep Features 1 & 2",
       color = "American Identity") +
  theme_minimal()
p1
```



From the graph, we can see that there is no obvious separation in the projection (question) space along senses of American identity because the blue and pink points are all blended together without clear divisions. If there is any expected structure, we should be able to see that points clearly group at either extreme, and blending near the middle. But from this AE model, we fail to do so. AE can learn the most unique, interesting variance in the data. In this case, where we can not observe clear separation against American Identity, it means that `American Identity` is not the most unique feature the projection (question) space captures. Among 14 questions, not all questions are relevant to whether people have a strong or weak American Identity. I guess that is the reason why we are unable to observe clear separations from the output layer when we aggregate them all together.

Question 3

```
{
  tic()
ae2 <- h2o.deeplearning(x = predictors, # input layer
                        training_frame = anes_h2o, # which data frame we are using for this task
```

```
                          autoencoder = TRUE,
                          hidden = c(2, 2, 2), # number of hidden layers, and how many nodes in each.
                          epochs = 100,  # number of times to see the full input data
                          activation = "Tanh") # recall, non-linear activation
  toc()
}
```

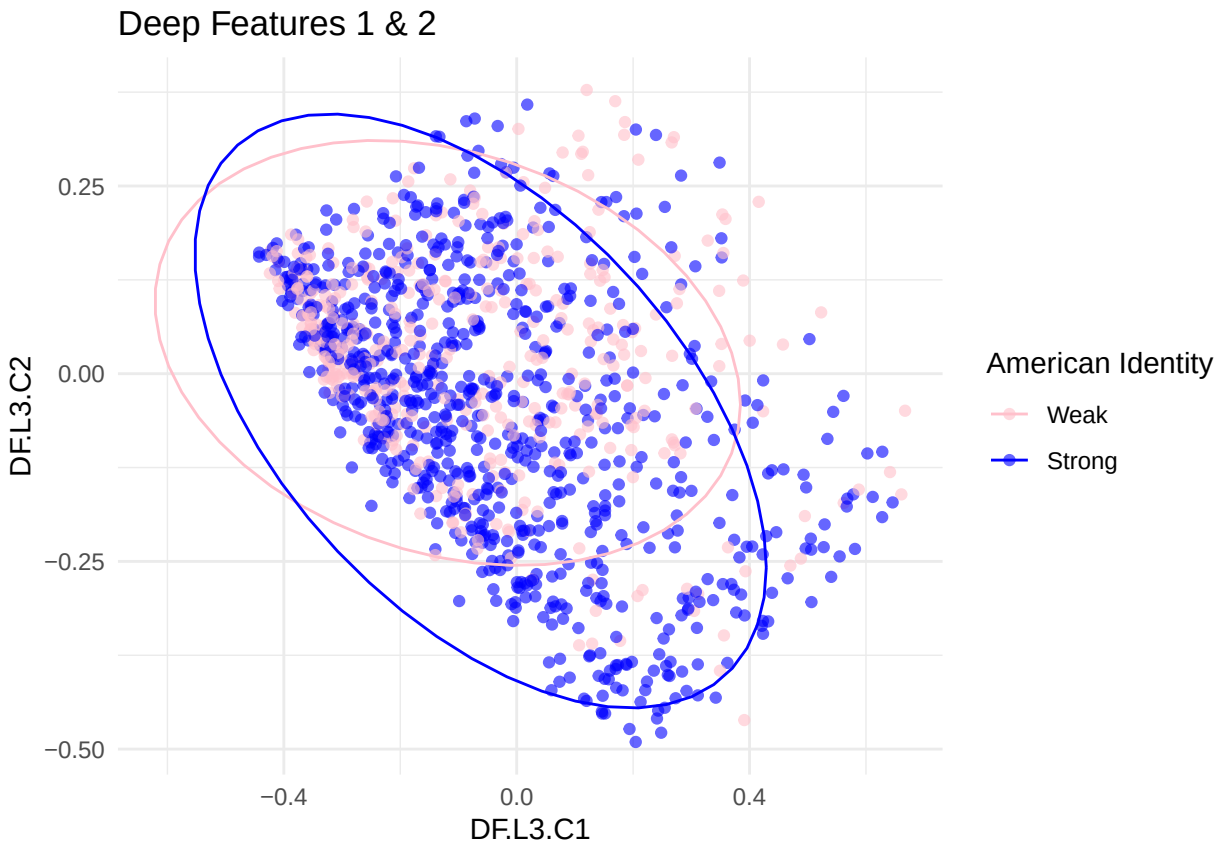
```
##   |                                                                      |
## 1.272 sec elapsed
```

```
# feature extraction
ae2_codings <- h2o.deepfeatures(ae2,
                                data = anes_h2o,
                                layer = 3) %>%
  as.data.frame() %>%
  mutate(strong_amer_ident = as.vector(anes_h2o[ , 17])) # retain the 17th feature
```

```
##   |                                                                      |
```

Question 4

```
p2 <- ggplot(ae2_codings, aes(x = DF.L3.C1,
                              y = DF.L3.C2,
                              color = factor(strong_amer_ident))) +
  geom_point(alpha = 0.6) +
  stat_ellipse() +
  scale_color_manual(values=c("pink", "blue"),
                     name="American Identity",
                     breaks=c("0", "1"),
                     labels=c("Weak", "Strong")) +
  labs(title = "Deep Features 1 & 2",
       color = "American Identity") +
  theme_minimal()
p2
```



From the graph, we can see that there is no obvious separation in the projection (question) space along senses of American identity, since the blue and pink points are blended together without clear divisions. Compared to plot 1, I think the separation is not significantly clearer. In general, the more hidden layers, and thus the more processing that occurs to efficiently encode the input space and learn the lower-dimensional representation of the data, the deeper the AE. In this case, which we need a complex framework to efficiently learn, we still cannot observe any pattern in the deeper hidden layer. It strongly support that there is no pattern or structure in the projection (question) space along senses of American identity. People's American identity is not present in their responses to social questions. Meanwhile, we should be always aware that there is a trade-off in complexity. When we deepen the network, the model would be more unbiased but with more variance, and therfore we may have a better representation of the original data but it's easy to have overfitting problem with high computational costs.

The Race Identity

Question 5

The procesure is exact same with Question 1. I just repeat here.

```
# fitting
set.seed(1234)
anes_short$strong_race_ident <- as.factor(anes_short$strong_race_ident)

# initializing the h2o cluster; have to do this to work with the h2o engine
my_h2o <- h2o.init()
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
```

```
##     H2O cluster uptime:         11 hours 39 minutes
##     H2O cluster timezone:       America/New_York
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.32.0.1
##     H2O cluster version age:    6 months and 28 days !!!
##     H2O cluster name:           H2O_started_from_R_nyjiang_edm253
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   3.53 GB
##     H2O cluster total cores:    8
##     H2O cluster allowed cores:  8
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, XGBoost, Algos, AutoML, Core V3, TargetEncoder, Core V4
##     R Version:                  R version 3.6.1 (2019-07-05)
```

```
## Warning in h2o.clusterInfo():
## Your H2O cluster version is too old (6 months and 28 days)!
## Please download and install the latest version from http://h2o.ai/download/
```

```
# h2o df
anes_h2o <- anes_short %>%
  as.h2o()
```

```
##   |                                                                      |
```

```
# Store response and predictors separately (per h2o syntax)
response <- c("amer_ident", "strong_amer_ident", "race_ident", "strong_race_ident")
predictors <- setdiff(colnames(anes_h2o), response)
```

```
{
  tic()
ae3 <- h2o.deeplearning(x = predictors, # input layer
                        training_frame = anes_h2o, # which data frame we are using for this task
                        autoencoder = TRUE,
                        hidden = 2, # number of hidden layers, and how many nodes in each.
                        epochs = 100,  # number of times to see the full input data
                        activation = "Tanh") # recall, non-linear activation
  toc()
}
```

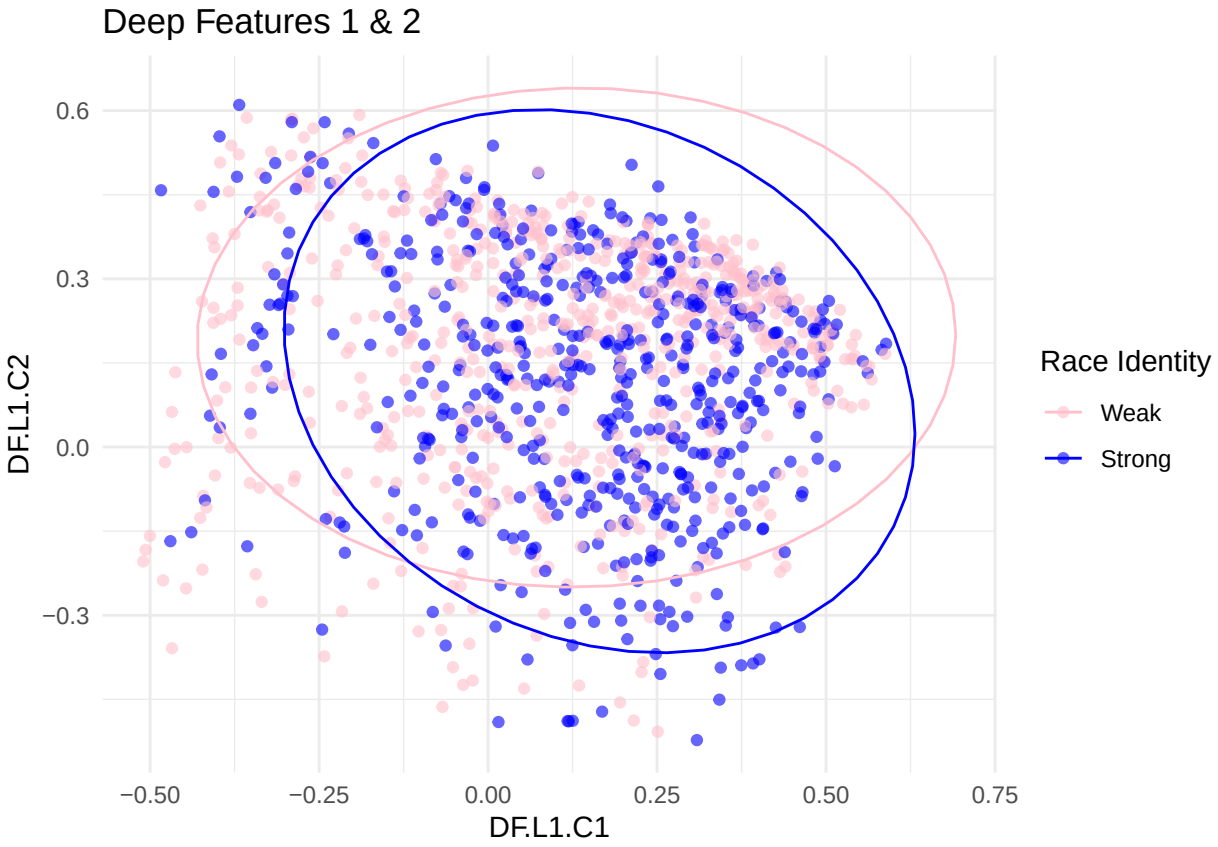
```
##   |                                                                      |
## 1.259 sec elapsed
```

```
# feature extraction
ae3_codings <- h2o.deepfeatures(ae3,
                                data = anes_h2o,
                                layer = 1) %>%
  as.data.frame() %>%
  mutate(strong_race_ident = as.vector(anes_h2o[ , 18])) # retain the 17th feature
```

```
##   |                                                                      |
```

Question 6

```
p3 <- ggplot(ae3_codings, aes(x = DF.L1.C1,
                              y = DF.L1.C2,
                              color = factor(strong_race_ident))) +
  geom_point(alpha = 0.6) +
  stat_ellipse() +
  scale_color_manual(values=c("pink", "blue"),
                     name="Race Identity",
                     breaks=c("0", "1"),
                     labels=c("Weak", "Strong")) +
  labs(title = "Deep Features 1 & 2",
       color = "Race Identity") +
  theme_minimal()
p3
```



From the graph, we can see that there is no obvious separation in the projection (question) space along senses of Race identity because the blue and pink points are blended together without clear divisions. If there is any expected structure, we should be able to see that points clearly group at either extreme, and blending near the middle. But from this AE model, we fail to do so. AE can learn the most unique, interesting variance in the data. In this case, where we can not observe clear separation against Race Identity, it means that `Race Identity` is not the most unique feature the projection (question) space captures. Among 14 questions, not all questions are relevant to whether people have a strong or weak Race Identity, although there are some relavant to people's sense of their racial identities, such as `forceblack`, `stopblack`. I guess that is the reason why we are unable to observe clear separations from the output layer when we aggregate them all together.

Question 7

```
{
  tic()
ae4 <- h2o.deeplearning(x = predictors, # input layer
                        training_frame = anes_h2o, # which data frame we are using for this task
                        autoencoder = TRUE,
                        hidden = c(2, 2, 2), # number of hidden layers, and how many nodes in each.
                        epochs = 100,  # number of times to see the full input data
                        activation = "Tanh") # recall, non-linear activation
  toc()
}
```

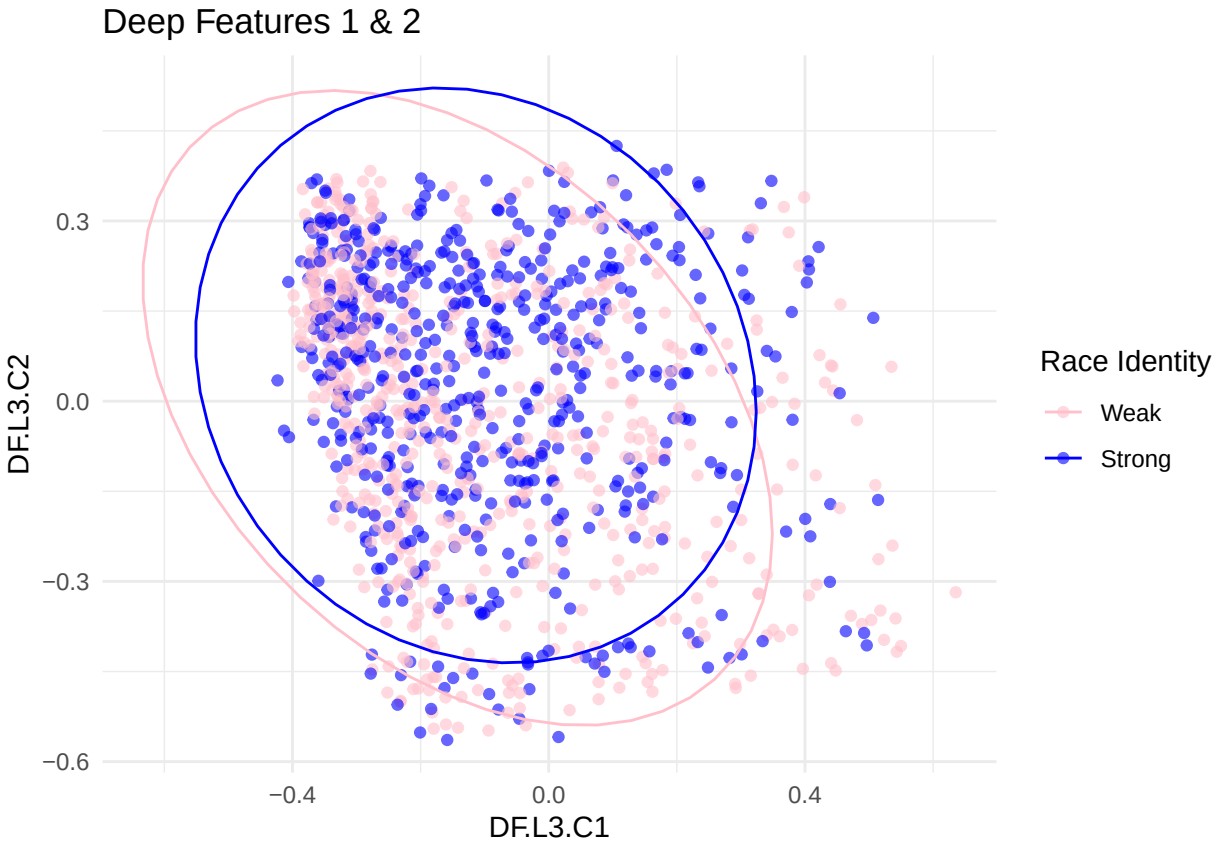
```
##   |                                                                      |
## 1.26 sec elapsed
```

```
# feature extraction
ae4_codings <- h2o.deepfeatures(ae4,
                                data = anes_h2o,
                                layer = 3) %>%
  as.data.frame() %>%
  mutate(strong_race_ident = as.vector(anes_h2o[ , 18])) # retain the 17th feature
```

```
##   |                                                                      |
```

Question 8

```
p4 <- ggplot(ae4_codings, aes(x = DF.L3.C1,
                              y = DF.L3.C2,
                              color = factor(strong_race_ident))) +
  geom_point(alpha = 0.6) +
  stat_ellipse() +
  scale_color_manual(values=c("pink", "blue"),
                     name="Race Identity",
                     breaks=c("0", "1"),
                     labels=c("Weak", "Strong")) +
  labs(title = "Deep Features 1 & 2",
       color = "Race Identity") +
  theme_minimal()
p4
```



From the graph, we can see that there is no obvious separation in the projection (question) space along senses of Race identity, since the blue and pink points are blended together without clear divisions. Compared to plot 3, I think the separation is not significantly clearer. In general, the more hidden layers, and thus the more processing that occurs to efficiently encode the input space and learn the lower-dimensional representation of the data, the deeper the AE. In this case, which we need a complex framework to efficiently learn, we still cannot observe any pattern in the deeper hidden layer. It strongly support that there is no pattern or structure in the projection (question) space along senses of Race identity. People's Race identity is not present in their responses to social questions. Meanwhile, we should be always aware that there is a trade-off in complexity. When we deepen the network, the model would be more unbiased but with more variance, and therfore we may have a better representation of the original data but it's easy to have overfitting problem with high computational costs.

Question 9

Compared to the patterns found from the SOM models we constructed last week, the patterns we find from today's AE models are the same - that is, there is no obvious patterns in people's responses to social questions against either American identity or Race identity. From the SOMs and AEs, we may conclude that there is no pattern or structure in the projection (question) space along senses of American/ Race identity. And people's American/ Race identities are not present in their responses to the 14 social questions.

Auto-encoders are a form of unsupervised learning algorithm that attempt to construct a condensed representation of their input. They are functionally equivalent to a feed-forward neural network with target values which are precisely its inputs, so that its loss function outputs the difference between the model output and the original input. By forcing the input to flow through a layer with a much smaller dimensionality than the input data, the activations of that layer can be considered a compressed representation of the original data.

Self-organizing Maps (SOM) are another form of unsupervised algorithm that attempts to perform dimensionality reduction of the input data. A SOM consists of multiple units, each unit has a vector of the same

dimension as the input and a coordinate in the map grid which is of much lower dimensionality, usually only 1 or 2. After training, the SOM can be used to discretize input samples. A given input vector can be replaced by the vector of that inputs' BMU. Although the replaced vector still has the same dimensionality as the original input, it can be stored with one number – the ID of the corresponding BMU. This means that the replaced input can be used in a neural network and still maintain the benefits of a high dimensional vector, while being able to be stored with one number, effectively compressing it.

The advantages of using SOM are that (1) the data is easily interpretted and understood; (2) SOMs are capable of handling several types of classification problems while providing a useful, interactive, and intelligable summary of the data; (3) SOMs are fully capable of clustering large, complex data sets. With a few optimization techniques, a SOM can be trained in a short amount of time. The disadvantages of using SOM are that (1) it requires necessary and sufficient data in order to develop meaningful clusters; (2) it is often difficult to obtain a perfect mapping where groupings are unique within the map; (3) SOMs require that nearby data points behave similarly. If we use the autoencoders, the advantages are that (1) It can learn non-linear transformations with a non-linear activation function and multiple layers; (2) It doesn't have to learn dense layers. It can use convolutional layers to learn which is better for video, image and series data; (3) It provides a representation of each layer as the output. The disadvantage is that as the complexity of the images increase, autoencoders struggle to keep up and images start to get blurry. We may waste a lot of time on something that adds more complexity than value to the end result. If we can combine the SOM and AE models, we may get better compression, by combining the benefits of dense and discrete representations.