```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import missingno as msno
import scipy.stats as st
from sklearn import ensemble, tree, linear_model

#import hrdata.csv to hrdata dataframe
hrdata = pd.read_csv("/content/drive/MyDrive/Colab
Notebooks/hrdata.csv")

#count columns and rows
print("Count Columns and rows")
print("columns " + str(hrdata.shape[1]))
print("rows " + str(hrdata.shape[0]) + "\n")

#describe datafram
print("\nDescribe")
print(hrdata.describe())
```

Interpretation

The mean and std columns are interesting to note, for example in a catagory that only has 1 or 0, the lower the mean the more there are 0's than 1's. The mean is also helpful for showing the averages in a densly populated dataset

```
#top and bottom 5 rows
print("\nTop Five Rows")
print(hrdata.loc[[0,1,2,3,4]])
print("\nBottom Five Rows")
print(hrdata.iloc[[-1,-2,-3,-4,-5]])

#numerical columns
print("\nNumerical Columns")
print(hrdata.select_dtypes(include = [np.number]).columns)

#categorical columns
print("\nCategorical Columns")
print(hrdata.select_dtypes(include = [np.object]).columns)
```

```
Top Five Rows
   Unnamed: 0  rec_num  enrollee_id  ... target  state
city_development_matrics
0           0        1         8949  ...    1.0     CA
9.20
1           1        2        29725  ...    0.0     CA
```

```
7.76
2           2        3        11561  ...    0.0     CA
6.24
3           3        4        33241  ...    1.0     CA
7.89
4           4        5          666  ...    0.0     CA
7.67

[5 rows x 18 columns]

Bottom Five Rows
       Unnamed: 0  rec_num  enrollee_id  ... target  state
city_development_matrics
21286       21286    21287        12215  ...    NaN     CA
8.04
21285       21285    21286         7873  ...    NaN     CA
8.04
21284       21284    21285        31762  ...    NaN     CA
8.87
21283       21283    21284          195  ...    NaN     CA
8.97
21282       21282    21283         1289  ...    NaN     CA
9.20

[5 rows x 18 columns]

Numerical Columns
Index(['Unnamed: 0', 'rec_num', 'enrollee_id',
'city_development_index',
       'training_hours', 'target', 'city_development_matrics'],
      dtype='object')

Categorical Columns
Index(['city', 'gender', 'relevent_experience', 'enrolled_university',
       'education_level', 'major_discipline', 'experience',
'company_size',
       'company_type', 'last_new_job', 'state'],
      dtype='object')
```

```
#missing values
print("\nMissing Values Numerically")
missingValues = hrdata.isnull().sum().sort_values(ascending = False)
print(missingValues)

print("\nMissing Values Percentage")
print(missingValues/len(hrdata)*100)

#missing values displayed in bar plot
print("\nMissing Values plots")
plt.figure()
```

```
missingValues.sort_values(ascending = True).plot.bar()
plt.figure()
msno.bar(hrdata)
plt.figure()
msno.matrix(hrdata.sample(200))
plt.figure()
msno.heatmap(hrdata)
```

Interpretation

The heatmap shows that a majority of catagories have little to no correlation to the presence of null values except for company_type and company_size which show a 0.8 positive correlation.

The compared to the regular bar plot, the missingno bar plot is more effective in showing how many missing values are in each category. For example you can see that company_type has the most missing values with company_name following close behind

```
#plot for each object category
for catagoryName in hrdata[hrdata.select_dtypes(include = 
[np.object]).columns]:
  plt.figure()
  sns.countplot(x = catagoryName, data = hrdata)
  plt.show();

  plt.figure()
  sns.countplot(x = catagoryName, data = hrdata, hue = "target")
  plt.show();
```

Interpretations

The city plot shows that two paticular cities have way more than the other ones, coming in at first about 5000, then about 3000. Though it is difficult to tell which city unless you looked at the raw data or displayed it in another graph

The gender plot shows that there are significantly more males than females in the data set. For all genders the ratio compared to the target appear similar however you could not accurately make this assumation. Resampling may increase the accuracy of the dataset

The relevent experience plot shows that there are more of those with the relevent experience than those without

The enrolled university plot shows that there are significantly more not enrolled than there are enrolled full time and part time

The education level plot shows that a majority in the dataset are graduates

The major discipline plot shows that an extremely large majority are those with STEM disciplines. This information is not very useful since most data scientists would likely have STEM disciplines and while you could draw some conclusion there are so few of the other disiplinces so it makes it less accurate.

The experience graph shows that a majority of those in the dataset have more than 20 years of experience, while the rest show a relatively varied amount of years of experience with a flat 20 being the least common

The company size plot is relatively varied with a company size between 50-99 being the most common

The company type plot shows that the most common type of company is the Pvt LTD

The last new job plot shows that most have had one year since there last job, while the other years show a similar count

The state plot only shows that everyone is from california, this category could probably be removed since it doesn't really reveal any interesting or meaningful conclusions

```
#plot for each numerical category
for catagoryName in hrdata[hrdata.select_dtypes(include =
[np.number]).columns]:
  plt.figure()
  plt.ylabel("Count")
  plt.xlabel(catagoryName)
  plt.hist(x = catagoryName, data = hrdata)
  plt.show()

  plt.figure()
  sns.displot(x = catagoryName, data = hrdata, kde=False)
  plt.show()
```

Interpretations

The city_development_index and city_development_matrices plot the exact same data and are redundent. The unnamed, rec_num and enrollee_id plots are not beneficial data to graph as they dont really show any meaningful data and the target plot just visually shows the difference between those not looking and looking for a job change. The rest of the plots are useful in showing their relevent data, training hours shows that as hours increase count tends to decrease, both city_development plots show that as it increases count also tends to increase.

```
#plot for each numerical category
correlationNums = hrdata.select_dtypes(include = [np.number])
sns.heatmap(correlationNums.corr())
plt.figure()
plt.show()

columns = correlationNums.corr().nlargest(10,
"city_development_matrics")["city_development_matrics"].index
c = np.corrcoef(hrdata[columns].values.T)
plt.figure()
sns.heatmap(c, annot=True, xticklabels = columns.values, yticklabels =
columns.values)
```

```
plt.figure()
sns.scatterplot(x = "training_hours", y = "city_development_index",
data = hrdata)
plt.show()
```

Interpretation

Scatterplots may not be a very good way to show any correlations for this set of numerical data, the only scatterplot that might be useful is comparing training_hours with city_development_index or city_development_matrics since both city_developments show the same data. That scatterplot shows generally the higher the city_development the more likely someone would have more training hours. The next course of action might be to drop either the city_development_indexs or the city_development_matrics as they hold the same information. Also to possibly only keep one of either: unnammed or rec_num as rec_num is the same as unnammed plus 1. There also isn't really a point to graphing enrollee_id since they are just employee numbers.

```
for catagoryName in hrdata[hrdata.select_dtypes(include =
[np.number]).columns]:
  plt.figure()
  sns.boxplot(x = catagoryName, data = hrdata)
  plt.show()
```

The different values of experience include less than 1 (<1), 1-20 and greater than 20 (>20). You could group the different values into groups 0, 1 and 2 in order to reveal some sort of trend or correlation, however there may be better ways about it.

Some catagories may be very skewed in certain directions and require some form of rebalancing, for example a huge portion of people are male compared to female and almost all are of a STEM discipline. Also those with more than 20% data missing might need to be removed as it may not be entirely reliable data. Next course of action would probably be to remove one of the city_developments as they show the exact same data and are redundent. I would also remove unnammed as it is an exact copy of rec_num however it begins counting at 0 instead of 1. The state catagory is also not useful as everyone is from california. Enrollee id is not useful plottable data however it could be important for identifying employee information.