In [1]:
```
import pandas as pd
df = pd.read_csv('C:/Users/eliko/Documents/NeedsAnalysis2Data.csv')
df.head()
```

Out[1]:

	Focal1EmpRating	Focal1SupRating	Focal1EmpImportance	Focal1SupImportance	Focal1EmpTr
0	2	1	3	5	
1	3	1	4	5	
2	2	3	4	5	
3	4	1	5	4	
4	4	1	2	3	

5 rows × 23 columns

In [57]:
```
#check data type for values
type(df["Focal1EmpRating"])
```

Out[57]:
```
pandas.core.series.Series
```

In [3]:
```
df['Focal1EmpRating'].mean()
```

Out[3]:
```
2.74
```

In [5]:
```
df['Focal1SupRating'].mean()
```

Out[5]:
```
2.16
```

In [43]:
```
df.columns
```

Out[43]:
```
Index(['Focal1EmpRating', 'Focal1SupRating', 'Focal1EmpImportance',
       'Focal1SupImportance', 'Focal1EmpTrainable', 'Focal1SupTrainable',
       'Focal2EmpRating', 'Focal2SupRating', 'Focal2EmpImportance',
       'Focal2SubImportance', 'Focal2EmpTrainable', 'Focal2SupTrainable',
       'Focal3EmpRating', 'Focal3SupRating', 'Focal3EmpImportance',
       'Focal3SupImportance', 'Focal3EmpTrainable', 'Focal3SupTrainable',
       'OralExpression', 'MechanicalAbility', 'SafetyClimate1',
       'SafetyClimate2', 'SafetyClimate3'],
      dtype='object')
```

In [10]:
```
#quick look at all column means, as collected in dict
means = {}
for x in df.columns:
    means[x] = df[x].mean()
print(means)
```

```
{'Focal1EmpRating': 2.74, 'Focal1SupRating': 2.16, 'Focal1EmpImportance': 4.
1, 'Focal1SupImportance': 4.6, 'Focal1EmpTrainable': 3.74, 'Focal1SupTrainabl
e': 4.0, 'Focal2EmpRating': 4.16, 'Focal2SupRating': 2.22, 'Focal2EmpImportan
ce': 1.92, 'Focal2SubImportance': 3.7, 'Focal2EmpTrainable': 1.7, 'Focal2SupT
rainable': 2.22, 'Focal3EmpRating': 2.4, 'Focal3SupRating': 2.2, 'Focal3EmpIm
portance': 4.1, 'Focal3SupImportance': 4.04, 'Focal3EmpTrainable': 3.68, 'Foc
al3SupTrainable': 4.18, 'OralExpression': 41.86, 'MechanicalAbility': 65.88,
'SafetyClimate1': 2.24, 'SafetyClimate2': 2.42, 'SafetyClimate3': 2.28}
```

In [31]:
```
#Getting the differences between supervisor and employee means
diffmeans = []
listmeans = list(means.items())
i = 0
while i < len(listmeans)-5:
    diffmeans = diffmeans + [[f"Difference in {listmeans[i][0]} and {listmeans
    i += 2

print(diffmeans)
```

```
[['Difference in Focal1EmpRating and Focal1SupRating.', 0.58], ['Difference i
n Focal1EmpImportance and Focal1SupImportance.', -0.5], ['Difference in Focal
1EmpTrainable and Focal1SupTrainable.', -0.26], ['Difference in Focal2EmpRati
ng and Focal2SupRating.', 1.94], ['Difference in Focal2EmpImportance and Foca
l2SubImportance.', -1.78], ['Difference in Focal2EmpTrainable and Focal2SupTr
ainable.', -0.52], ['Difference in Focal3EmpRating and Focal3SupRating.', 0.
2], ['Difference in Focal3EmpImportance and Focal3SupImportance.', 0.06], ['D
ifference in Focal3EmpTrainable and Focal3SupTrainable.', -0.5]]
```

In [32]:
```
#Time to perform T-tests to support potential significant differences
from scipy.stats import ttest_ind
res = ttest_ind(df['Focal1EmpRating'], df['Focal1SupRating'])
print(res)
```

```
Ttest_indResult(statistic=3.472750228120718, pvalue=0.0007682640406015517)
```

In [45]:
```
#Now, let's do T-test's between each of our employee and supervisor ratings
ttests = []
i = 0
keys = df.columns
while i < len(df.columns)-5:
    ttests = ttests + [[f"Independent T-Test for {keys[i]} and {keys[i+1]}.",
    i += 2
print(ttests)
```

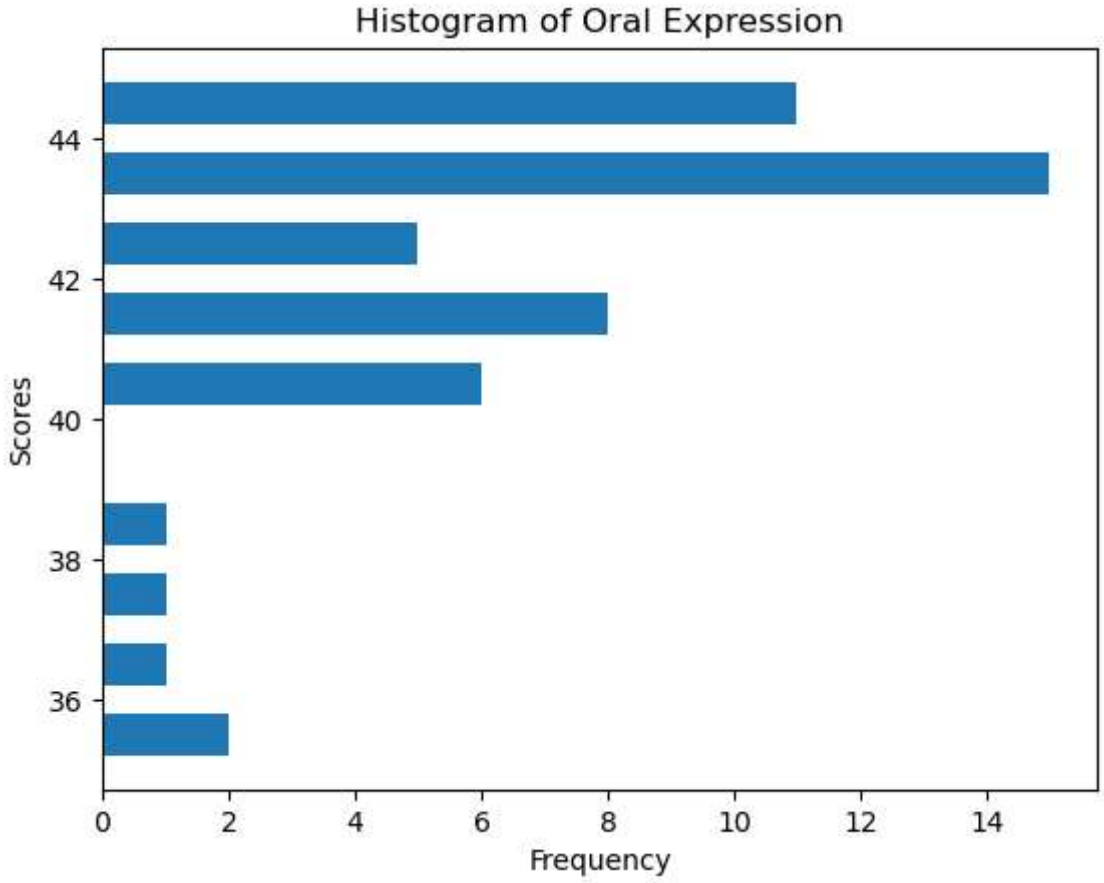
```
[['Independent T-Test for Focal1EmpRating and Focal1SupRating.', Ttest_indRes
ult(statistic=3.472750228120718, pvalue=0.0007682640406015517)], ['Independen
t T-Test for Focal1EmpImportance and Focal1SupImportance.', Ttest_indResult(s
tatistic=-3.553711577967667, pvalue=0.0005862648781109677)], ['Independent T-
Test for Focal1EmpTrainable and Focal1SupTrainable.', Ttest_indResult(statist
ic=-1.0383230849780485, pvalue=0.30167477024247064)], ['Independent T-Test fo
r Focal2EmpRating and Focal2SupRating.', Ttest_indResult(statistic=9.35773164
289695, pvalue=2.9997832437400253e-15)], ['Independent T-Test for Focal2EmpIm
portance and Focal2SubImportance.', Ttest_indResult(statistic=-10.67024879961
031, pvalue=4.272286934423485e-18)], ['Independent T-Test for Focal2EmpTraina
ble and Focal2SupTrainable.', Ttest_indResult(statistic=-2.585790805464462, p
value=0.011187077692173473)], ['Independent T-Test for Focal3EmpRating and Fo
cal3SupRating.', Ttest_indResult(statistic=0.8890008890013323, pvalue=0.37617
95694828107)], ['Independent T-Test for Focal3EmpImportance and Focal3SupImpo
rtance.', Ttest_indResult(statistic=0.2684158175954511, pvalue=0.788943473423
0891)], ['Independent T-Test for Focal3EmpTrainable and Focal3SupTrainable.',
Ttest_indResult(statistic=-2.0897105209369777, pvalue=0.039233469170979224)]]
```

In [56]:
```
#Now we want to filter out those that are not statistically significant.
for x in ttests:
    if x[1][1] <= 0.05:
        print(f"{x[0]} is significant!", round(x[1][1],6))
```

```
Independent T-Test for Focal1EmpRating and Focal1SupRating. is significant!
0.000768
Independent T-Test for Focal1EmpImportance and Focal1SupImportance. is signif
icant! 0.000586
Independent T-Test for Focal2EmpRating and Focal2SupRating. is significant!
0.0
Independent T-Test for Focal2EmpImportance and Focal2SubImportance. is signif
icant! 0.0
Independent T-Test for Focal2EmpTrainable and Focal2SupTrainable. is signific
ant! 0.011187
Independent T-Test for Focal3EmpTrainable and Focal3SupTrainable. is signific
ant! 0.039233
```

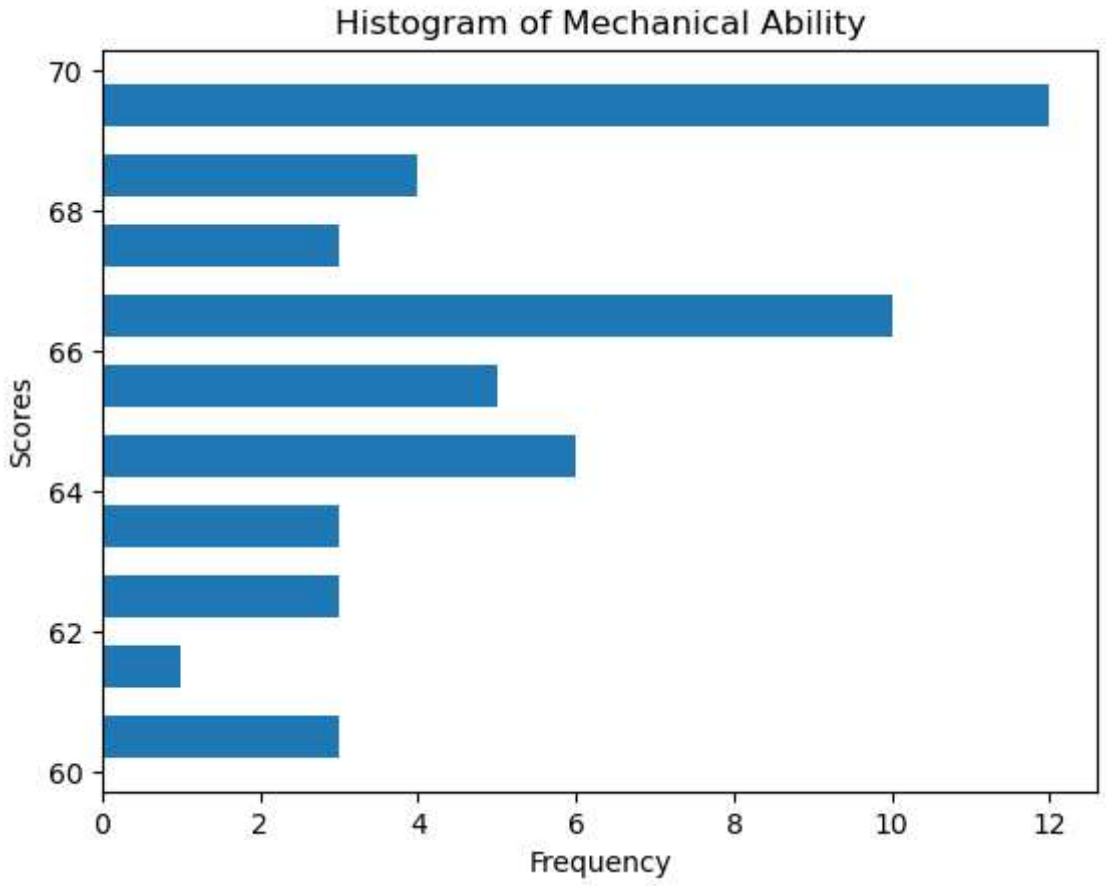
In [78]:
```
#Now we are going to graph the Oral Expressivenss and Mechanical Ability Selec
import matplotlib.pyplot as plt
plt.hist(df["OralExpression"], rwidth=0.6, orientation = "horizontal")
plt.xlabel('Frequency')
plt.ylabel('Scores')
plt.title('Histogram of Oral Expression')
```

Out[78]:
```
Text(0.5, 1.0, 'Histogram of Oral Expression')
```



In [80]:
```
plt.hist(df["MechanicalAbility"], rwidth=0.6, orientation = "horizontal")
plt.xlabel('Frequency')
plt.ylabel('Scores')
plt.title('Histogram of Mechanical Ability')
```

Out[80]: Text(0.5, 1.0, 'Histogram of Mechanical Ability')



In [84]:
```
#Let's also take a look at the max
OEmax = df["OralExpression"].max()
print(f"The max for Oral Expression is {OEmax}, which is {60 - OEmax} points u
MAmax = df["MechanicalAbility"].max()
print(f"The max for Mechanical Ability is {MAmax}, which is {100 - MAmax} poin
```

```
The max for Oral Expression is 45, which is 15 points under a perfect score.
The max for Mechanical Ability is 70, which is 30 points under a perfect scor
e.
```

In [86]:
```
#Let's also look at the averages
OEmean = df["OralExpression"].mean()
print(f"The mean for Oral Expression is {OEmean}.")
MEmean = df["MechanicalAbility"].mean()
print(f"The mean for Mechanical Ability is {MEmean}.")
```

```
The mean for Oral Expression is 41.86.
The mean for Mechanical Ability is 65.88.
```

In [91]:
```
#Lastly, we need to take a look at the three-question Safety Climate Scale. Le
SC1mean = df["SafetyClimate1"].mean()
SC2mean = df["SafetyClimate2"].mean()
SC3mean = df["SafetyClimate3"].mean()
print(f"Safety Climate 1 Mean: {SC1mean}, Safety Climate 2 Mean: {SC2mean}, Sa
print(f"The overall mean is {round((SC1mean + SC2mean + SC3mean)/3, 3)}.")
```

```
Safety Climate 1 Mean: 2.24, Safety Climate 2 Mean: 2.42, Safety Climate 3 Me
an: 2.28
The overall mean is 2.313.
```

In [ ]: