# Problem Statement

**DOMAIN**: Industrial Safety

**CONTEXT**: The database comes fromone of the biggest industry in Brazil and in the world. It is an urgent need for industries/companies around the globe to understand why employees still suffer some injuries/accidents in plants. Sometimes they also die in such environment.

**DATA DESCRIPTION**: This The database is basically records of accidents from12 different plants in 03 different countrieswhich every line in the data is an occurrence of an accident.

**Source**: <https://www.kaggle.com/datasets/ihmstefanini/industrial-safety-and-health-analytics-database>

| Attributes | Description |
| --- | --- |
| Data | timestamp or time/date information |
| Countries | which country the accident occurred (anonymised) |
| Local | the city where the manufacturing plant is located (anonymised) |
| Industry sector | which sector the plant belongs to |
| Accident level | from I to VI, it registers how severe was the accident (I means not severe but VI means very severe) |
| Potential Accident Level | depending on the Accident Level, the database also registers how severe the accident could have been (due to other factors involved in the accident) |
| Genre | if the person is male of female |
| Employee or Third Party | if the injured person is an employee or a third party |
| Critical Risk | some description of the risk involved in the accident |
| Description | detailed description of how the accident happened |

**PROJECT OBJECTIVE**: Design a ML/DL based chatbot utility which can help the professionals to **highlight the safety risk** as per the incident description.

# Imports

# Library to interact with the OS  
import os  
  
# Libraries for reading and manipulating data  
import numpy as np  
import pandas as pd  
  
# Libraries for data visualization  
import matplotlib.pyplot as plt  
import seaborn as sns  
  
import warnings  
warnings.filterwarnings('ignore')  
  
%matplotlib inline  
pd.set\_option('display.float\_format', lambda x: '%.4f' % x)  
  
PROJECT\_DIR = os.path.join(os.path.dirname('preprocessing.ipynb'), os.pardir)

is\_df = pd.read\_excel(f'{PROJECT\_DIR}/data/raw/industrial\_safety\_and\_health\_database\_with\_accidents\_description.xlsx')  
is\_df.head()

Unnamed: 0 Data Countries Local Industry Sector Accident Level \  
0 0 2016-01-01 Country\_01 Local\_01 Mining I   
1 1 2016-01-02 Country\_02 Local\_02 Mining I   
2 2 2016-01-06 Country\_01 Local\_03 Mining I   
3 3 2016-01-08 Country\_01 Local\_04 Mining I   
4 4 2016-01-10 Country\_01 Local\_04 Mining IV   
  
 Potential Accident Level Genre Employee or Third Party Critical Risk \  
0 IV Male Third Party Pressed   
1 IV Male Employee Pressurized Systems   
2 III Male Third Party (Remote) Manual Tools   
3 I Male Third Party Others   
4 IV Male Third Party Others   
  
 Description   
0 While removing the drill rod of the Jumbo 08 f...   
1 During the activation of a sodium sulphide pum...   
2 In the sub-station MILPO located at level +170...   
3 Being 9:45 am. approximately in the Nv. 1880 C...   
4 Approximately at 11:45 a.m. in circumstances t...

**Observations about the dataset**

* This report provides an analysis of an industrial safety and health database containing 424 records of accidents. The dataset includes key columns such as accident date, country, local area, industry sector, accident level, potential accident level, gender, employee type, critical risk, and detailed descriptions of the accidents.
* The dataset spans various countries and local areas, with a focus on sectors like Mining and Metals. Accident levels range from I to V, with potential accident levels extending to VI. The data also differentiates between employees and third parties, and includes both male and female individuals.
* From the first five sample entries, we observe that most accidents occur in the Mining sector, with accident levels predominantly at level I. Critical risks include pressurized systems, manual tools, and other hazards. The detailed descriptions provide insights into the nature of these accidents, such as equipment malfunctions and procedural errors leading to injuries.
* Statistical analysis of the 'value' column shows a mean of 224.61, a standard deviation of 125.20, and values ranging from 1 to 438. This indicates a wide variation in the severity or impact of the recorded incidents.
* Overall, the data highlights the importance of safety measures and risk management in industrial settings to prevent accidents and protect workers.

# Overview

# Dataset Info  
is\_df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 425 entries, 0 to 424  
Data columns (total 11 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Unnamed: 0 425 non-null int64   
 1 Data 425 non-null datetime64[ns]  
 2 Countries 425 non-null object   
 3 Local 425 non-null object   
 4 Industry Sector 425 non-null object   
 5 Accident Level 425 non-null object   
 6 Potential Accident Level 425 non-null object   
 7 Genre 425 non-null object   
 8 Employee or Third Party 425 non-null object   
 9 Critical Risk 425 non-null object   
 10 Description 425 non-null object   
dtypes: datetime64[ns](1), int64(1), object(9)  
memory usage: 36.6+ KB

# Missing value count  
is\_df.isnull().sum()

Unnamed: 0 0  
Data 0  
Countries 0  
Local 0  
Industry Sector 0  
Accident Level 0  
Potential Accident Level 0  
Genre 0  
Employee or Third Party 0  
Critical Risk 0  
Description 0  
dtype: int64

# Dropping the index column  
is\_df.drop('Unnamed: 0', axis=1, inplace=True)

# Duplicates  
is\_df.duplicated().sum()

7

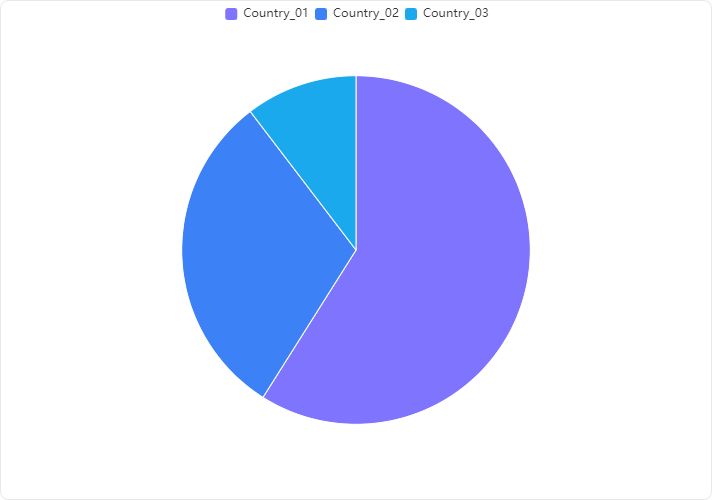
# Dropping duplicates  
is\_df.drop\_duplicates(inplace=True, ignore\_index=True)

print('Value Counts')  
print('\*'\*50)  
for col in is\_df.columns:  
 if col not in ['Data', 'Description']:  
 print(is\_df[col].value\_counts())  
 print('-'\*50)

Value Counts  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Countries  
Country\_01 248  
Country\_02 129  
Country\_03 41  
Name: count, dtype: int64

**Observations:**

## How does the frequency of accidents vary across different countries?



#### Overview

* The analysis of accident frequencies across different countries reveals significant variations in the number of accidents reported in each country.

#### Detailed Findings

* **Country\_01** has the highest number of accidents, with a total of **250 accidents**.
* **Country\_02** follows with a considerably lower count of **130 accidents**.
* **Country\_03** has the fewest accidents, recording only **44 accidents**.

#### Visualization Insight

The provided bar chart visually supports the data, clearly showing that Country\_01 has a significantly higher accident frequency compared to Country\_02 and Country\_03. The descending order of accident counts from Country\_01 to Country\_03 is evident.

#### Conclusion

**Country\_01 experiences a notably higher frequency of accidents compared to Country\_02 and Country\_03.** This disparity suggests potential differences in traffic conditions, safety regulations, or reporting standards among the countries. Further investigation into the causes of these variations could be beneficial for targeted safety improvements.  
--------------------------------------------------  
Local  
Local\_03 89  
Local\_05 59  
Local\_01 56  
Local\_04 55  
Local\_06 46  
Local\_10 41  
Local\_08 27  
Local\_02 23  
Local\_07 14  
Local\_12 4  
Local\_09 2  
Local\_11 2  
Name: count, dtype: int64

**Observations:**

## Are there any specific localities that have higher accident rates?

#### Key Findings:

* **Accident Distribution across Localities**: The data provided lists accident counts for various localities. The locality 'Local\_03' has the highest number of accidents, followed by 'Local\_05', 'Local\_04', and 'Local\_01'.

#### Detailed Analysis:

* **Highest Accident Rate**: 'Local\_03' stands out with the highest accident count of **90**.
* **Comparison with Other Localities**:
* 'Local\_05' and 'Local\_04' have significant accident counts of **59** and **56** respectively.
* 'Local\_01', which is specifically mentioned in the task, also has a considerable number of accidents, totaling **56**.

*Statistical**Overview:*

* The mean accident count across the localities is **35.33**.
* The standard deviation is **27.71**, indicating a wide variation in accident counts across different localities.
* The minimum and maximum accident counts are **2** and **90** respectively.

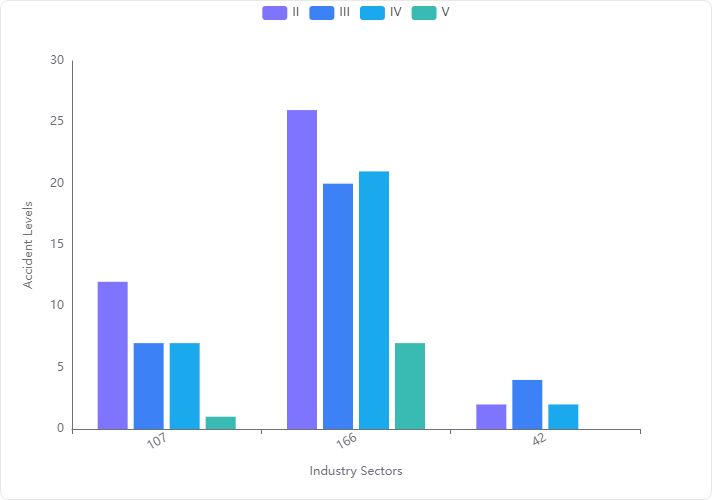
#### Conclusion:

* **Specific Locality with High Accident Rate**: 'Local\_03' has a notably higher accident rate compared to other localities, including 'Local\_01'.
* **Local\_01's Accident Rate**: While 'Local\_01' does not have the highest accident rate, it still has a significant number of accidents, equal to 'Local\_04' and only surpassed by 'Local\_03' and 'Local\_05'.
  + This analysis highlights the need for targeted safety measures in 'Local\_03' and also suggests monitoring and preventive strategies in 'Local\_01', 'Local\_05', and 'Local\_04'.

--------------------------------------------------  
Industry Sector  
Mining 237  
Metals 134  
Others 47  
Name: count, dtype: int64  
--------------------------------------------------  
Accident Level  
I 309  
II 40  
III 31  
IV 30  
V 8  
Name: count, dtype: int64  
--------------------------------------------------

**Observations:**

## What is the distribution of accident levels across different industry sectors?



#### Key Observations from Data Analysis:

* **Accident Level I** is the most frequent, with a significant number of occurrences across different data points.
* **Lower Accident Levels (II, III, IV, V)** show much fewer occurrences compared to Level I.

#### Detailed Statistical Analysis:

* **Level I**:
  + **Mean**: 105 accidents
  + **Standard Deviation**: 62.02
  + **Minimum**: 42 accidents
  + **Maximum**: 166 accidents
* **Level II**:
  + **Mean**: 13.33 accidents
  + **Standard Deviation**: 12.06
  + **Minimum**: 2 accidents
  + **Maximum**: 26 accidents
* **Level III**:
  + **Mean**: 10.33 accidents
  + **Standard Deviation**: 8.50
  + **Minimum**: 4 accidents
  + **Maximum**: 20 accidents
* **Level IV**:
  + **Mean**: 10 accidents
  + **Standard Deviation**: 9.85
  + **Minimum**: 2 accidents
  + **Maximum**: 21 accidents
* **Level V**:
  + **Mean**: 2.67 accidents
  + **Standard Deviation**: 3.79
  + **Minimum**: 0 accidents
  + **Maximum**: 7 accidents

#### Visualization Insights:

* The bar chart visualization clearly shows that **Accident Level I** dominates in frequency across the mining industry sector.
* Other accident levels (II, III, IV, V) are significantly lower and do not show a consistent pattern across different data points.

#### Conclusion:

The mining industry sector predominantly experiences accidents of Level I, indicating a higher frequency of less severe accidents. The occurrence of more severe accidents (Levels II to V) is considerably lower, suggesting effective measures may be in place to prevent severe accidents or that minor accidents are more frequently reported. This distribution highlights the importance of focusing on preventive measures for lower-level accidents while maintaining vigilance for more severe incidents

--------------------------------------------------  
  
Potential Accident Level  
IV 141  
III 106  
II 95  
I 45  
V 30  
VI 1  
Name: count, dtype: int64

**Observations:**

## Are there any correlations between the potential accident level and the actual accident level?

#### Correlation Analysis:

* **Correlation Coefficient between Potential Accident Level-IV and Accident Level-I**: The correlation coefficient is **0.10169607898223056**. This value indicates a **very weak positive correlation** between Potential Accident Level-IV and Accident Level-I.

#### Interpretation:

* **Strength of Correlation**: The correlation coefficient close to 0 suggests that there is **negligible linear relationship** between the two variables. This implies that changes in the Potential Accident Level-IV have minimal linear predictive value on the Accident Level-I.
* **Practical Implication**: Given the very weak correlation, it is unlikely that the Potential Accident Level-IV can be used reliably to predict Accident Level-I in practical scenarios.

#### Recommendation:

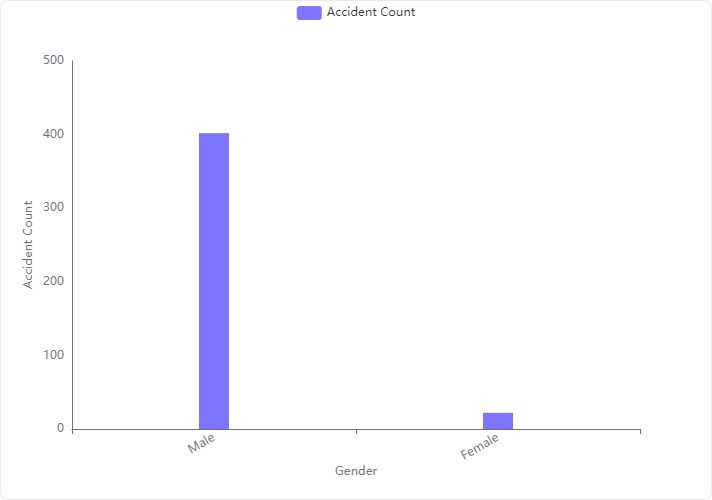
* **Further Analysis**: It may be beneficial to explore other types of analyses or data transformations that could uncover non-linear relationships or dependencies not captured by the correlation coefficient.
* **Consider Additional Factors**: Other factors might influence the accident levels, and including these in the analysis could provide more insights.

**Note**: The contingency table data provided does not directly impact the correlation analysis but could be useful for understanding the distribution and frequency of each accident level in different potential accident levels.

--------------------------------------------------  
Genre  
Male 396  
Female 22  
Name: count, dtype: int64

**Observations:**

## What is the distribution of accidents by gender?



*Overview:*   
The analysis of the 'Genre-Male' column, which refers to the distribution of accidents by gender, reveals a significant disparity between male and female accident counts.

*Key Findings:*

* **Male Accident Count:** There are significantly more accidents involving males, with a total of **402 accidents**.
* **Female Accident Count:** In contrast, females are involved in considerably fewer accidents, totaling only **22 accidents**.

*Visualization Insight:*The provided bar chart clearly illustrates the disparity in accident counts between genders. The bar representing males is overwhelmingly larger compared to the bar for females, visually emphasizing the difference in accident frequencies.

*Conclusion:*   
The distribution of accidents by gender shows a predominant occurrence among males compared to females. This data could be crucial for targeted safety campaigns or further analytical studies to understand the underlying causes of such disparities.

--------------------------------------------------  
Employee or Third Party  
Third Party 185  
Employee 178  
Third Party (Remote) 55  
Name: count, dtype: int64

**Observations:**

## How do accident frequencies differ between employees and third parties?

#### Overview of Data

The data provided includes accident frequencies categorized under 'Employee' and 'Third Party'. The 'Third Party' category is further divided into general and remote incidents.

#### Key Findings

* **Accident Frequency for Employees**: The accident frequency for employees is recorded at **179** incidents.
* **Accident Frequency for Third Parties**: Combining general and remote incidents, the total accident frequency for third parties is **245** (188 general + 57 remote).

#### Conclusion

* **Higher Frequency for Third Parties**: The total accident frequency for third parties (**245**) is significantly higher than that for employees (**179**).
* **Potential Focus Areas**: This suggests a need for targeted safety measures and interventions specifically aimed at environments involving third parties to reduce the overall accident rates.

*Recommendation:*   
Implement enhanced safety protocols and training sessions particularly tailored towards third-party interactions and remote third-party locations to mitigate risks and reduce the frequency of accidents.

--------------------------------------------------  
Critical Risk  
Others 229  
Pressed 24  
Manual Tools 20  
Chemical substances 17  
Cut 14  
Venomous Animals 13  
Projection 13  
Bees 10  
Fall 9  
Vehicles and Mobile Equipment 8  
Fall prevention (same level) 7  
remains of choco 7  
Pressurized Systems 7  
Fall prevention 6  
Suspended Loads 6  
Blocking and isolation of energies 3  
Pressurized Systems / Chemical Substances 3  
Power lock 3  
Liquid Metal 3  
Machine Protection 2  
Electrical Shock 2  
Poll 1  
Individual protection equipment 1  
Projection/Manual Tools 1  
Burn 1  
Electrical installation 1  
Projection/Choco 1  
Projection/Burning 1  
Plates 1  
Confined space 1  
Traffic 1  
\nNot applicable 1  
Projection of fragments 1  
Name: count, dtype: int64

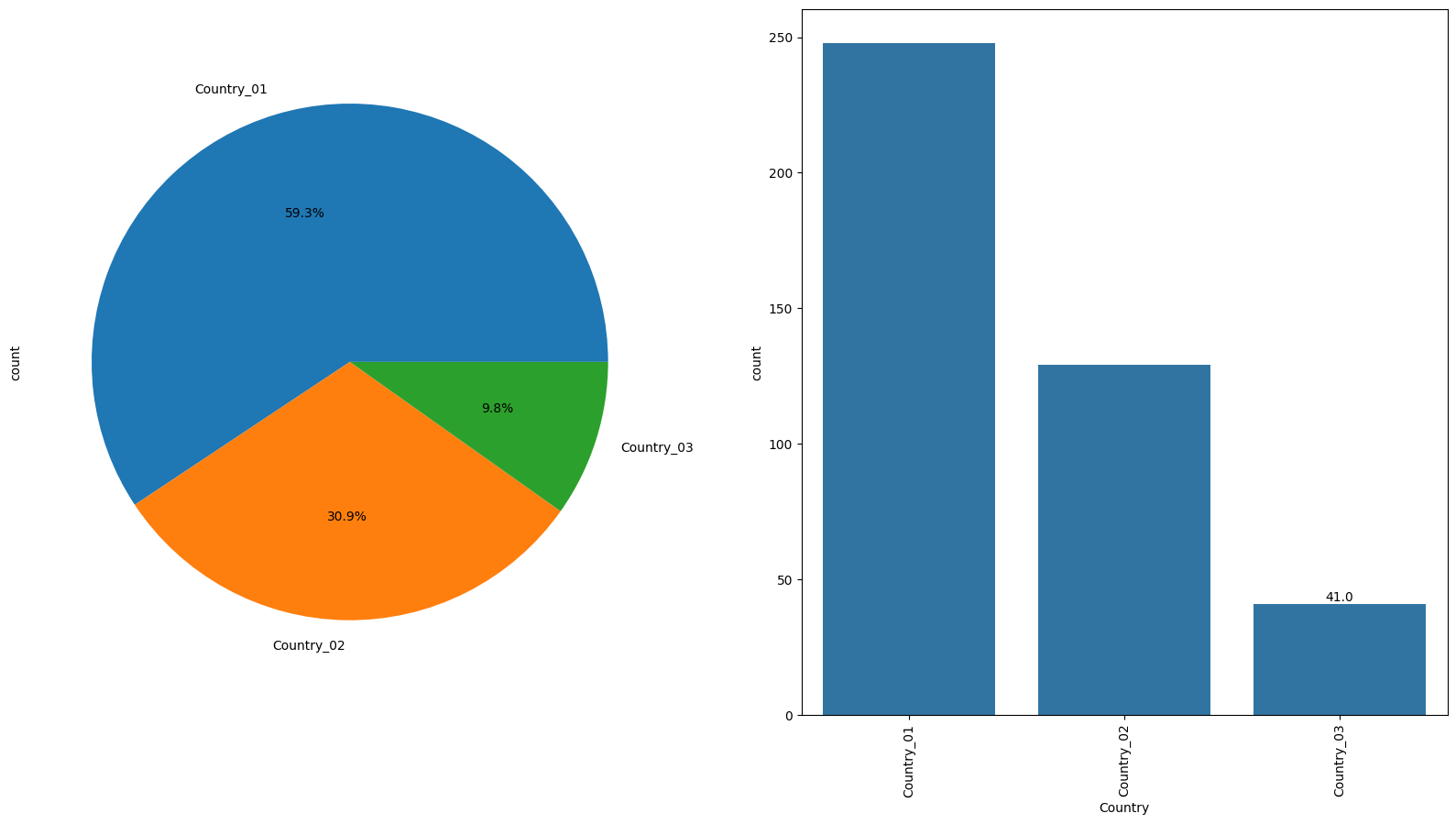
--------------------------------------------------

# Rename 'Data', 'Countries', 'Genre' columns in Data frame  
is\_df.rename(columns={'Data':'Date', 'Countries':'Country', 'Genre':'Gender'}, inplace=True)

# Univariate Analysis

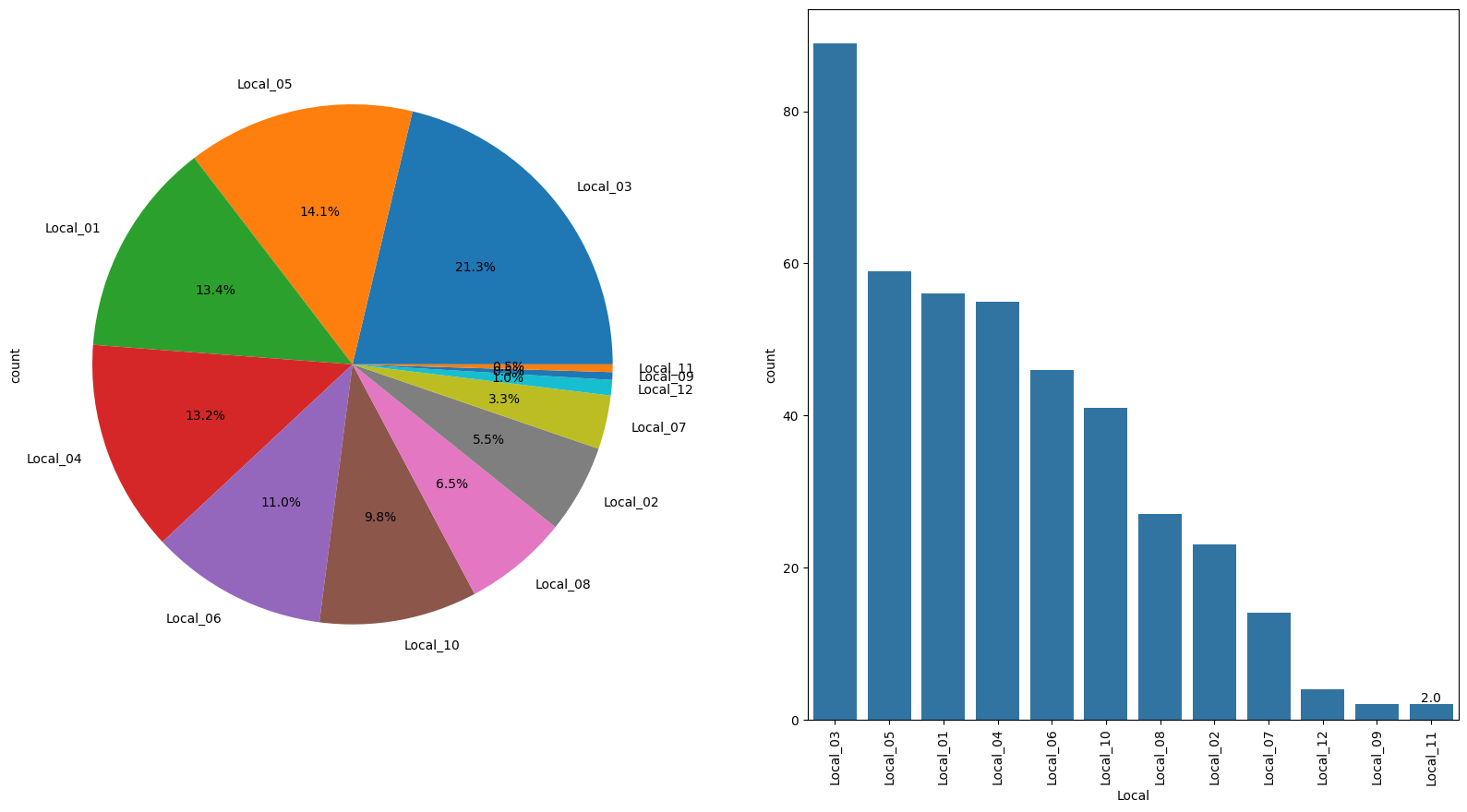
def labeled\_barplot(data, feature, perc=False):  
 """  
 Barplot with percentage at the top  
 data: dataframe  
 feature: dataframe column  
 perc: whether to display percentages instead of count (default is False)  
   
 """  
  
 figure,axes = plt.subplots(nrows =1,ncols = 2,figsize=(20,10))  
 data[feature].value\_counts().plot.pie(autopct='%1.1f%%',ax=axes[0])  
 total = len(data[feature]) # length of the column  
 plt.xticks(rotation=90)  
 ax = sns.countplot(  
 data=data,  
 x=feature,  
 order=data[feature].value\_counts().index,ax = axes[1]  
 )  
  
 for p in ax.patches:  
 if perc == True:  
 label = "{:1.1f}%".format(  
 100 \* p.get\_height() / total  
 ) # percentage of each class of the category  
 else:  
 label = p.get\_height() # count of each level of the category  
  
 x = p.get\_x() + p.get\_width() / 2 # width of the plot  
 y = p.get\_height() # height of the plot  
 ax.annotate(  
 label,  
 (x, y),  
 ha="center",  
 va="center",  
 xytext=(0, 5),  
 textcoords="offset points",  
 ) # annotate the percentage  
 plt.show() # show the plot

# Country distribution  
labeled\_barplot(is\_df, 'Country', perc=False)



**Observations:**  
As observed earlier, highest percentage of accidents almost 59% has occured in Country\_01. Least percentage of accidents, 9.8% has occurred in Country\_03.

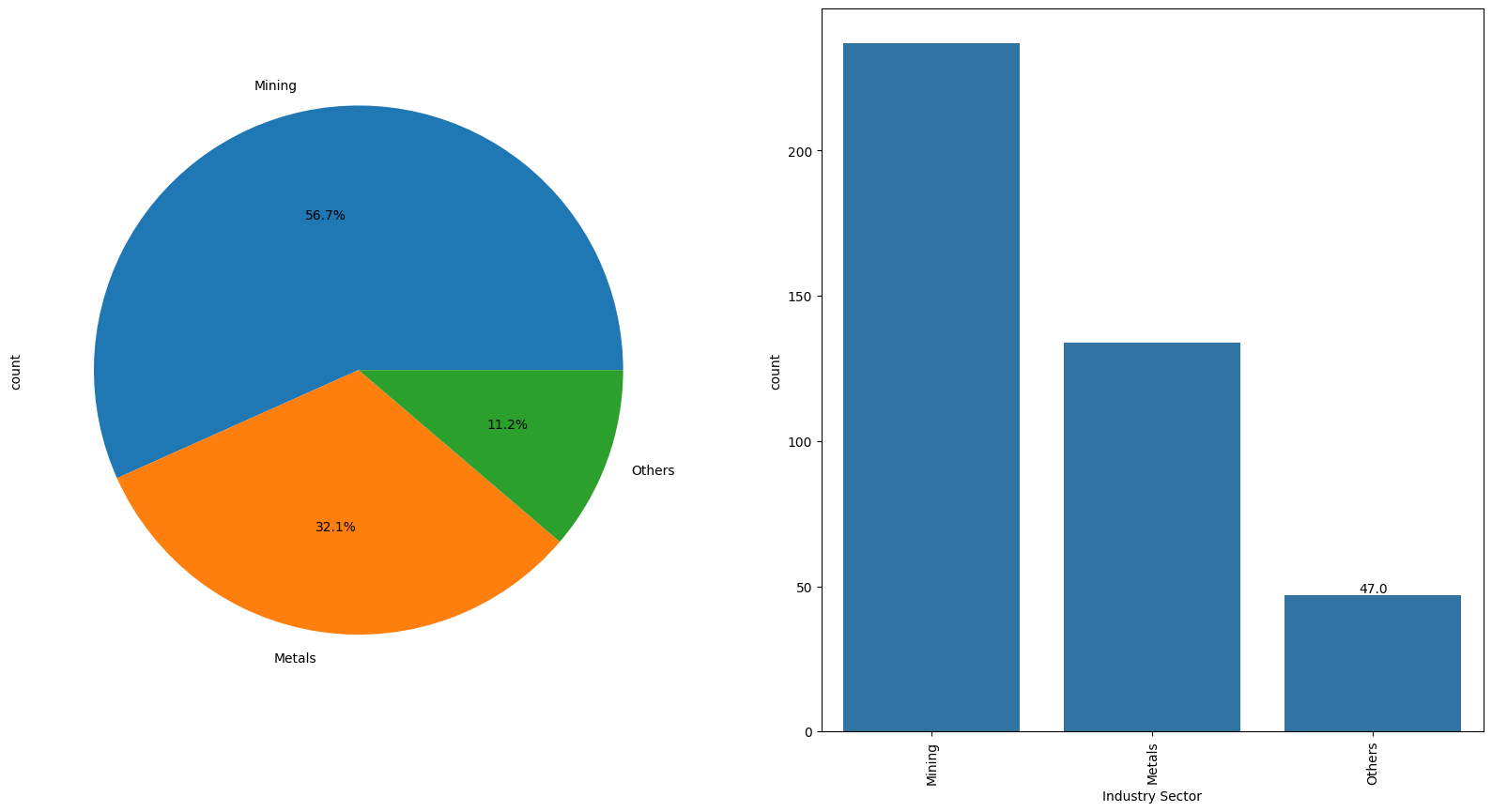
# Local region distribution  
labeled\_barplot(is\_df, 'Local', perc=False)



**Observations:**

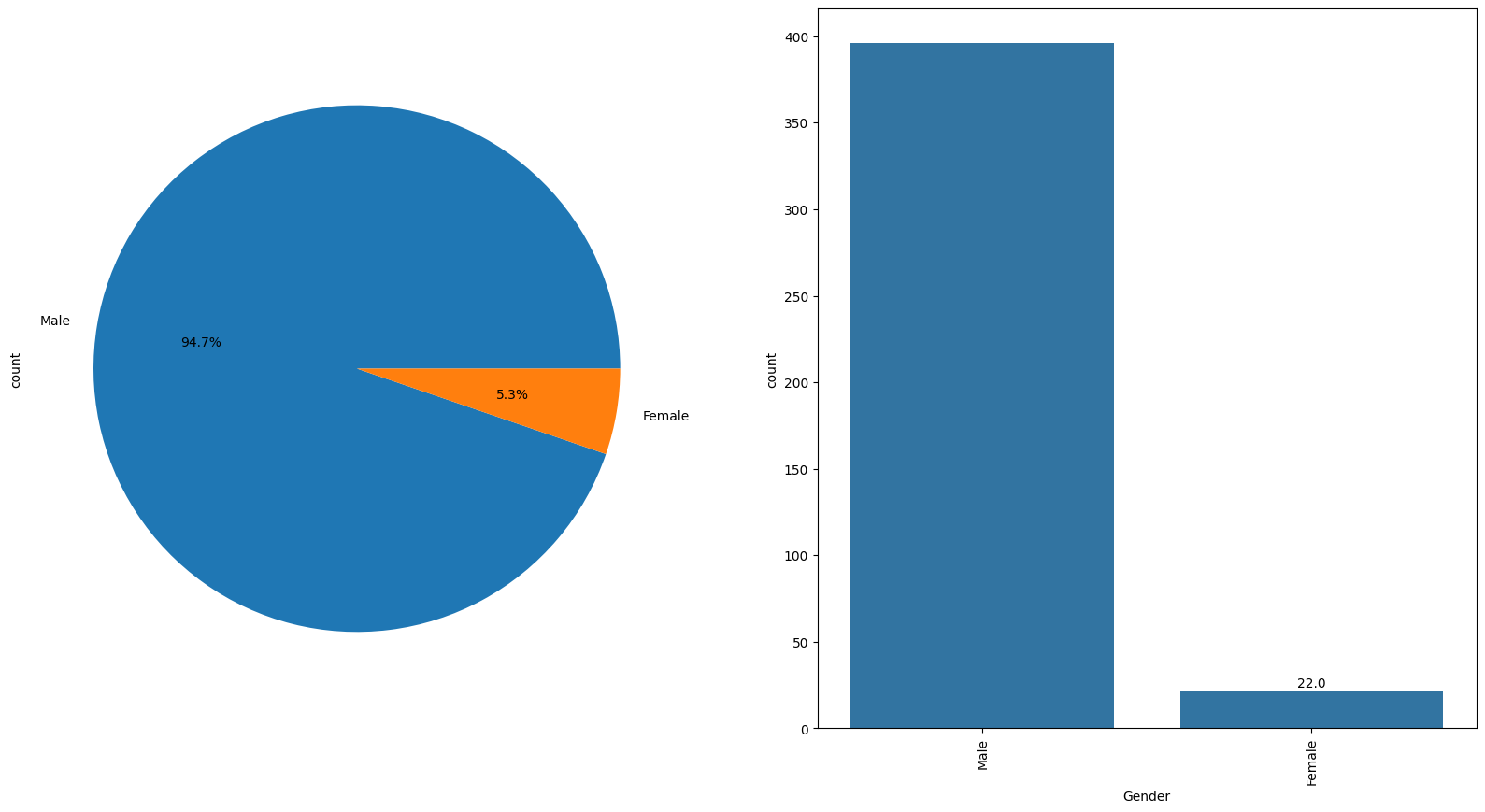
Higher pecentage of accidents has occured in Local\_03 Second highest accident has occurred in Local\_05.

# Industry Sector distribution  
labeled\_barplot(is\_df, 'Industry Sector', perc=False)



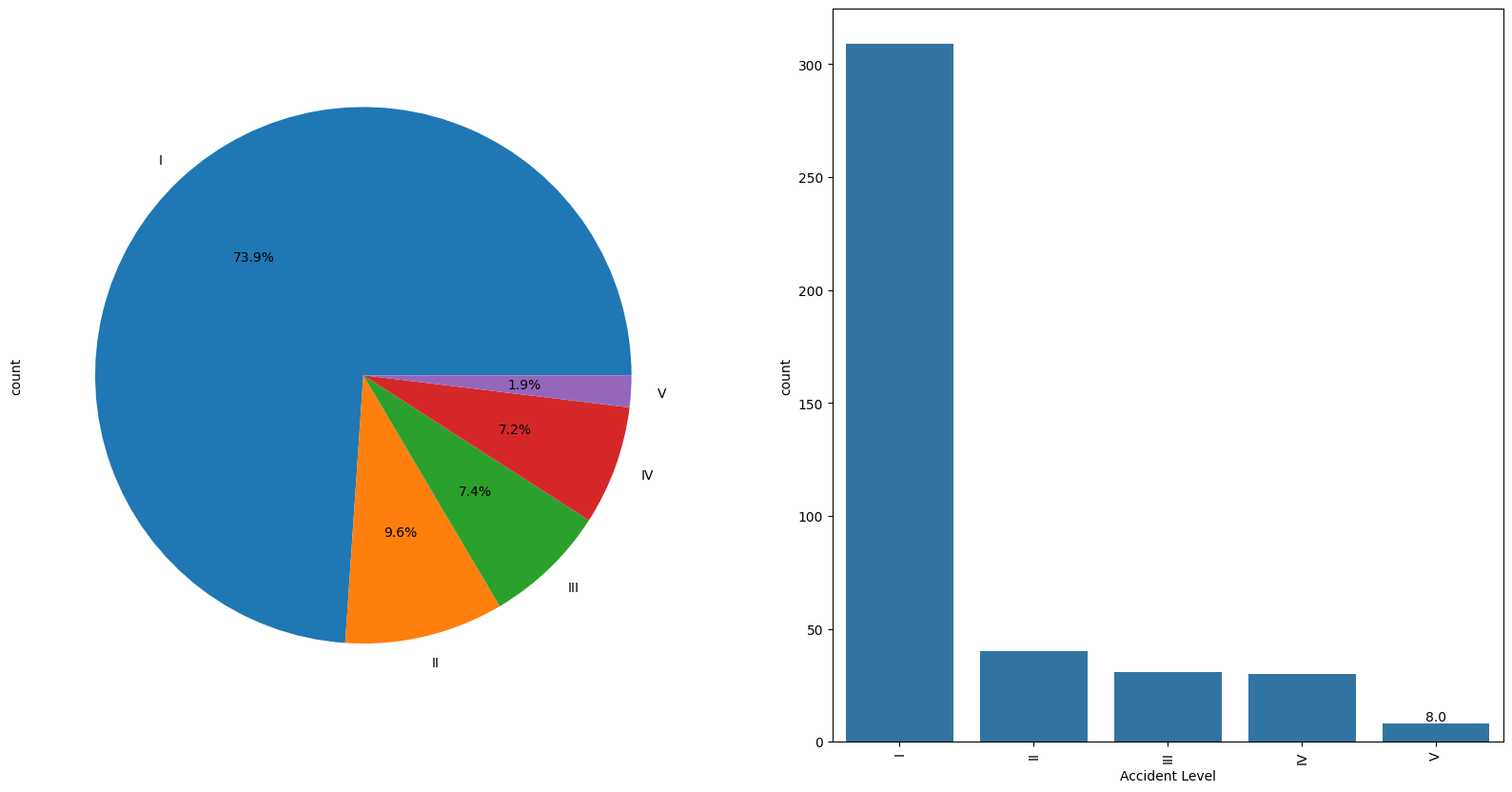
**Observations:**  
Higher percentage, ~ 56.7% of accidents has occurred in Mining industy Second highest in Metals.

# Gender distribution  
labeled\_barplot(is\_df, 'Gender', perc=False)



**Observations:**  
Men have undergone higher percentage of accidents around ~94.7% Women has less accident rate.

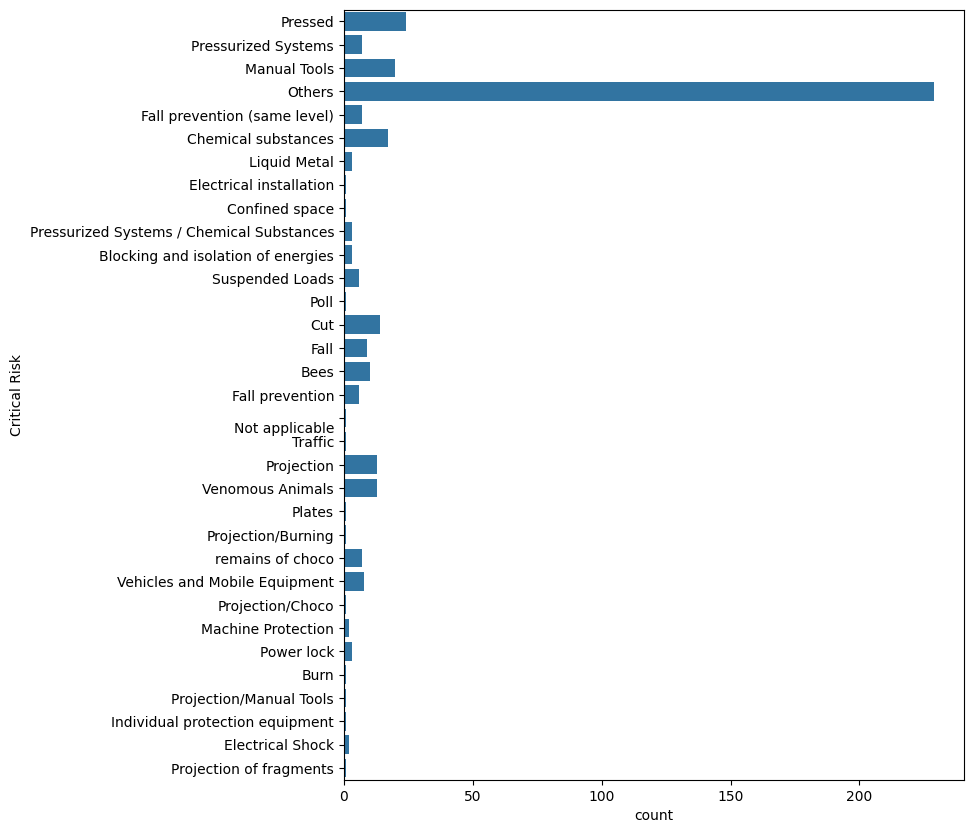
# Accident Level distribution  
labeled\_barplot(is\_df, 'Accident Level', perc=False)



**Observations:**

8% of accidents were very severe considering level V as highest severe. Higher no of accidents happened are less severe ~ 74 %.

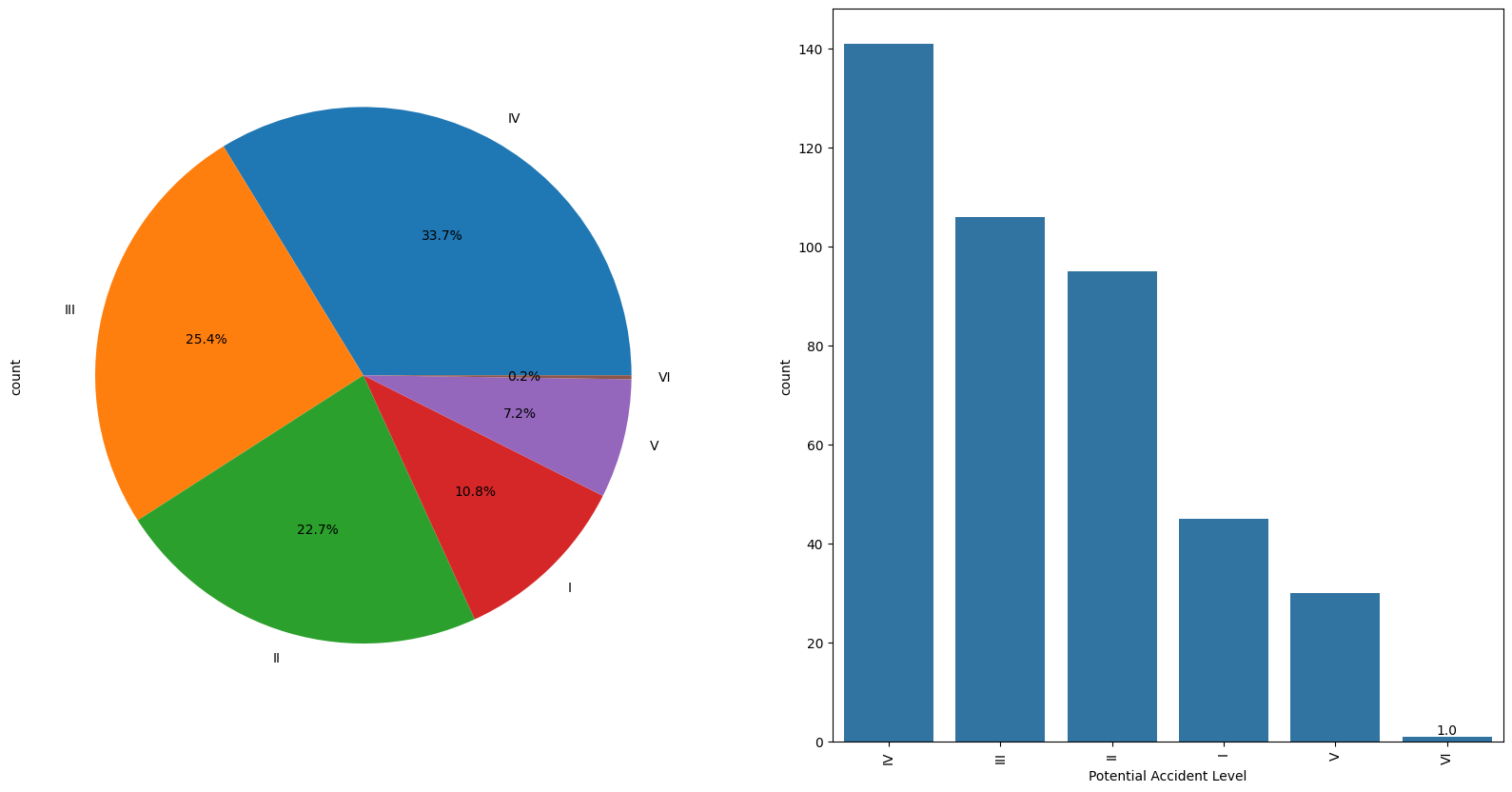
# Critical Risk distribution  
plt.figure(figsize=(8, 10))  
sns.countplot(is\_df, y='Critical Risk');



**Observations:**

Critical risk needs to further collected since most of the falls into other category. Pressed is the second most critical risk reported.

# Potential Accident Level distribution  
labeled\_barplot(is\_df, 'Potential Accident Level', perc=False)



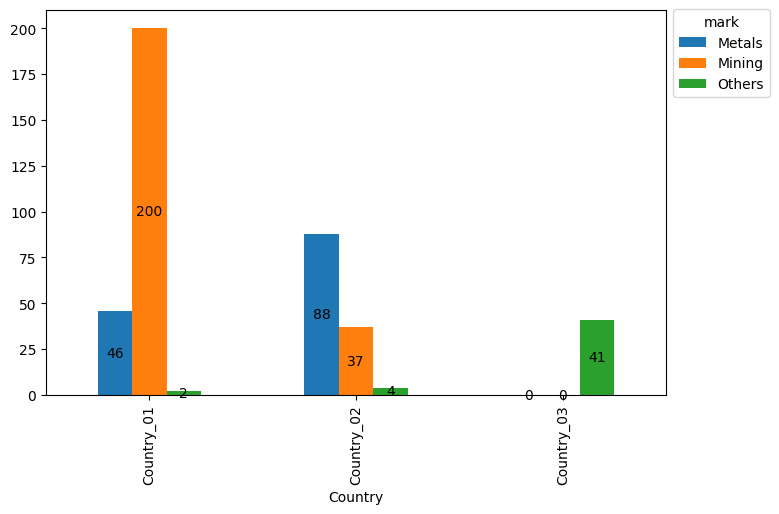
**Observations:**

Most of accidents are of potential level IV.

# Bivariate Analysis

#Defining a function  
def crosstab\_feature(dataframe,index,column,stack):  
 Country\_Local\_table = pd.crosstab(index = dataframe[index], columns = dataframe[column])  
 #ax =Country\_Local\_table.plot(kind = 'bar', figsize=(8,8))  
 ax =Country\_Local\_table.plot(kind = 'bar', figsize=(8,5),stacked = stack)  
 ax.legend(title='mark', bbox\_to\_anchor=(1, 1.02), loc='upper left')  
 # add annotations if desired  
 for c in ax.containers:  
 ax.bar\_label(c, label\_type='center')  
 # plt.title("Proportion of",column,"in different",index)  
 plt.show() # show the plot

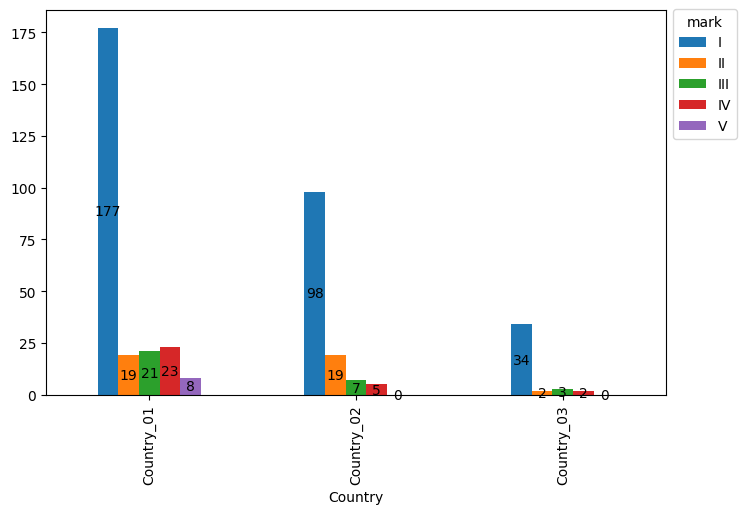
# Check proportion of Industry sector in different countries  
crosstab\_feature(is\_df, 'Country', 'Industry Sector', False)



**Observations:**

Country\_01 has highest percentage of accidents occuring is Mining industy, Country\_02 has highest percentage of accidents occurring in Metal industry, Country\_03 has highest percentage of accidents occurring in Others industry.

# Proportion of accident level in different coutries  
crosstab\_feature(is\_df, 'Country', 'Accident Level', False)



# Treating Attribute data

# Label encoding  
is\_df['Gender'] = is\_df['Gender'].apply(lambda x: {'Male': 0, 'Female': 1}[x])  
is\_df['Accident Level'] = is\_df['Accident Level'].apply(lambda x: {'I': 1, 'II': 2, 'III': 3, 'IV': 4, 'V': 5}[x])  
is\_df['Potential Accident Level'] = is\_df['Potential Accident Level'].apply(lambda x: {'I': 1, 'II': 2, 'III': 3, 'IV': 4, 'V': 5, 'VI': 6}[x])  
  
is\_df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 418 entries, 0 to 417  
Data columns (total 10 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Date 418 non-null datetime64[ns]  
 1 Country 418 non-null object   
 2 Local 418 non-null object   
 3 Industry Sector 418 non-null object   
 4 Accident Level 418 non-null int64   
 5 Potential Accident Level 418 non-null int64   
 6 Gender 418 non-null int64   
 7 Employee or Third Party 418 non-null object   
 8 Critical Risk 418 non-null object   
 9 Description 418 non-null object   
dtypes: datetime64[ns](1), int64(3), object(6)  
memory usage: 32.8+ KB

# Dropping datetime info  
is\_df.drop(['Date'], axis=1, inplace=True)

# One-hot encoding  
is\_df = pd.get\_dummies(is\_df, columns=['Country', 'Local', 'Industry Sector', 'Employee or Third Party', 'Critical Risk'], dtype=np.int64)

is\_df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 418 entries, 0 to 417  
Data columns (total 58 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Accident Level 418 non-null int64   
 1 Potential Accident Level 418 non-null int64   
 2 Gender 418 non-null int64   
 3 Description 418 non-null object  
 4 Country\_Country\_01 418 non-null int64   
 5 Country\_Country\_02 418 non-null int64   
 6 Country\_Country\_03 418 non-null int64   
 7 Local\_Local\_01 418 non-null int64   
 8 Local\_Local\_02 418 non-null int64   
 9 Local\_Local\_03 418 non-null int64   
 10 Local\_Local\_04 418 non-null int64   
 11 Local\_Local\_05 418 non-null int64   
 12 Local\_Local\_06 418 non-null int64   
 13 Local\_Local\_07 418 non-null int64   
 14 Local\_Local\_08 418 non-null int64   
 15 Local\_Local\_09 418 non-null int64   
 16 Local\_Local\_10 418 non-null int64   
 17 Local\_Local\_11 418 non-null int64   
 18 Local\_Local\_12 418 non-null int64   
 19 Industry Sector\_Metals 418 non-null int64   
 20 Industry Sector\_Mining 418 non-null int64   
 21 Industry Sector\_Others 418 non-null int64   
 22 Employee or Third Party\_Employee 418 non-null int64   
 23 Employee or Third Party\_Third Party 418 non-null int64   
 24 Employee or Third Party\_Third Party (Remote) 418 non-null int64   
 25 Critical Risk\_  
Not applicable 418 non-null int64   
 26 Critical Risk\_Bees 418 non-null int64   
 27 Critical Risk\_Blocking and isolation of energies 418 non-null int64   
 28 Critical Risk\_Burn 418 non-null int64   
 29 Critical Risk\_Chemical substances 418 non-null int64   
 30 Critical Risk\_Confined space 418 non-null int64   
 31 Critical Risk\_Cut 418 non-null int64   
 32 Critical Risk\_Electrical Shock 418 non-null int64   
 33 Critical Risk\_Electrical installation 418 non-null int64   
 34 Critical Risk\_Fall 418 non-null int64   
 35 Critical Risk\_Fall prevention 418 non-null int64   
 36 Critical Risk\_Fall prevention (same level) 418 non-null int64   
 37 Critical Risk\_Individual protection equipment 418 non-null int64   
 38 Critical Risk\_Liquid Metal 418 non-null int64   
 39 Critical Risk\_Machine Protection 418 non-null int64   
 40 Critical Risk\_Manual Tools 418 non-null int64   
 41 Critical Risk\_Others 418 non-null int64   
 42 Critical Risk\_Plates 418 non-null int64   
 43 Critical Risk\_Poll 418 non-null int64   
 44 Critical Risk\_Power lock 418 non-null int64   
 45 Critical Risk\_Pressed 418 non-null int64   
 46 Critical Risk\_Pressurized Systems 418 non-null int64   
 47 Critical Risk\_Pressurized Systems / Chemical Substances 418 non-null int64   
 48 Critical Risk\_Projection 418 non-null int64   
 49 Critical Risk\_Projection of fragments 418 non-null int64   
 50 Critical Risk\_Projection/Burning 418 non-null int64   
 51 Critical Risk\_Projection/Choco 418 non-null int64   
 52 Critical Risk\_Projection/Manual Tools 418 non-null int64   
 53 Critical Risk\_Suspended Loads 418 non-null int64   
 54 Critical Risk\_Traffic 418 non-null int64   
 55 Critical Risk\_Vehicles and Mobile Equipment 418 non-null int64   
 56 Critical Risk\_Venomous Animals 418 non-null int64   
 57 Critical Risk\_remains of choco 418 non-null int64   
dtypes: int64(57), object(1)  
memory usage: 189.5+ KB

# Description

# to use regular expressions for manipulating text data  
import re  
  
# to load the natural language toolkit  
import nltk  
nltk.download('stopwords') # loading the stopwords  
nltk.download('wordnet')   
  
# to remove common stop words  
from nltk.corpus import stopwords  
  
# to perform stemming  
from nltk.stem.porter import PorterStemmer  
  
# to create Bag of Words  
from sklearn.feature\_extraction.text import CountVectorizer

[nltk\_data] Downloading package stopwords to  
[nltk\_data] C:\Users\suhai\AppData\Roaming\nltk\_data...  
[nltk\_data] Package stopwords is already up-to-date!  
[nltk\_data] Downloading package wordnet to  
[nltk\_data] C:\Users\suhai\AppData\Roaming\nltk\_data...  
[nltk\_data] Package wordnet is already up-to-date!

# To lowercase  
is\_df['Description\_T'] = is\_df['Description'].apply(lambda x: x.lower())

# Removing non-alphanumeric chars  
is\_df['Description\_T'] = is\_df['Description\_T'].apply(lambda x: ''.join(re.sub('[^A-Za-z0-9]+', ' ', x)))

# Removing extra white spaces  
is\_df['Description\_T'] = is\_df['Description\_T'].str.strip()

# Stopword removal  
is\_df['Description\_T'] = is\_df['Description\_T'].apply(lambda x: ' '.join([word for word in x.split() if word not in stopwords.words('english')]))

is\_df.loc[0:10, ['Description', 'Description\_T']]

Description \  
0 While removing the drill rod of the Jumbo 08 f...   
1 During the activation of a sodium sulphide pum...   
2 In the sub-station MILPO located at level +170...   
3 Being 9:45 am. approximately in the Nv. 1880 C...   
4 Approximately at 11:45 a.m. in circumstances t...   
5 During the unloading operation of the ustulado...   
6 The collaborator reports that he was on street...   
7 At approximately 04:50 p.m., when the mechanic...   
8 Employee was sitting in the resting area at le...   
9 At the moment the forklift operator went to ma...   
10 While installing a segment of the polyurethane...   
  
 Description\_T   
0 removing drill rod jumbo 08 maintenance superv...   
1 activation sodium sulphide pump piping uncoupl...   
2 sub station milpo located level 170 collaborat...   
3 9 45 approximately nv 1880 cx 695 ob7 personne...   
4 approximately 11 45 circumstances mechanics an...   
5 unloading operation ustulado bag need unclog d...   
6 collaborator reports street 09 holding left ha...   
7 approximately 04 50 p mechanic technician jos ...   
8 employee sitting resting area level 326 raise ...   
9 moment forklift operator went manipulate big b...   
10 installing segment polyurethane pulley protect...

nltk.download('punkt')  
from nltk.stem import WordNetLemmatizer  
  
lemmatizer = WordNetLemmatizer()  
is\_df['Description\_WL'] = is\_df.apply(lambda row: nltk.word\_tokenize(row['Description\_T']), axis=1)  
def lemmatize\_list(words):  
 new\_words = []  
 for word in words:  
 new\_words.append(lemmatizer.lemmatize(word, pos='v'))  
 return ' '.join(new\_words)  
is\_df['Description\_WL'] = is\_df.apply(lambda x: lemmatize\_list(x['Description\_WL']), axis=1)

[nltk\_data] Downloading package punkt to  
[nltk\_data] C:\Users\suhai\AppData\Roaming\nltk\_data...  
[nltk\_data] Unzipping tokenizers\punkt.zip.

is\_df.loc[0:10, ['Description', 'Description\_WL']]

Description \  
0 While removing the drill rod of the Jumbo 08 f...   
1 During the activation of a sodium sulphide pum...   
2 In the sub-station MILPO located at level +170...   
3 Being 9:45 am. approximately in the Nv. 1880 C...   
4 Approximately at 11:45 a.m. in circumstances t...   
5 During the unloading operation of the ustulado...   
6 The collaborator reports that he was on street...   
7 At approximately 04:50 p.m., when the mechanic...   
8 Employee was sitting in the resting area at le...   
9 At the moment the forklift operator went to ma...   
10 While installing a segment of the polyurethane...   
  
 Description\_WL   
0 remove drill rod jumbo 08 maintenance supervis...   
1 activation sodium sulphide pump pip uncouple s...   
2 sub station milpo locate level 170 collaborato...   
3 9 45 approximately nv 1880 cx 695 ob7 personne...   
4 approximately 11 45 circumstances mechanics an...   
5 unload operation ustulado bag need unclog disc...   
6 collaborator report street 09 hold leave hand ...   
7 approximately 04 50 p mechanic technician jos ...   
8 employee sit rest area level 326 raise bore su...   
9 moment forklift operator go manipulate big bag...   
10 instal segment polyurethane pulley protective ...

**Observations:**

## Can we identify any patterns or commonalities in the descriptions of accidents?

## Patterns can be identified in the incident description with the help of NLP techniques. Initially, Stemming and Tokenization can be used for generating the word embeddings.

* Porter Stemmer is used for stemming of incident description and converting them into root-words.

from sklearn.feature\_extraction.text import TfidfVectorizer  
tfidf\_df = pd.DataFrame()  
for i in [1,2,3,4]:  
 tfidf = TfidfVectorizer(max\_features=1000, stop\_words='english',use\_idf=True, ngram\_range=(i,i))  
 X = tfidf.fit\_transform(is\_df['Cleansed\_Description']).toarray()  
 tfs = pd.DataFrame(X, columns=["TFIDF\_" + n for n in tfidf.get\_feature\_names\_out()])  
 tfidf\_df = pd.concat([tfidf\_df.reset\_index(drop=True), tfs.reset\_index(drop=True)], axis=1)  
  
tfidf\_df.head(3)  
  
final\_dataset= read\_df\_dummy\_encoding.join(tfidf\_df.reset\_index(drop=True))

**Observations:**

## Can we perform feature extraction on the descriptions of accidents?

TF-IDF vectorizer, Bag of Words (BOW), N-grams are some of the most popular text feature extraction methods.

TF-IDF vectorizer is a tool used in text processing, It converts words into numbers, assessing their importance with scoring, TF counts how often words appear in a document, IDF measures how unique words are across documents, to augment, Together, TF-IDF helps prioritize words that are most significant in the text corpus.