

# **Government College of Technology, Coimbatore**

## **TN Marginal Workers Assessment**

### **Team Members:**

- 1. VenkatesaPerumal R**
- 2. Aravinth M**
- 3. Sugash M**

### **Project Abstraction:**

The Tamil Nadu Marginal Workers Assessment, utilizing Data Analytics with Cognos, delves into the employment patterns of workers in Tamil Nadu employed for less than six months. This analysis provides key insights into their demographics, industries, and regional distribution, guiding targeted policies to enhance their working conditions and livelihoods.

### **1. Loading a Dataset:**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv('/content/DDW_B06SC_3300_State_TAMIL_NADU-2011.csv')
df

df.shape
df.info()
df.describe()
```

TN\_Marginal\_Workers\_Assessment.ipynb

File

Edit

View

Insert

Runtime

Tools

Help

All changes saved

+ Code

+ Text

✓ T4

Disk

	Table Code	State Code	District Code	Area Name	Total/ Rural/ Urban	Age group	Worked for 3 months or more but less than 6 months - Persons	Worked for 3 months or more but less than 6 months - Males	Worked for 3 months or more but less than 6 months - Females	Worked for less than 3 months - Persons	...	Industrial Category - N to O - Females	Industrial Category - P to Q - Persons	Industrial Category - P to Q - Males	Industrial Category - P to Q - Females	Industrial Category - R to U - HHI - Persons	Industrial Category - R to U - HHI - Males	Industrial Category - R to U - HHI - Females	Industrial Category - R to U - Non HHI - Persons
0	B0806SC	'33	'000	State - TAMIL NADU	Total	Total	1200828	589003	611825	221386	...	3565	11080	4019	7061	16833	4266	12567	122
1	B0806SC	'33	'000	State - TAMIL NADU	Total	'5-14	27791	14125	13666	2447	...	11	122	71	51	427	169	258	15
2	B0806SC	'33	'000	State - TAMIL NADU	Total	15-34	514340	259560	254780	92423	...	1754	7536	2718	4818	8346	2127	6219	66
3	B0806SC	'33	'000	State - TAMIL NADU	Total	35-59	542581	251957	290624	99202	...	1619	3205	1131	2074	6591	1487	5104	26
4	B0806SC	'33	'000	State - TAMIL NADU	Total	60+	115103	62833	52270	27165	...	175	211	93	118	1457	483	974	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
589	B0806SC	'33	'633	District - Tiruppur	Urban	'5-14	272	129	143	18	...	0	0	0	0	0	0	0	0

✓ Connected to Python 3 Google Compute Engine backend (GPU)

## 2. Data Preprocessing:

```
df.isnull().sum
df.dtypes

df.isnull().values.any()

df.drop(columns=['Table Code','District Code','State Code'],inplace=True)
```

TN\_Marginal\_Workers\_Assessment.ipynb

df.isnull().sum()	Table Code State Code District Code Area Name Total/ Rural/ Urban Industrial Category - R to U - HHI - Males Industrial Category - R to U - HHI - Females Industrial Category - R to U - Non HHI - Persons Industrial Category - R to U - Non HHI - Males Industrial Category - R to U - Non HHI - Females Length: 69, dtype: int64	0 0 0 0 0 0 0 0 0 0 0
df.dtypes	Table Code State Code District Code Area Name Total/ Rural/ Urban Industrial Category - R to U - HHI - Males Industrial Category - R to U - HHI - Females Industrial Category - R to U - Non HHI - Persons Industrial Category - R to U - Non HHI - Males Industrial Category - R to U - Non HHI - Females Length: 69, dtype: object	object object object object object int64 int64 int64 int64 int64 int64
df.isnull().values.any()		

Connected to Python 3 Google Compute Engine backend (GPU)

```
from sklearn.preprocessing import LabelEncoder
label=LabelEncoder()
df['Area Name']=label.fit_transform(df['Area Name'])
df['Age group']=label.fit_transform(df['Age group'])
```

```
df['Total/ Rural/ Urban']=label.fit_transform(df['Total/ Rural/ Urban'])
df.head()
```

TN\_Marginal\_Workers\_Assessment.ipynb

File Edit View Insert Runtime Tools Help All changes saved

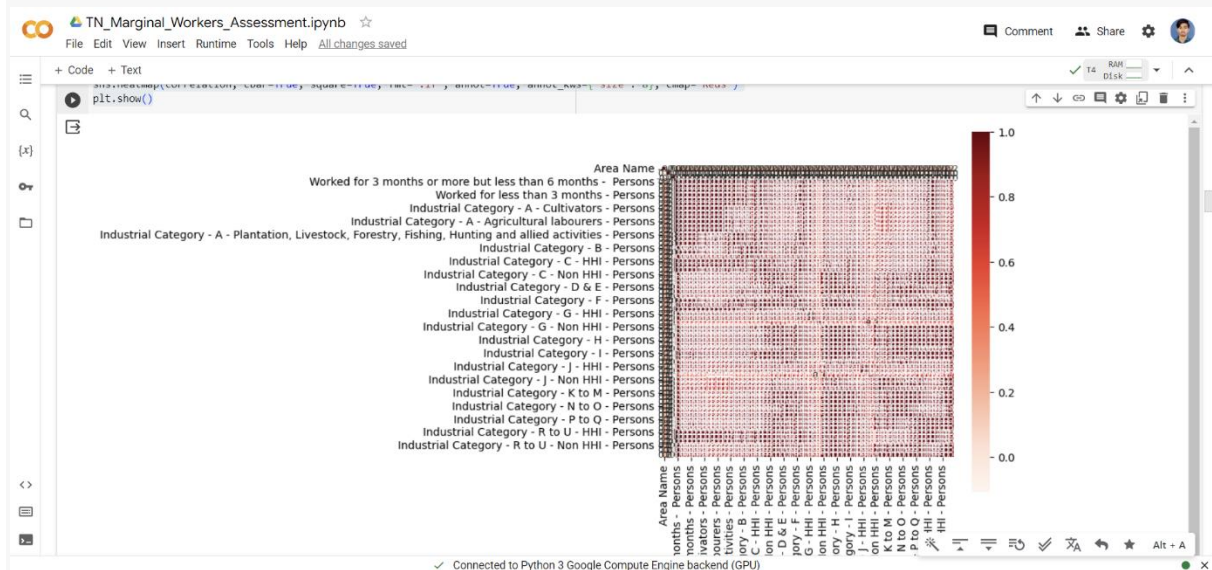
+ Code + Text

Area Name Total/ Rural/ Urban Age group Worked for 3 months or more but less than 6 months - Persons Worked for 3 months or more but less than 6 months - Males Worked for 3 months or more but less than 6 months - Females Worked for less than 3 months - Persons Worked for less than 3 months - Males Worked for less than 3 months - Females Industrial Category - A - Cultivators - Persons Industrial Category - N to O - Females Industrial Category - P to Q - Persons Industrial Category - P to Q - Males Industrial Category - P to Q - Females Industrial Category - R to U - HHI - Persons Industrial Category - R to U - HHI - Males Industrial Category - R to U - HHI - Females Industrial Category - R to U - Non HHI - Persons

0	32	1	4	1200828	589003	611825	221386	99368	122018	64235	...	3565	11080	4019	7061	16833	4266	12567	1220
1	32	1	5	27791	14125	13666	2447	1247	1200	1710	...	11	122	71	51	427	169	258	193
2	32	1	0	514340	259560	254780	92423	43892	48531	24863	...	1754	7536	2718	4818	8346	2127	6219	689
3	32	1	1	542581	251957	290624	99202	40691	58511	29692	...	1619	3205	1131	2074	6591	1487	5104	264
4	32	1	2	115103	62833	52270	27165	13465	13700	7930	...	175	211	93	118	1457	483	974	70

5 rows x 66 columns

```
correlation=df.corr()
plt.figure(figsize=(6,6))
sns.heatmap(correlation, cbar=True, square=True, fmt='.1f', annot=True,
annot_kws={'size': 8}, cmap='Reds')
plt.show()
```



### 3. Data Visualization:

```
# Perform demographic analysis

# Calculate distribution based on age

age_distribution = df['Age group'].value_counts()
age_distribution

df['Combined Category - A - Persons'] = (
```

```

    df['Industrial Category - A - Plantation, Livestock, Forestry,
Fishing, Hunting and allied activities - Persons'].astype(str) +
    df['Industrial Category - A - Agricultural labourers -
Persons'].astype(str) +
    df['Industrial Category - A - Cultivators - Persons'].astype(str)
)
df['Combined Category - A - Persons']

industrial_category_distribution_A = df['Combined Category - A -
Persons'].value_counts()
industrial_category_distribution_A

industrial_category_distribution_B = df['Industrial Category - B -
Persons'].value_counts()
industrial_category_distribution_B

df['Combined Category - C - Persons'] = (
    df['Industrial Category - C - HHI - Persons'].astype(str) +
    df['Industrial Category - C - Non HHI - Persons'].astype(str)
)
industrial_category_distribution_C = df['Combined Category - C -
Persons'].value_counts()
industrial_category_distribution_C

industrial_category_distribution_D_E = df['Industrial Category - D & E
- Persons'].value_counts()
industrial_category_distribution_D_E

industrial_category_distribution_F = df['Industrial Category - F -
Persons'].value_counts()
industrial_category_distribution_F

df['Combined Category - G - Persons'] = (
    df['Industrial Category - G - HHI - Persons'].astype(str) +
    df['Industrial Category - G - Non HHI - Persons'].astype(str)
)
industrial_category_distribution_G = df['Combined Category - G -
Persons'].value_counts()
industrial_category_distribution_G

industrial_category_distribution_H = df['Industrial Category - H -
Persons'].value_counts()
industrial_category_distribution_H

industrial_category_distribution_I = df['Industrial Category - I -
Persons'].value_counts()
industrial_category_distribution_I

```

```

df['Combined Category - J - Persons'] = (
    df['Industrial Category - J - HHI - Persons'].astype(str) +
    df['Industrial Category - J - Non HHI - Persons'].astype(str)
)
industrial_category_distribution_J = df['Combined Category - J -
Persons'].value_counts()
industrial_category_distribution_J

industrial_category_distribution_K_and_M = df['Industrial Category - K
to M - Persons'].value_counts()
industrial_category_distribution_K_and_M

industrial_category_distribution_N_and_O = df['Industrial Category - N
to O - Persons'].value_counts()

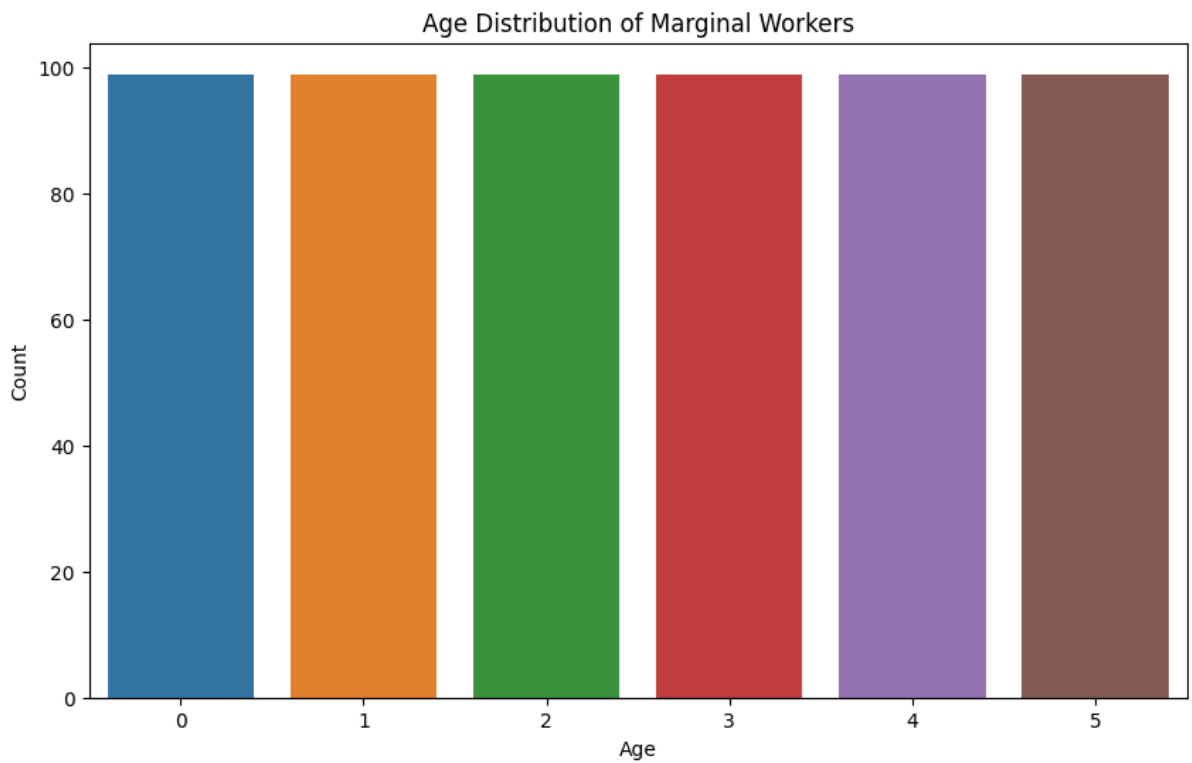
industrial_category_distribution_P_and_Q = df['Industrial Category - P
to Q - Persons'].value_counts()
industrial_category_distribution_P_and_Q

df['Combined Category - R_to_U - Persons'] = (
    df['Industrial Category - R to U - HHI - Persons'].astype(str) +
    df['Industrial Category - R to U - Non HHI - Persons'].astype(str)
)
industrial_category_distribution_R_to_U = df['Combined Category - R_to_U
- Persons'].value_counts()
industrial_category_distribution_R_to_U

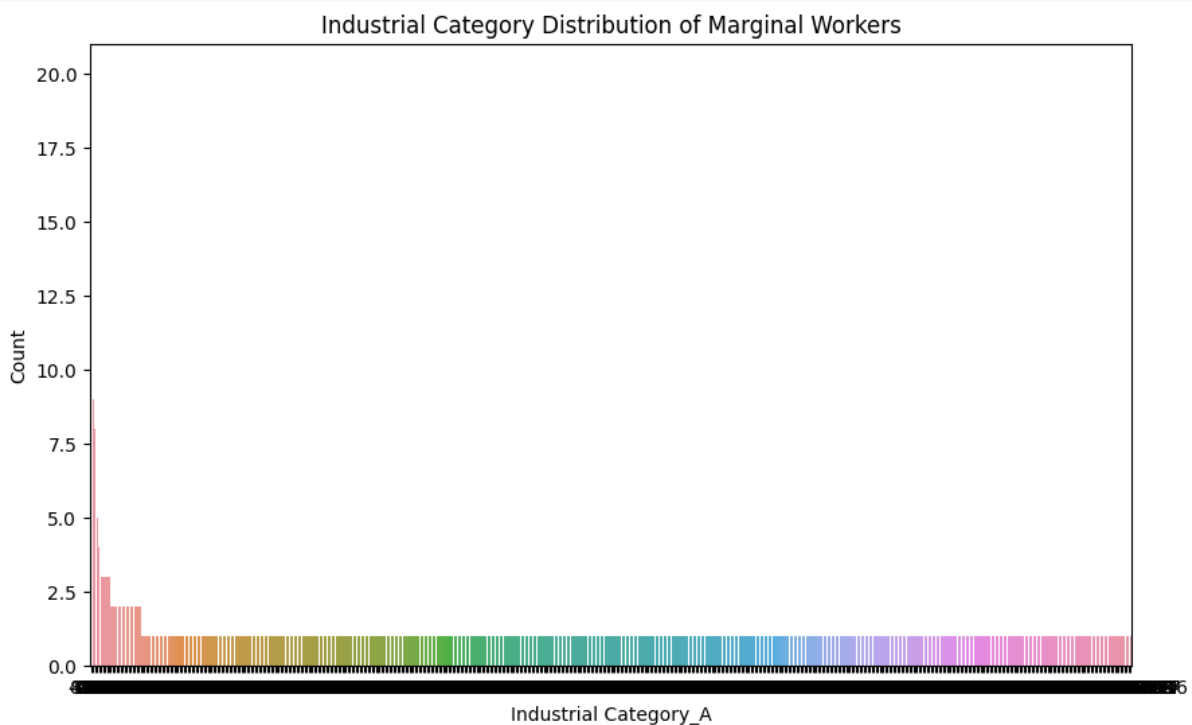
# Create visualizations

# Visualization 1: Age distribution
plt.figure(figsize=(10, 6))
sns.barplot(x=age_distribution.index, y=age_distribution.values)
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age Distribution of Marginal Workers')
plt.show()

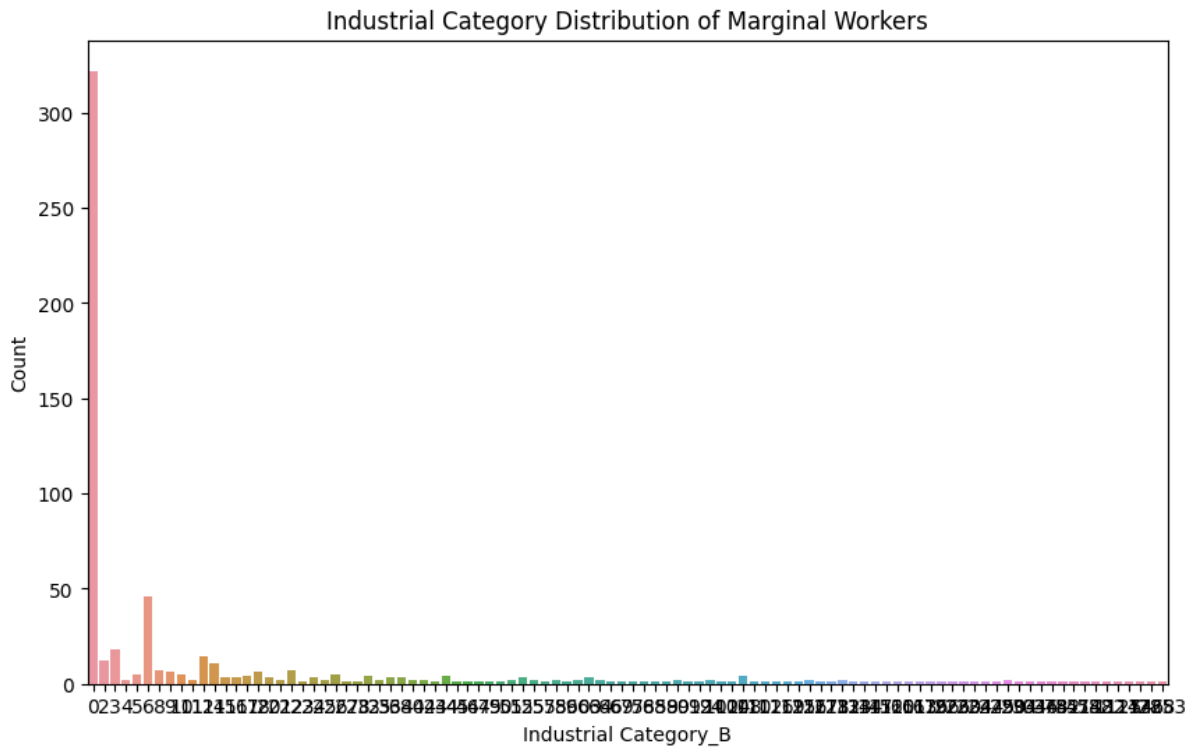
```



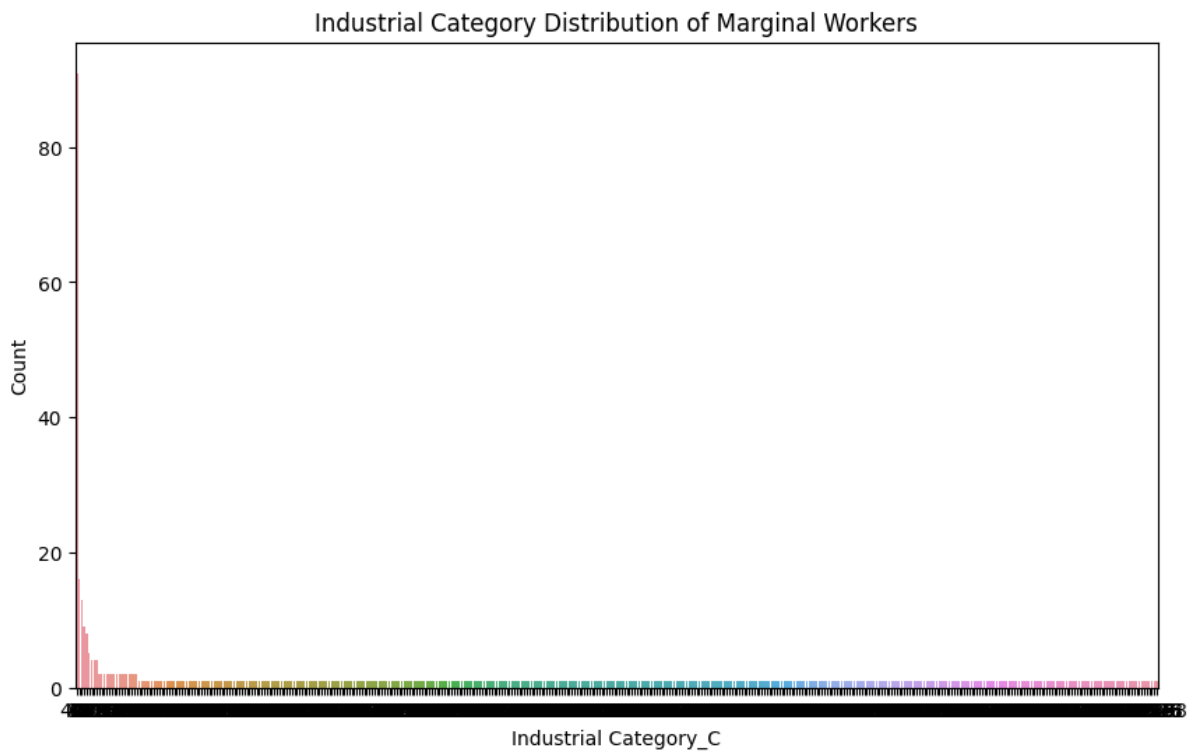
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_A.index,
y=industrial_category_distribution_A.values)
plt.xlabel('Industrial Category_A')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



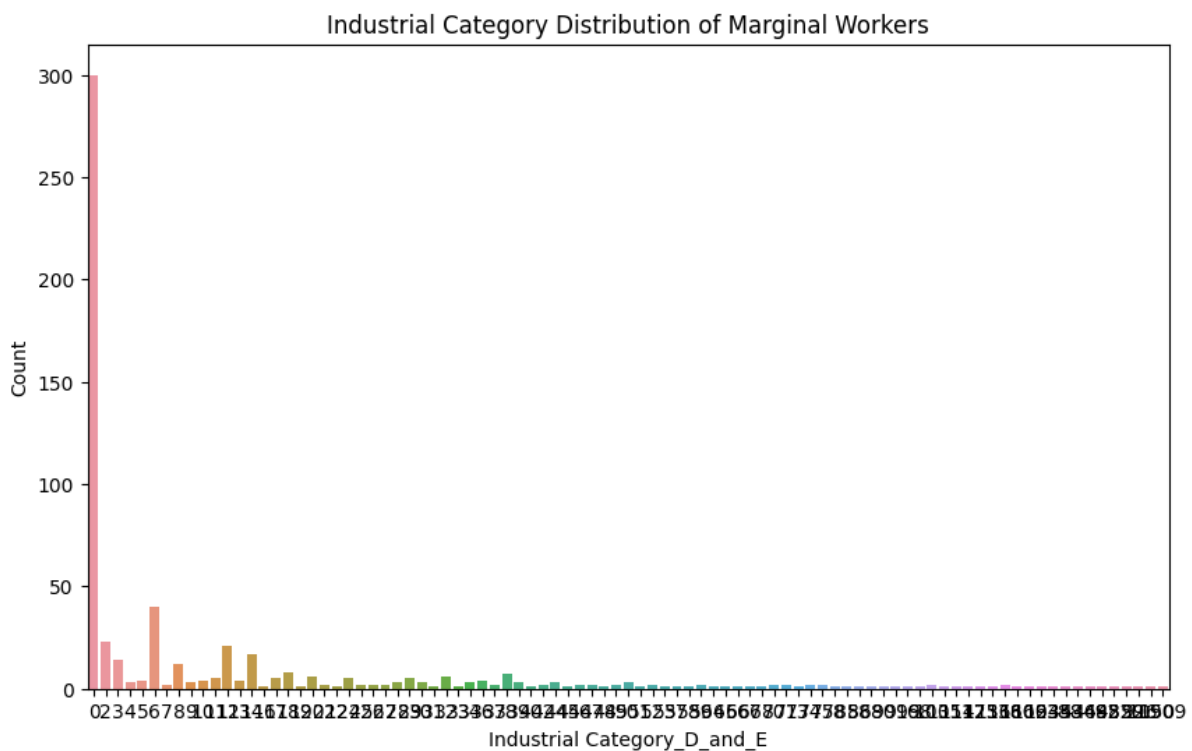
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_B.index,
y=industrial_category_distribution_B.values)
plt.xlabel('Industrial Category_B')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_C.index,
y=industrial_category_distribution_C.values)
plt.xlabel('Industrial Category_C')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```

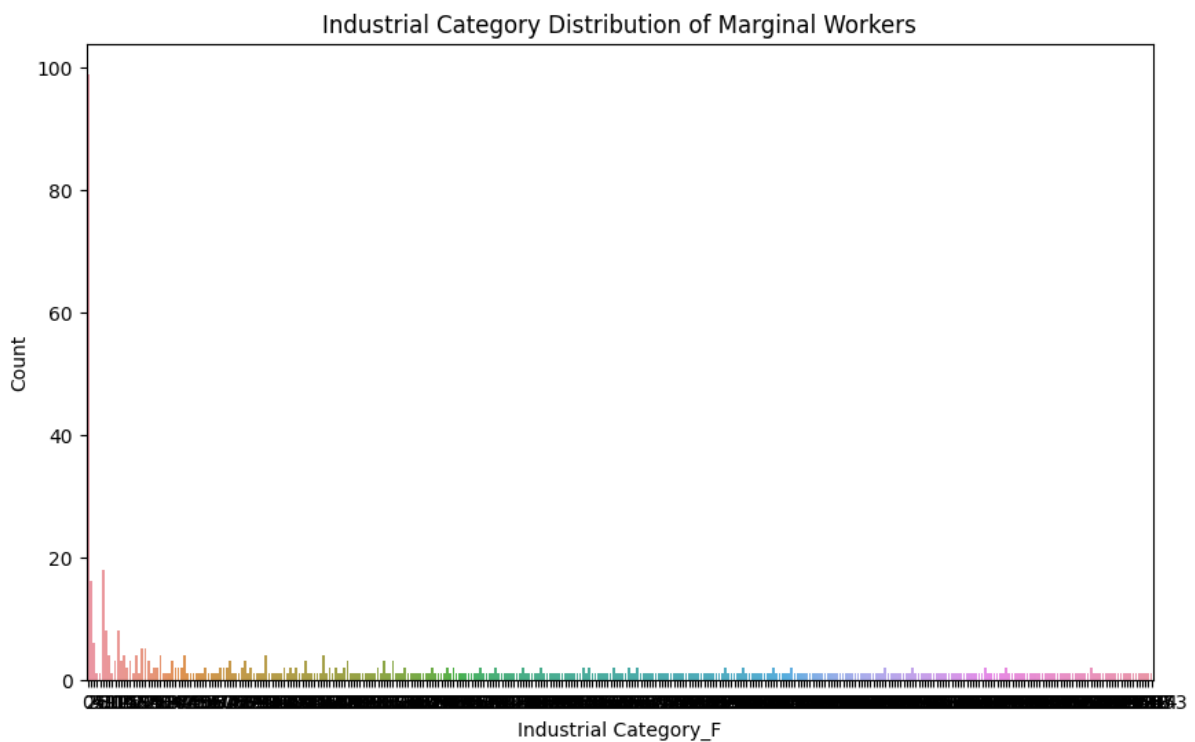


```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_D_E.index,
y=industrial_category_distribution_D_E.values)
plt.xlabel('Industrial Category_D_and_E')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```

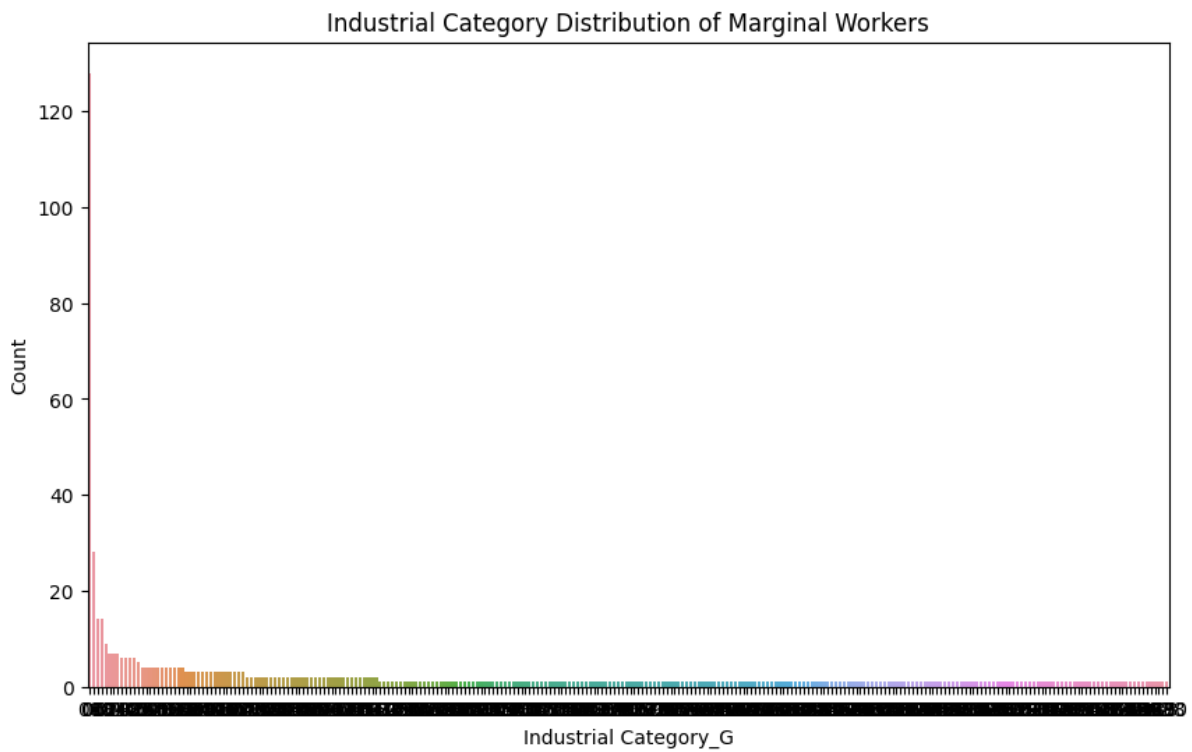




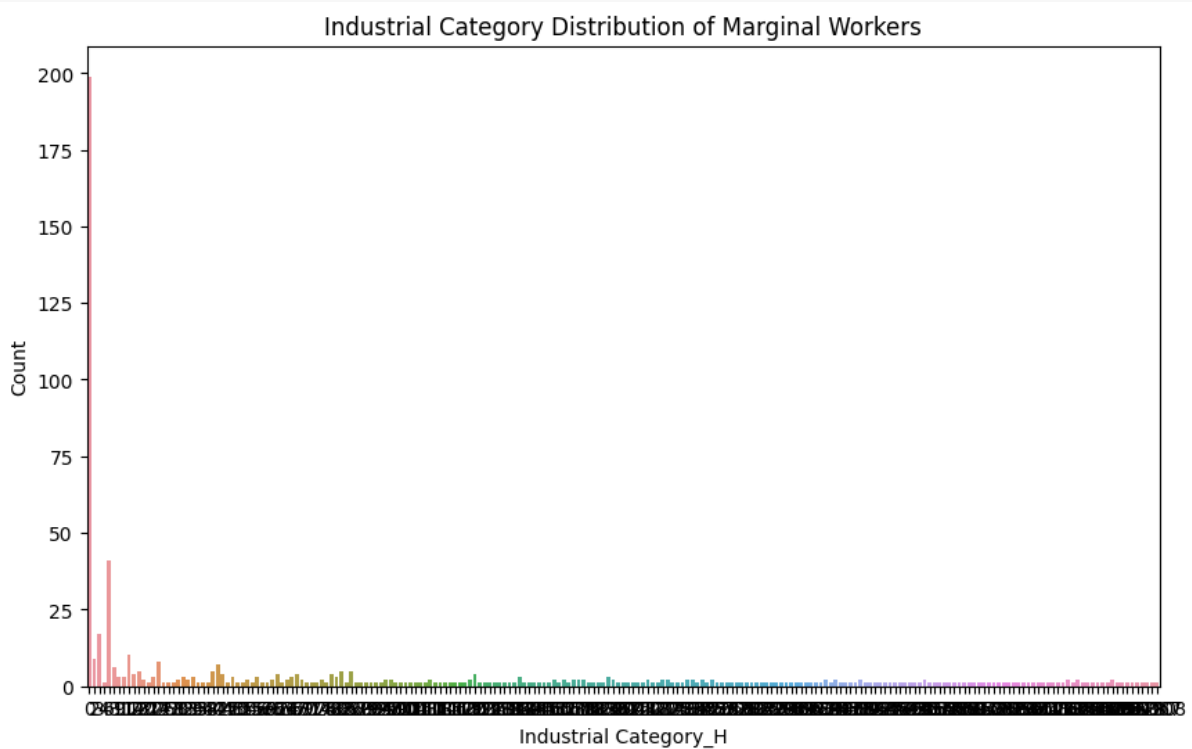
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_F.index,
y=industrial_category_distribution_F.values)
plt.xlabel('Industrial Category_F')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



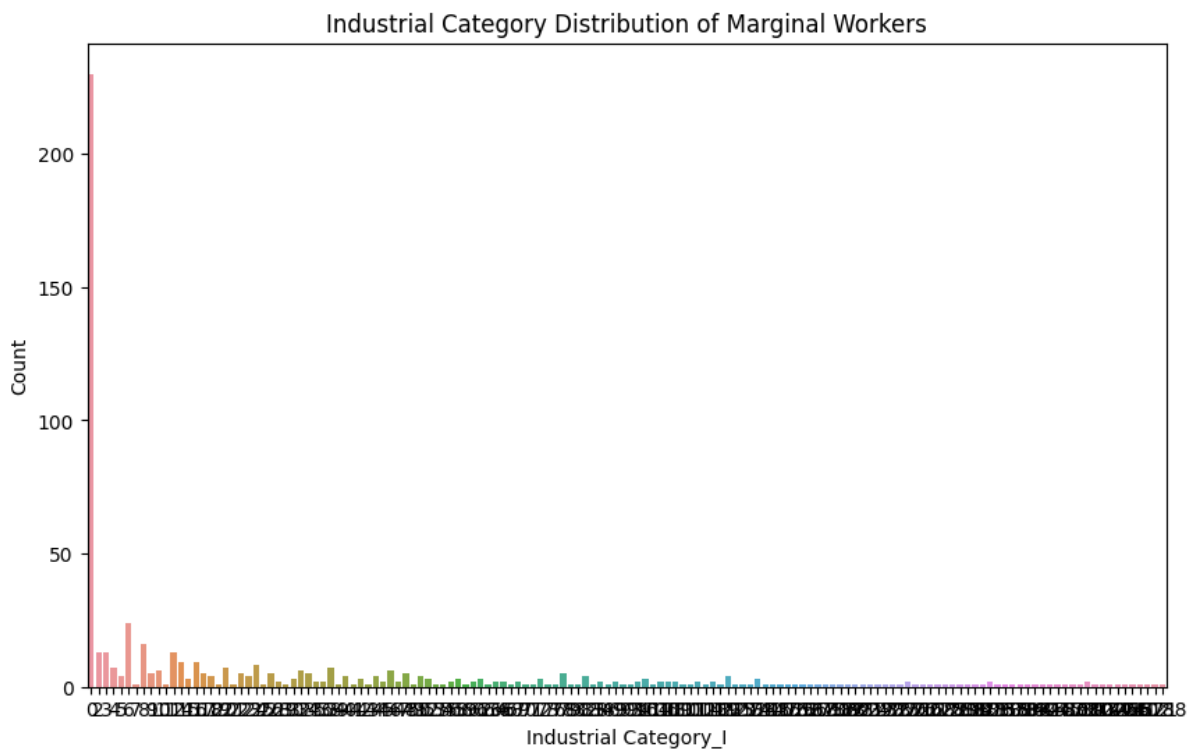
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_G.index,
y=industrial_category_distribution_G.values)
plt.xlabel('Industrial Category_G')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



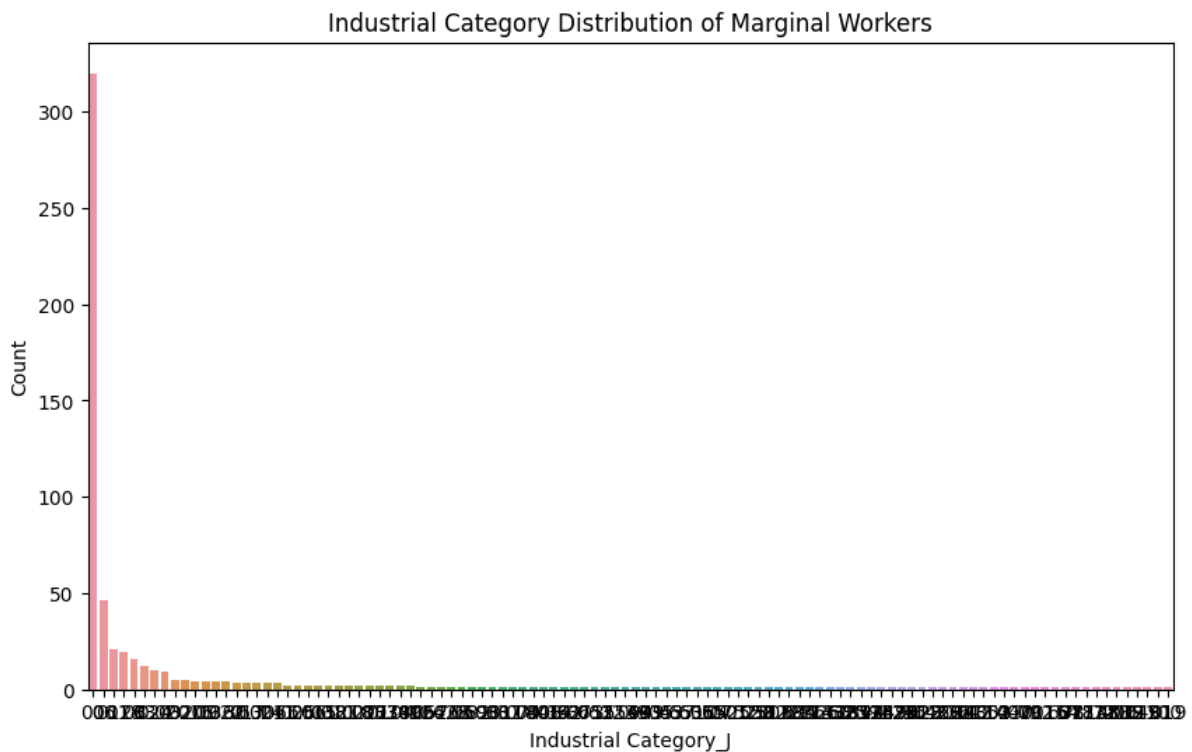
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_H.index,
y=industrial_category_distribution_H.values)
plt.xlabel('Industrial Category_H')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



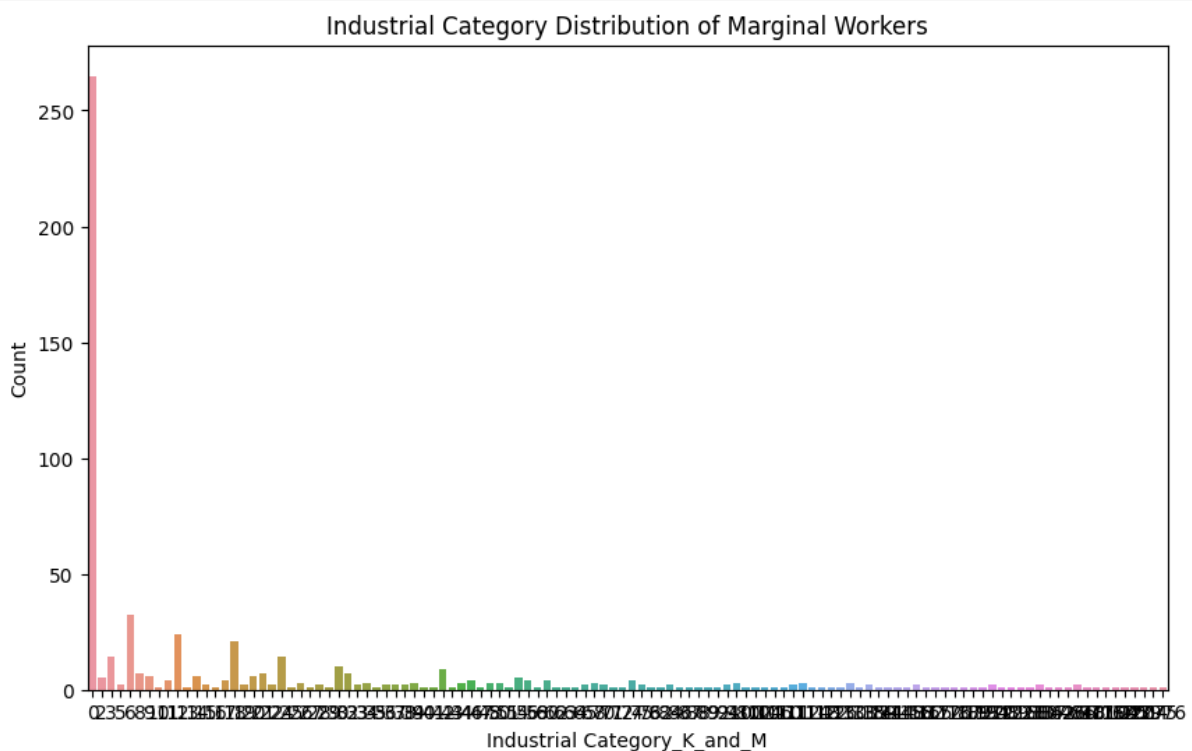
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_I.index,
y=industrial_category_distribution_I.values)
plt.xlabel('Industrial Category_I')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



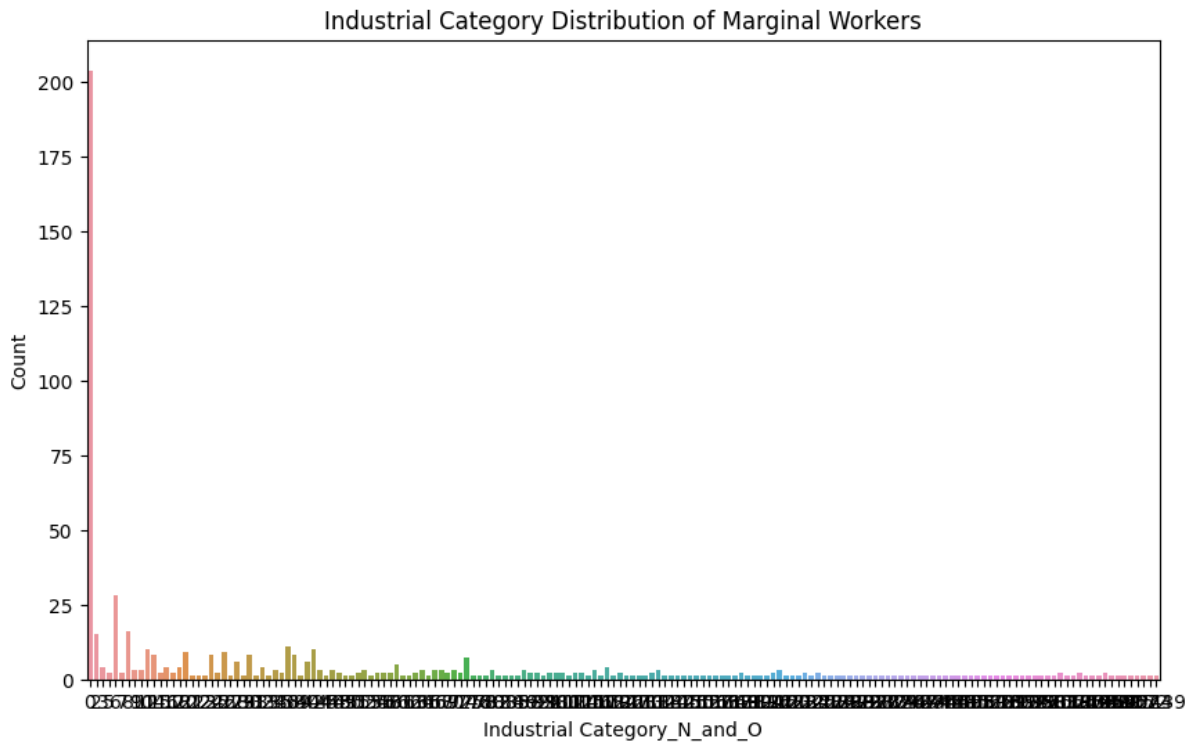
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_J.index,
y=industrial_category_distribution_J.values)
plt.xlabel('Industrial Category_J')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



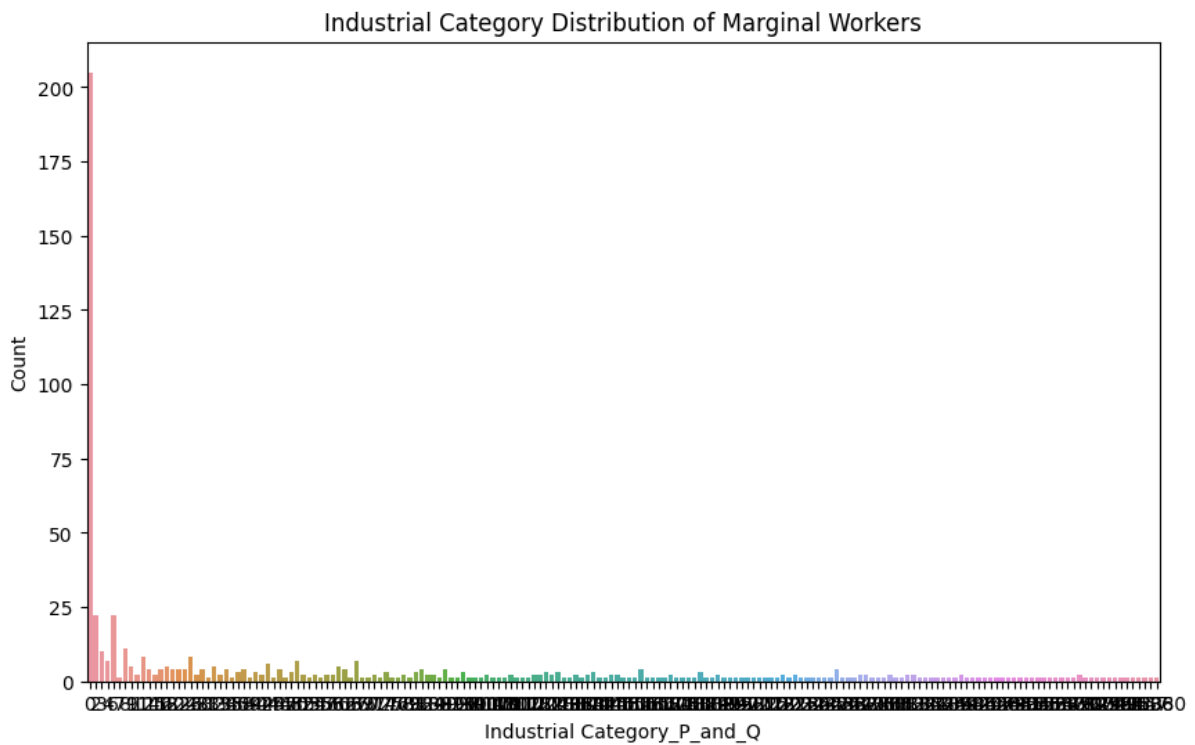
```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_K_and_M.index,
y=industrial_category_distribution_K_and_M.values)
plt.xlabel('Industrial Category_K_and_M')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_N_and_O.index,
y=industrial_category_distribution_N_and_O.values)
plt.xlabel('Industrial Category_N_and_O')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_P_and_Q.index,
y=industrial_category_distribution_P_and_Q.values)
plt.xlabel('Industrial Category_P_and_Q')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```



```
plt.figure(figsize=(10, 6))
sns.barplot(x=industrial_category_distribution_R_to_U.index,
y=industrial_category_distribution_R_to_U.values)
plt.xlabel('Industrial Category_R_to_U')
plt.ylabel('Count')
plt.title('Industrial Category Distribution of Marginal Workers')
plt.show()
```

