Project 1
By Amna Khalid & Youssef Bourouphael

Dataset: NFL combine data (https://www.kaggle.com/savvastj/nfl-combine-data)
QIDs = Position (*Pos*), Forty Yard Dash Time (*Forty*), Round Drafted (*Round*)
K Values Used = 2, 4, 9

How We Defined Our QIDs and Why?

When we first saw the data we decided to break down each column and see if there was a valid option as a QID, however, if it was sensitive data, we marked it down so that we could drop it. After going through all the columns we were left with 13 columns that can qualify as a QID. This is where we discussed which would make the most sense to put together. We wanted "position" because we felt that in the game of football that is an important piece, so if this data were to be used it should be included. Then we decided that the players forty times should be included because speed is such a big factor in the sport. Lastly, we chose the round they were drafted because we felt it could be interesting to see the correlation between positions, their forty times, and when they were drafted.

Initially, to reach k-anonymity for the dataset, suppression of any PII, in this case it's the attributes that directly identifies an individual is carried out. Any other attribute was not considered a direct linker for finding PII thus only the 'Player' and 'Pfr_ID' column are suppressed to remove names and ids of NFL players. We also suppressed the year that they were drafted because using the pick, round, year, someone can infer who the data is about. This left the data with 13 unique attributes.

How We Measure Utility

Utility in this sense was how specific can this data be, such that we could see different positions and their speeds and clearly see where they were drafted. The more we felt a researcher could use this data and see a correlation between the three the more utility it had. Therefore the more specific each column was the more utility it had.

How This Utility Measure Was Impacted By Our K Values

K=2

We started with K=2 to keep things as specific as possible. This was the hardest because it's where we really had set ourselves up for the future Ks. The problem was that there were too many unique combinations (frequency of the quids ==1) that could be traced back and figured out. We really tried to keep the positions as they were, so the first attempt was to generalize forty times. These times were two decimal points long and therefore led to a lot of unique combinations. The first attempt was to just drop the second decimal, but that did not

bring down the unique values enough. To go even further we had to create ranges for these forty times (you can see them in the forty() function). Once these ranges were created, we had fewer unique combinations, but it was still about over 70. We then moved on to the rounds and decided we would have to make ranges for the rounds, since there were 9 options, it was just creating a lot of unique values. After we generalized it into 3 different ranges, we saw a decrease in unique values, but still, there remained over 50. So our final choice was to combine positions, since there are about 25 of them, it was definitely a big cause for these unique combinations. With a lot of trial and error, we decided the best way was to group positions together according to where they play on the field. An example of this looked like this:

```
OL = ["OL","G","C","OT","OG","LS","TE"] # Offensive Linemen
DL =["NT","DT","DE","EDGE"]             # Defensive Linemen
LM = OL + DL                            # Linemen
```

So anyone behind the line of scrimmage would be an Offensive Back, anyone in the backfield of defense was a Defensive Back, and so on. As soon as we did that, we were left with about 3 combinations that were still unique. After many more trials, trying to get rid of them, we thought it would be best to do some local generalization here. The basic problem was that there were outliers who were either “K/P” or “OB” and they were the only ones with the forty times they had. Instead of generalizing all the forty times even more I just locally generalized it for these two positions. Once we did that, we achieved anonymity for K = 2. In short, K=2, even though small, forced us to generalize all of our rows and even more after that. The utility definitely took a loss, but with the categories we used and the ranges we picked, this still is usable data. We lost the opportunity to see each specific position, forty, and round.

K = 2	Unique values BEFORE k-anonymizing	Unique values AFTER k-anonymizing
‘Pos’	25	4
‘Forty’	159	7
‘Round’	7	3
Total	191	14

K = 4

When we changed K to 4, we had about 15 combinations that did not meet this K value. We saw a lot of outliers within the “Undrafted” round and therefore had to combine “Undrafted” with rounds “4-8”. That took care of about half the outliers. Then there was an issue with the

“OB” local generalization, we had a “<4.00” and “4-4.49” from what we did in the last step, as outliers. To fix this we had to combine the two (locally) so that it would be “<4.5”. That took care of most but then we were left with an outlier in the “WR/DB” position that did not meet the K requirement because of their forty time. In order to fix that we combined all WR/DB speeds that are over “4.7” to just be “>4.7”. Thus it worked for K =2. The utility lost on this was greater than the K=2 because we generalized the rounds even more making it harder to see where people really fell in the draft. Also as we generalized the forty times for OBs and WR/DBs we lose good data because speed matters down to the 2nd decimal place so by creating even bigger ranges we lose more of the value of that data.

K = 4	Unique values BEFORE k-anonymizing	Unique values AFTER k-anonymizing
‘Pos’	25	4
‘Forty’	159	5
‘Round’	7	2
Total	191	11

K=9

Once we changed K to 9, we were hit with a lot more values that did meet the K requirements. We were forced to combine “OB” with “K/P” and “LM” to get rid of a good amount of the combinations that did not fit with K. We then had to locally generalize the forty times for those positions even more by making them all if it is under 4.5 to be <4.5. This is different from K=4 because that generalization was just for “OB”, now it is for “OB”, “K/P”, and “LM”. We also had to locally generalize all WR/DB speeds even more by combining the 4.4-4.59 and the 4.2-4.39 making it 4.2-4.59. This final K value really made the data even less usable. The utility is lost here because we had to generalize kickers, quarterbacks, and running backs (and more). These positions are not closely related and therefore the data on them is not that useful. Also when it comes to the WR/DB, these are the fastest guys on the field so when we create these big ranges especially between 4.2 and 4.59, it really loses the data because the data only matters when it’s close enough to the second decimal. Otherwise, a really fast guy who ran a 4.2 is with a less fast person who ran a 4.54.

K = 9	Unique values BEFORE k-anonymizing	Unique values AFTER k-anonymizing
‘Pos’	25	3

‘Forty’	159	3
‘Round’	7	2
Total	191	8