Recommendations to be Used in Mitigating the Hazards

After viewing the tasks and the hazards experienced when executing the tasks, we came up with recommendations that are meant to mitigate the hazards. In this report we are going to analyze and see the different recommendations and assess them on different aspects.

In [1]:
```
# Importing the necessary libraries to be used in our analysis
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Setting the data columns to be displayed wholly
pd.set_option('display.max_columns',500)
```

In [3]:
```
# reading the first dataset
df1 = pd.read_excel('Mitigations.xlsx')
```

In [4]:
```
# reading the second dataset
df2 = pd.read_excel('Recommendations.xlsx')
```

In the first dataset we had empty cells which meant that there we zero hazards mitigated by the recommendation under that specific hazard. We therefore had to make the dataset to reflect this by filling all the empty cells.

In [5]:
```
# filling the empty cells with 0
df1.fillna(0,inplace = True)
```

In [6]:
```
# redefining the index of the dataframe
df1.index = df1['Unnamed: 0']
```

In [7]:
```
# droping the column as it is now the index
df1.drop(columns = 'Unnamed: 0',inplace = True)
```

In [8]:
```
# Transposing the dataset for better interpretation
df1 = df1.T
# Removing the index title
df1 = df1.rename_axis(None,axis = 1)
# Dropped the 'sum' row as we had no use for it in our analysis
df1.drop(['Sum'],inplace = True)
df1
```

Out[8]:

	Pre-existing hazards	Hazards Eliminate by SHORT TERM Recommendations	Hazards Eliminate by MEDIUM TERM Recommendations	Hazards Eliminate by LONG TERM Recommendations	Residual hazards remaining
Difficult Access	122.0	13.5	5.0	65.0	38.5
Awkward Posture / Ergonomics	114.0	11.0	15.0	47.0	41.0
Vibration	58.0	2.0	0.0	0.0	56.0
Repetitive	37.0	0.0	0.0	0.0	37.0
Sustained	28.0	0.0	0.0	3.5	24.5
Manual Handing	60.0	10.5	4.5	11.5	33.5
Working at Heights	9.0	0.0	0.0	4.5	4.5
Working underneath rollingstock	37.0	0.0	0.0	17.0	20.0
Stored energy	20.0	0.0	0.0	0.0	20.0
Lifting/movement of heavy components	36.0	0.0	0.0	27.5	8.5
Inadequate procedures	56.0	1.0	0.0	0.0	55.0
Chemicals	31.0	5.0	4.0	13.0	9.0
Moveable vehicles	31.0	0.0	0.0	22.0	9.0
Noisy Conditions	68.0	6.0	0.0	0.0	62.0
Potential exposure to biohazards	38.0	0.0	0.0	19.0	19.0
No Hazards Identified	80.0	0.0	0.0	0.0	80.0

In [9]:

```
# made all the values to float dtype
df1 = df1.astype(float)
```

In [10]:

```
# to get the descriptive statistics of the data
df1[0:-1].describe()
```

Out[10]:

	Pre-existing hazards	Hazards Eliminate by SHORT TERM Recommendations	Hazards Eliminate by MEDIUM TERM Recommendations	Hazards Eliminate by LONG TERM Recommendations	Residual hazards remaining
count	15.000000	15.000000	15.000000	15.000000	15.000000
mean	49.666667	3.266667	1.900000	15.333333	29.166667
std	31.923271	4.776679	4.071679	19.144812	18.736583
min	9.000000	0.000000	0.000000	0.000000	4.500000

	Pre-existing hazards	Hazards Eliminate by SHORT TERM Recommendations	Hazards Eliminate by MEDIUM TERM Recommendations	Hazards Eliminate by LONG TERM Recommendations	Residual hazards remaining
25%	31.000000	0.000000	0.000000	0.000000	14.000000
50%	37.000000	0.000000	0.000000	11.500000	24.500000
75%	59.000000	5.500000	2.000000	20.500000	39.750000
max	122.000000	13.500000	15.000000	65.000000	62.000000

From the descriptive statistics shown above it is seen that there are 15 hazards, where currently there is an average of 50 occurrences of each hazard, with a standard deviation of 32. Short-term recommendations reduce an average of 3 occurrences of each hazard with a standard deviation of 5. Medium-recommendations reduce an average of 2 occurrences of each hazard and lastly the Long-term recommendations reduce an average of 15 occurrences of each hazard. If all the recommendations are implemented, the average number of occurrences per hazard will be reduced to 29 from 50.

In [11]:

```
# to get an understanding of the data
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 16 entries, Difficult Access  to No Hazards Identified
Data columns (total 5 columns):
 #   Column                                          Non-Null Count  Dtype  
---  ------                                          --------------  -----  
 0   Pre-existing hazards                            16 non-null     float64
 1   Hazards Eliminate by SHORT TERM Recommendations  16 non-null     float64
 2   Hazards Eliminate by MEDIUM TERM Recommendations 16 non-null     float64
 3   Hazards Eliminate by LONG TERM Recommendations   16 non-null     float64
 4   Residual hazards remaining                      16 non-null     float64
dtypes: float64(5)
memory usage: 768.0+ bytes
```

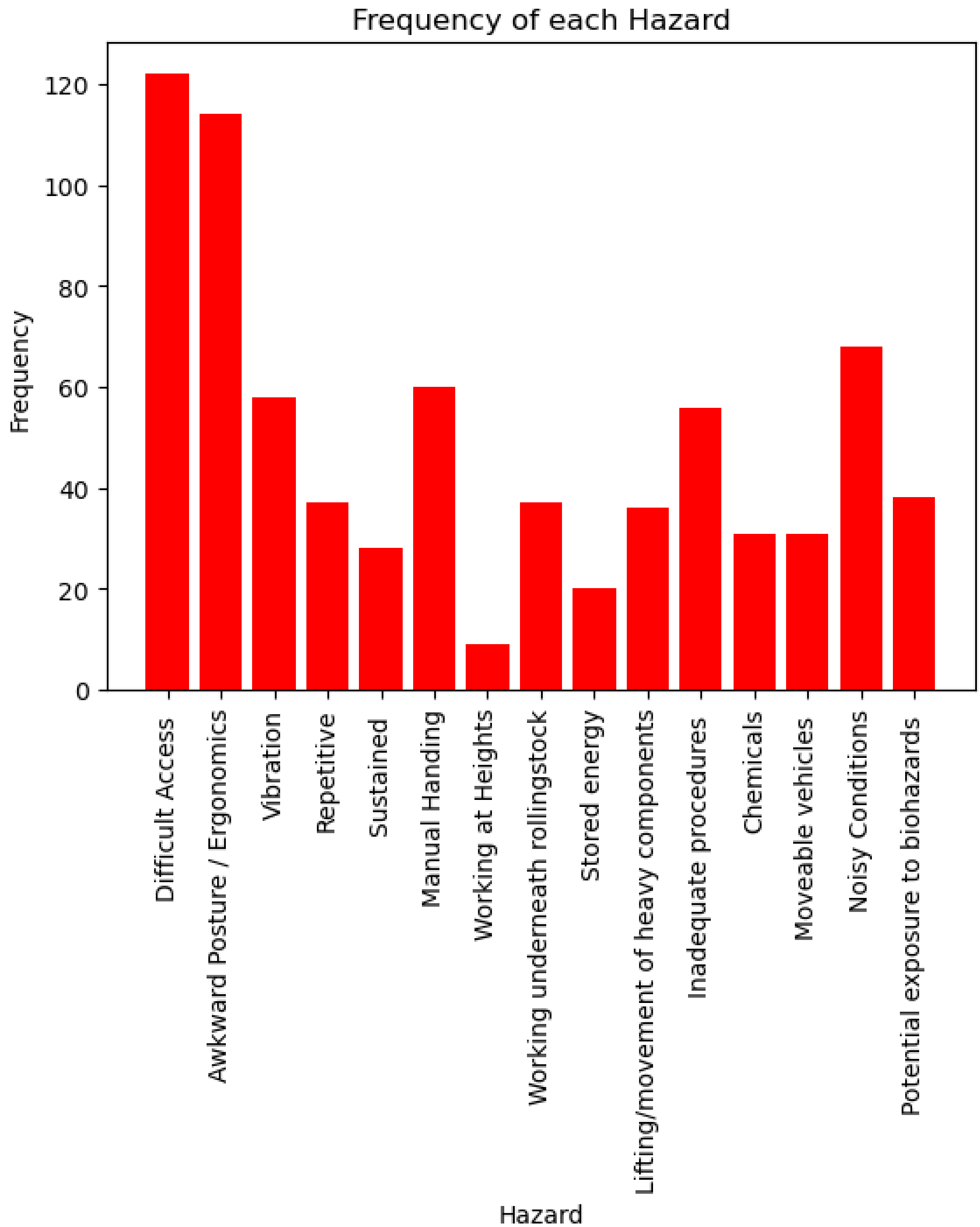
In [12]:

```
# renaming the columns for easier analysis
df1.rename(columns = {'Hazards Eliminate by SHORT TERM Recommendations':'Short_term',\
             'Hazards Eliminate by MEDIUM TERM Recommendations':'Medium_term',\
              'Hazards Eliminate by LONG TERM Recommendations':'Long_term',\
              'Pre-existing hazards':'Frequency_of_Hazard'},inplace = True)
```

The plot below shows the distribution of hazards, and it is clear 'Difficult Access' has the highest frequency while 'working at heights' has the least

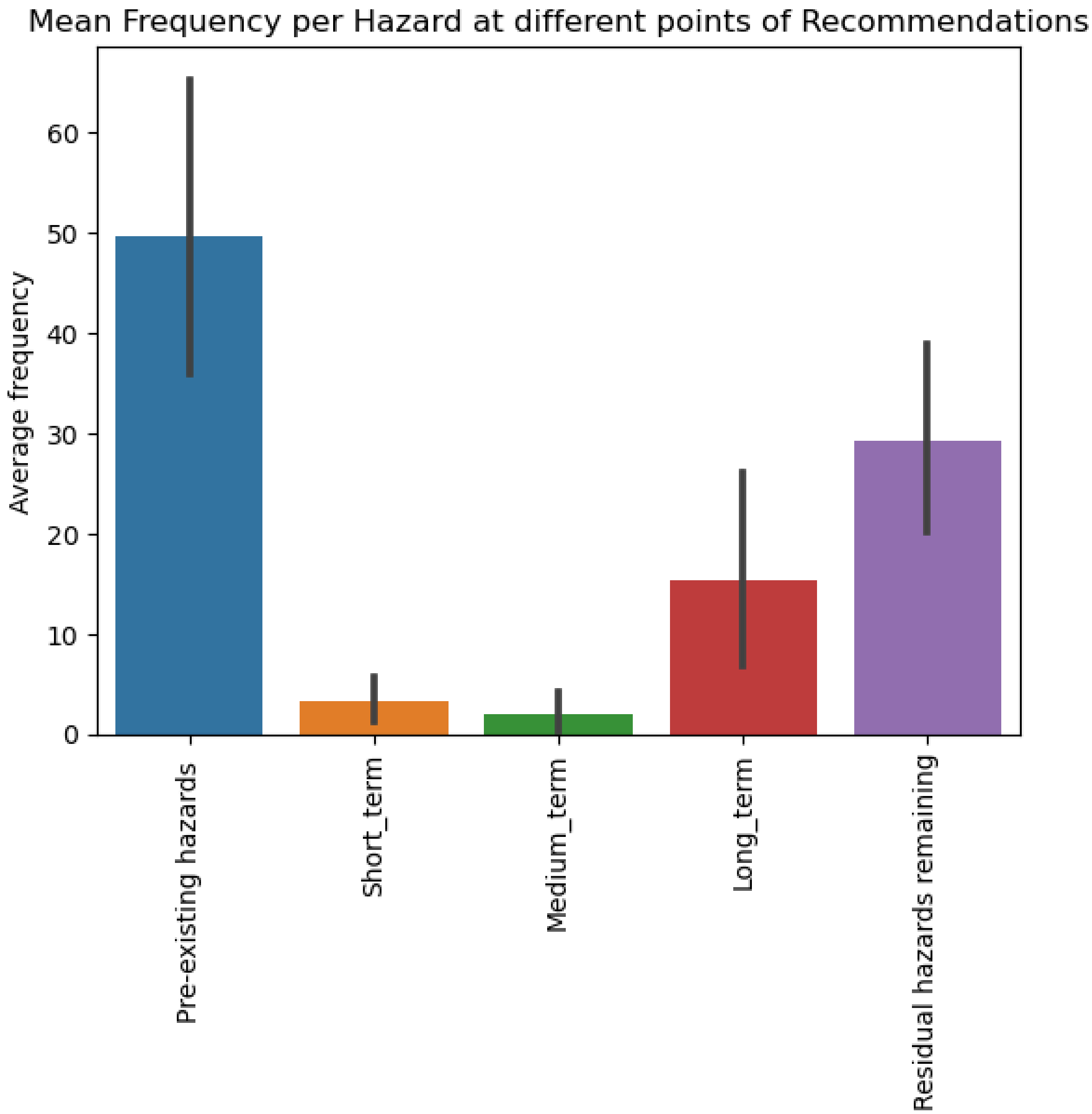
In [13]:

```
# plotting the distribution of hazards
plt.title('Frequency of each Hazard')
plt.bar(x = df1[0:-1].index,height = df1.iloc[0:-1,0], color = 'r')
plt.ylabel('Frequency')
plt.xlabel('Hazard')
plt.xticks(rotation = 90);
```



In [14]:

```
# plotting to see the mean at different levels
sns.barplot(df1[0:-1])
plt.title('Mean Frequency per Hazard at different points of Recommendations')
plt.ylabel('Average frequency')
plt.xticks(rotation = 90);
```

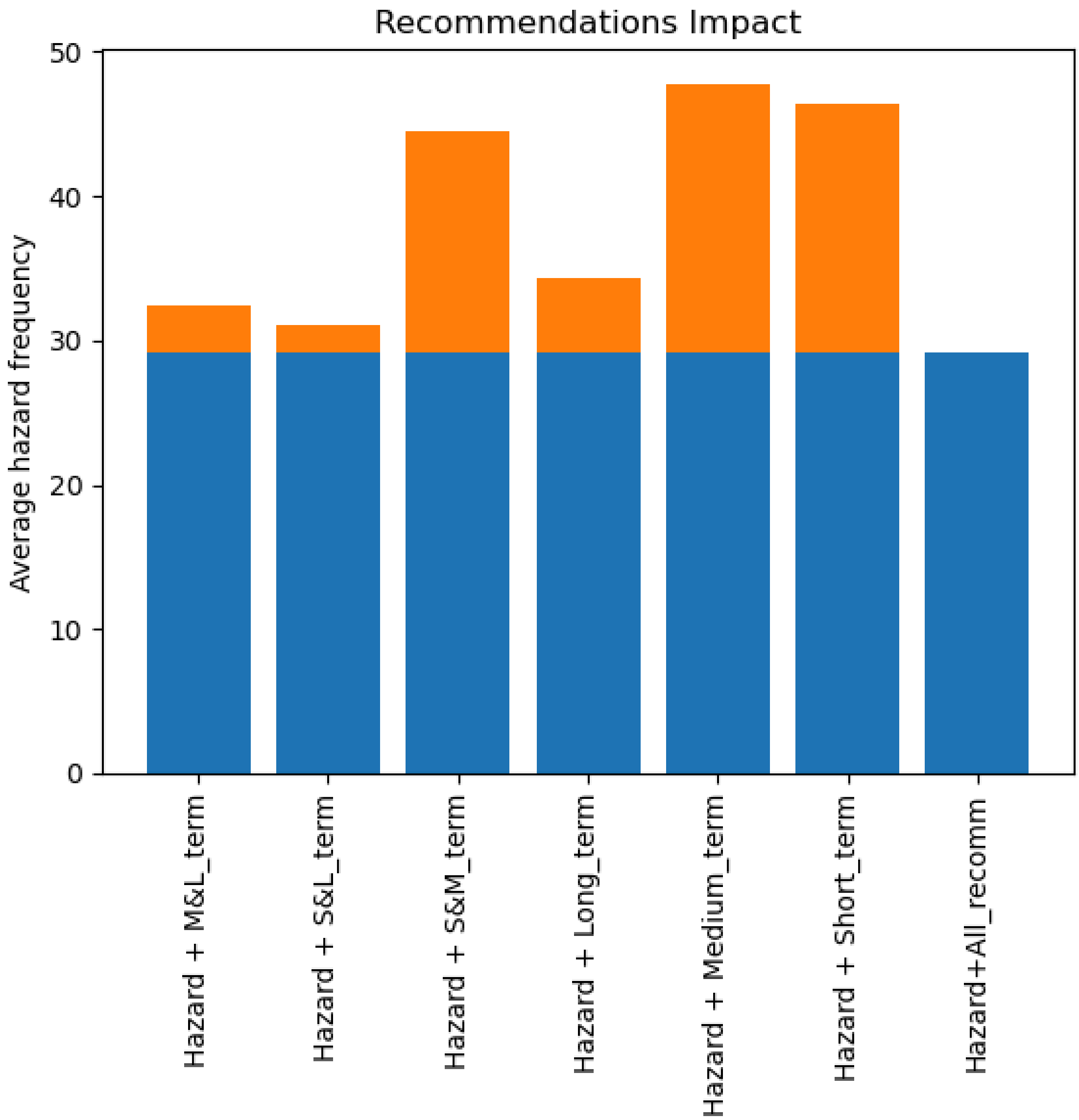


As seen on the descriptive statistics, different recommendations have different impacts. We have to visualize the effect of implementing particular recommendations or a combination of recommendations.

In [15]:

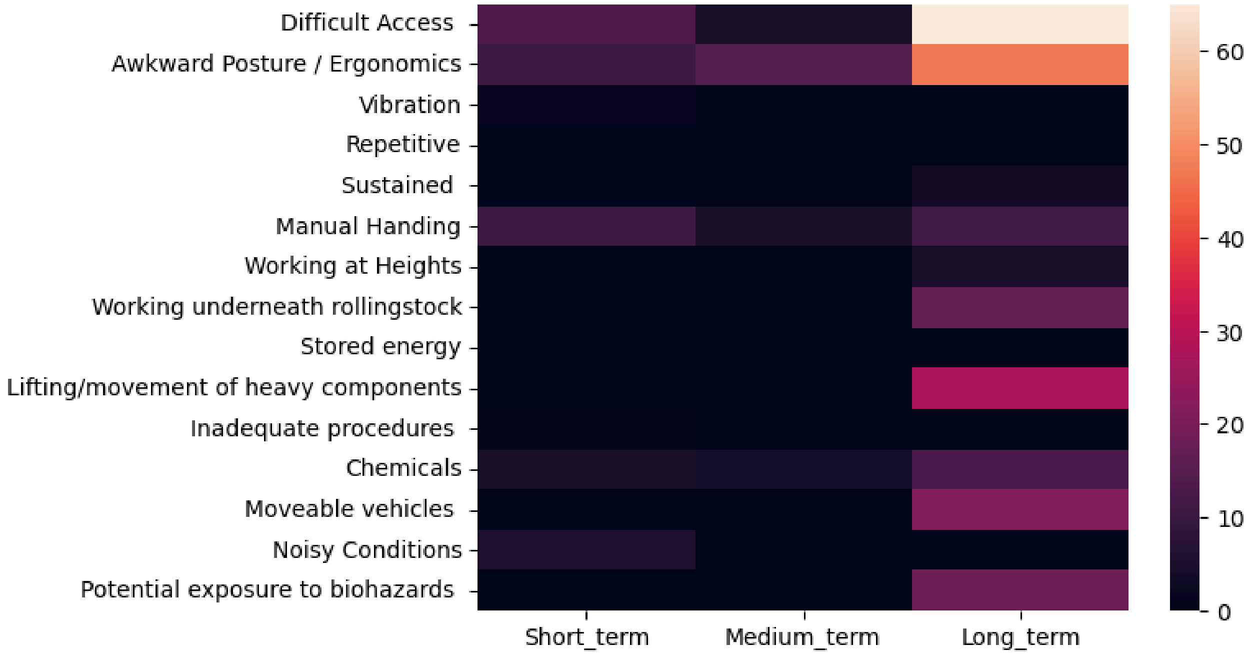
```
# plotting the effect of recommendations
x = ['Hazard + M&L_term','Hazard + S&L_term','Hazard + S&M_term',\
     'Hazard + Long_term','Hazard + Medium_term','Hazard +
Short_term','Hazard+All_recomm']
y1 = []
for i in range(7):
    y1.append(df1[0:-1]['Residual hazards remaining'].mean())
y2 = [df1[0:-1]['Short_term'].mean(),df1[0:-1]['Medium_term'].mean()\
      ,df1[0:-1]['Long_term'].mean(),\
      df1[0:-1]['Short_term'].mean()+df1[0:-1]['Medium_term'].mean(),\
      df1[0:-1]['Short_term'].mean()+df1[0:-1]['Long_term'].mean(),\
      df1[0:-1]['Medium_term'].mean()+df1[0:-1]['Long_term'].mean(),0]
plt.bar(x,y1)
plt.bar(x,y2,bottom = y1)
plt.xticks(rotation = 90)
```

```
plt.title("Recommendations Impact")
plt.ylabel("Average hazard frequency");
```



From the above visualization it is seen that the implementation of the medium_term recommenadtions only has the least effect while implementing all has the highest effect. Implementing a combination of Short_term and Long_term recommendations has the second most impact.

In [16]:
```
# heatmap showing how each recommendation type affects a hazard
sns.heatmap(df1[0:-1][['Short_term','Medium_term','Long_term']]);
```



The heatmap shows that Long_term and short term recommendations affect Difficult access the most, while medium term recommendations affect Ergonomics the most

In [17]:

```
# dropping unnecessary columns
df2=df2[['ID','Recommendation','Hazard_removed','Number_of_hazards','Term'
```

In [18]:

```
# dropping null values along the number of hazards column
df2=df2.dropna(subset = ['Number_of_hazards'])
# fill the empty cells with relevant entries
df2.ffill(inplace=True)
df2.head()
```

Out[18]:

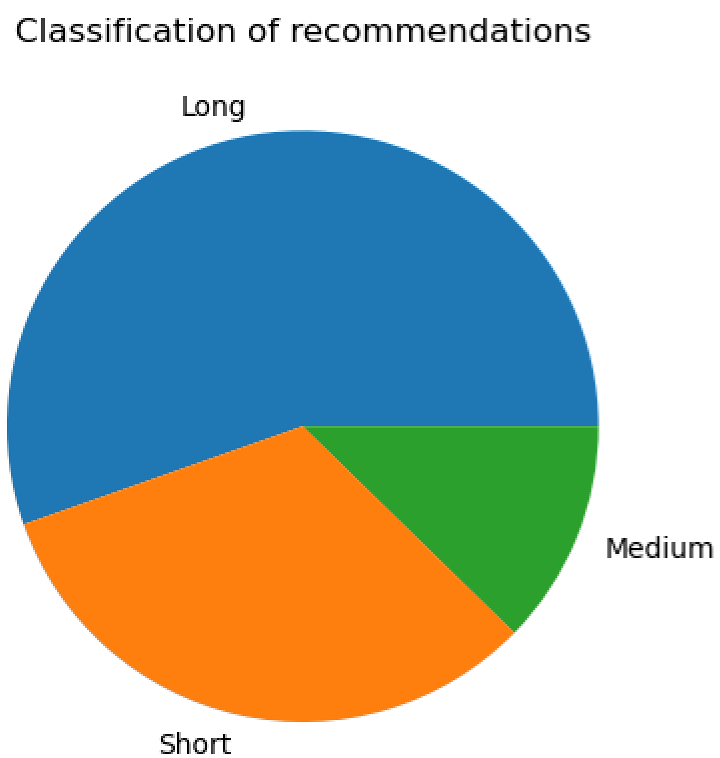
	ID	Recommendation	Hazard_removed	Number_of_hazards	Term
0	HF-REC1	Modification to engine bay kick plate and re-r...	Difficult Access	15.0	Long
1	HF-REC1	Modification to engine bay kick plate and re-r...	Awkward Posture / Ergonomics	18.0	Long
2	HF-REC1	Modification to engine bay kick plate and re-r...	Sustained	3.5	Long
20	HF-REC4	Purpose built (moveable) step/platform for foo...	Difficult Access	10.0	Short
21	HF-REC4	Purpose built (moveable) step/platform for foo...	Awkward Posture / Ergonomics	5.0	Short

In [19]:

```
# showing the number of recommendations under each classification
df2['Term'] = df2['Term'].str.strip()
Term_dist = df2['Term'].value_counts()
Term_dist
```

Out[19]:
```
Term
Long      36
Short     21
Medium     8
Name: count, dtype: int64
```

In [20]:
```
# plotting the distribution of recommendations
plt.pie(Term_dist,labels = Term_dist.index)
plt.title('Classification of recommendations');
```



In [21]:
```
# showing the number of hazards mitigated by each recommendation
Recom = pd.DataFrame(df2.groupby(['ID','Term'])
['Number_of_hazards'].sum().sort_values(ascending = False))
Recom
```

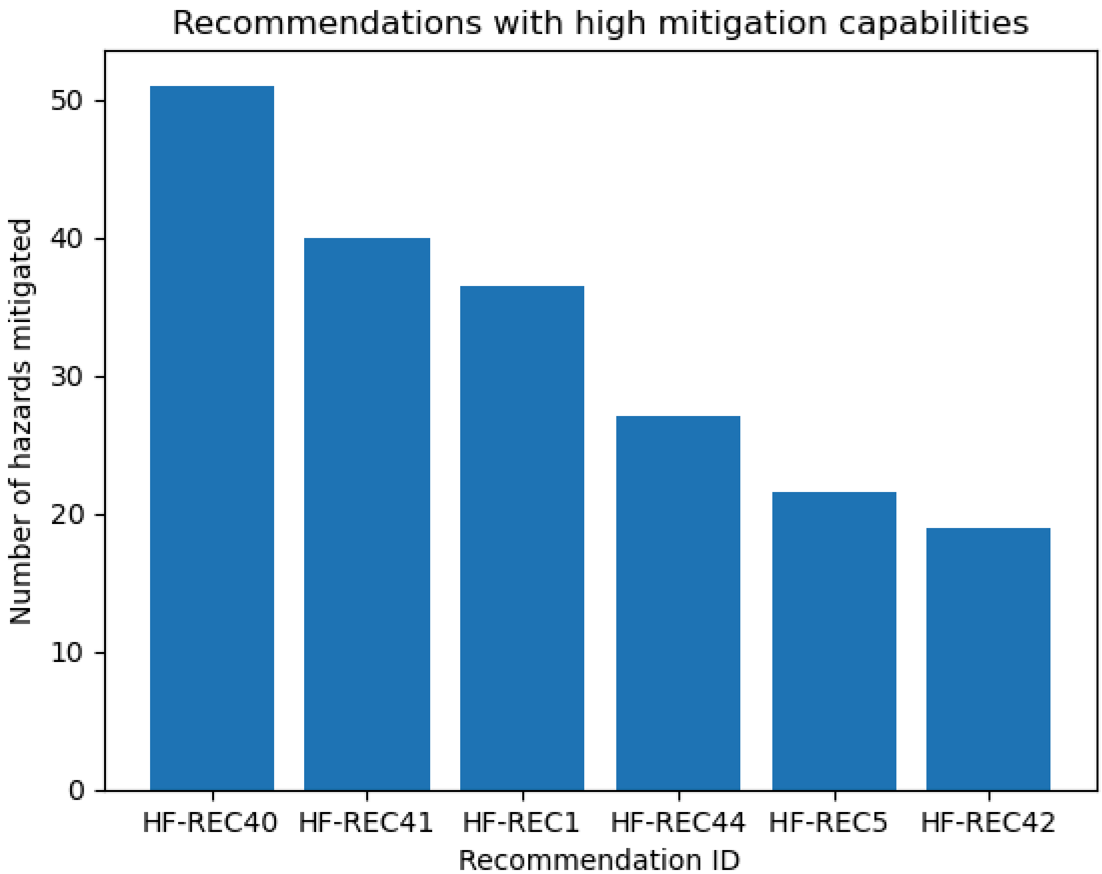
Out[21]:

ID	Term	Number_of_hazards
HF-REC40	Long	51.0
HF-REC41	Long	40.0
HF-REC1	Long	36.5
HF-REC44	Long	27.0
HF-REC5	Medium	21.5
HF-REC42	Long	19.0
HF-REC4	Short	15.0
HF-REC39	Long	12.0

ID	Term	Number_of_hazards
HF-REC21	Long	12.0
HF-REC13	Short	9.5
HF-REC30	Long	9.0
HF-REC43	Long	8.0
HF-REC31	Short	5.0
HF-REC48	Long	5.0
HF-REC12	Short	5.0
HF-REC28	Medium	4.5
HF-REC46	Long	2.5
HF-REC23	Long	2.5
HF-REC36	Short	2.0
HF-REC47	Long	2.0
HF-REC33	Short	2.0
HF-REC29	Short	2.0
HF-REC27	Long	2.0
HF-REC25	Short	2.0
HF-REC20	Short	2.0
HF-REC16	Long	2.0
HF-REC9	Short	2.0
HF-REC35	Short	1.5
HF-REC34	Medium	1.5
HF-REC38	Long	1.0
HF-REC32	Short	1.0
HF-REC24	Long	1.0
HF-REC18	Medium	1.0
HF-REC26	Long	0.5

In [22]:

```
#A distribution of the top recommendations
plt.bar(x=[x for x,y in Recom[:6].index],height = Recom[:6]
['Number_of_hazards'])
plt.title('Recommendations with high mitigation capabilities')
plt.xlabel('Recommendation ID')
plt.ylabel('Number of hazards mitigated');
```



HF-REC40 which is a recommendation that dictates maintenance pits shall be incorporated into the design to support safe access to the underneath of locomotives. This has the greatest impact and could be prioritized. Most of the high impact recommenadtions are long term as anticipated by the long term recommendations impact.