Subjective evaluation of active noise cancellation in headphones

In [ ]:
```
import numpy as np
```

Dummy Data Generation

In [ ]:
```
import random as rnd

samples_quantity = 30
measurements_per_sample = 8
```

$X_1 \equiv$ ANC Status

In [ ]:
```
anc_status = np.empty((samples_quantity, measurements_per_sample))

for i in range(samples_quantity):
    for j in range(0, measurements_per_sample, 4):
        anc_status[i][j] = 0 # Sin ANC
        anc_status[i][j+1] = 0 # Sin ANC
        anc_status[i][j+2] = 1 # Con ANC
        anc_status[i][j+3] = 1 # Con ANC
```

$X_2 \equiv$ Measured Noise

In [ ]:
```
import math

measured_noise_spl = np.empty((samples_quantity, measurements_per_sample)

noise_mean = 65
noise_deviation = 2
background_noise_spl = rnd.gauss(noise_mean, noise_deviation)
delta_spl = 15
generated_noise_spl = noise_mean + 3 * noise_deviation + delta_spl

for i in range(samples_quantity):
    for j in range(measurements_per_sample):
        background_noise_spl = rnd.gauss(noise_mean, noise_deviation)

        if (j % 2 == 0):
            measured_noise_spl[i][j] = background_noise_spl # Solo ruido
        else:
            measured_noise_spl[i][j] = 10 * math.log10(10 ** (background_
```

$Y_1, Y_2 \equiv$ Perceived Noise Level and Sound Quality

In [ ]:
```
perceived_noise = np.empty((samples_quantity, measurements_per_sample))
perceived_quality = np.empty((samples_quantity, measurements_per_sample))
```

```
for i in range(samples_quantity):
    for j in range(0, measurements_per_sample, 4):
        perceived_noise[i][j] = rnd.randint(1, 4) # Sin ANC, poco ruido
        perceived_noise[i][j+1] = rnd.randint(4, 5) # Sin ANC, mucho ruid
        perceived_noise[i][j+2] = rnd.randint(1, 2) # Con ANC, poco ruido
        perceived_noise[i][j+3] = rnd.randint(1, 3) # Con ANC, mucho ruid

        perceived_quality[i][j] = rnd.randint(3, 5) # Sin ANC, poco ruido
        perceived_quality[i][j+1] = rnd.randint(3, 5) # Sin ANC, mucho ru
        perceived_quality[i][j+2] = rnd.randint(2, 5) # Con ANC, poco rui
        perceived_quality[i][j+3] = rnd.randint(1, 4) # Con ANC, mucho ru
```

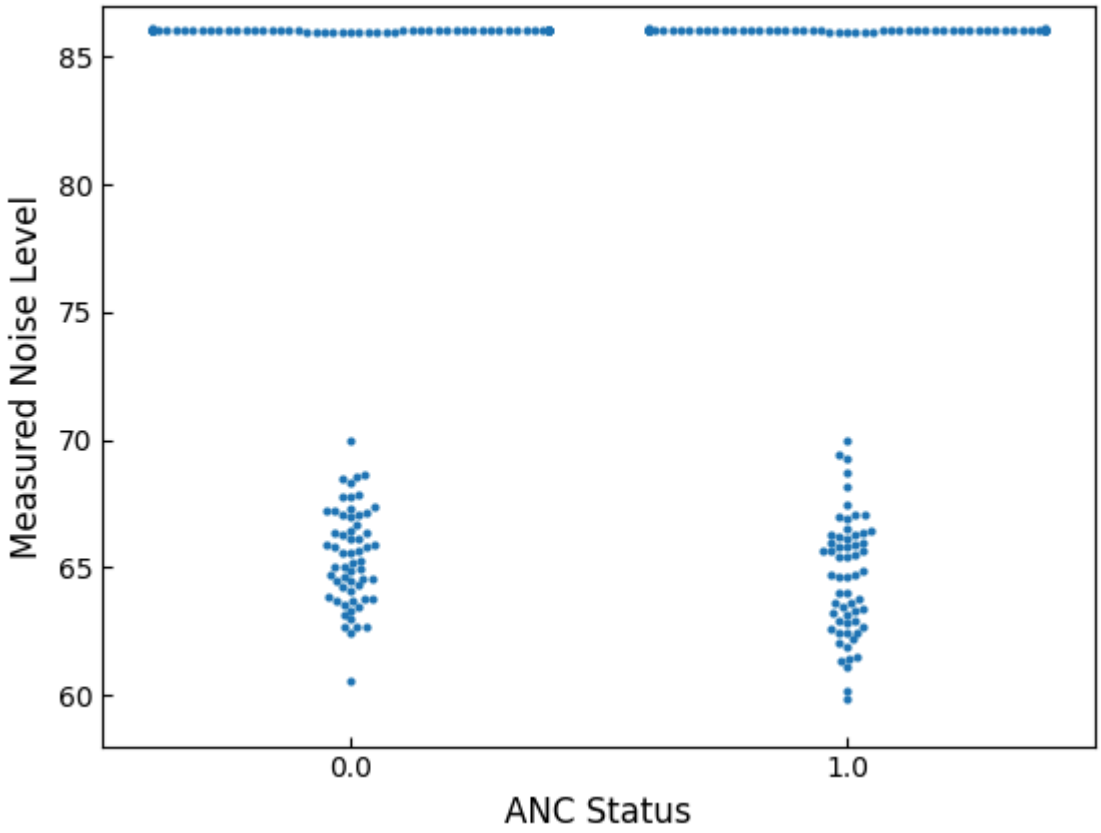
Results Dot Plots

In [ ]:
```
import seaborn as sns
import matplotlib.pyplot as plt

x_1 = anc_status.ravel()
x_2 = measured_noise_spl.ravel()
y_1 = perceived_noise.ravel()
y_2 = perceived_quality.ravel()
```

$X_2 \equiv$ Measured Noise Level Dot Plot

In [ ]:
```
y_min = noise_mean - 3 * noise_deviation - 1
y_max = noise_mean + 3 * noise_deviation + delta_spl + 1

sns.swarmplot(x = x_1, y = x_2, size = 3)
plt.tick_params(axis = 'both', direction = 'in')
plt.ylim(y_min, y_max)
plt.xlabel('ANC Status', fontsize = 12)
plt.ylabel('Measured Noise Level', fontsize = 12)
plt.show()
```

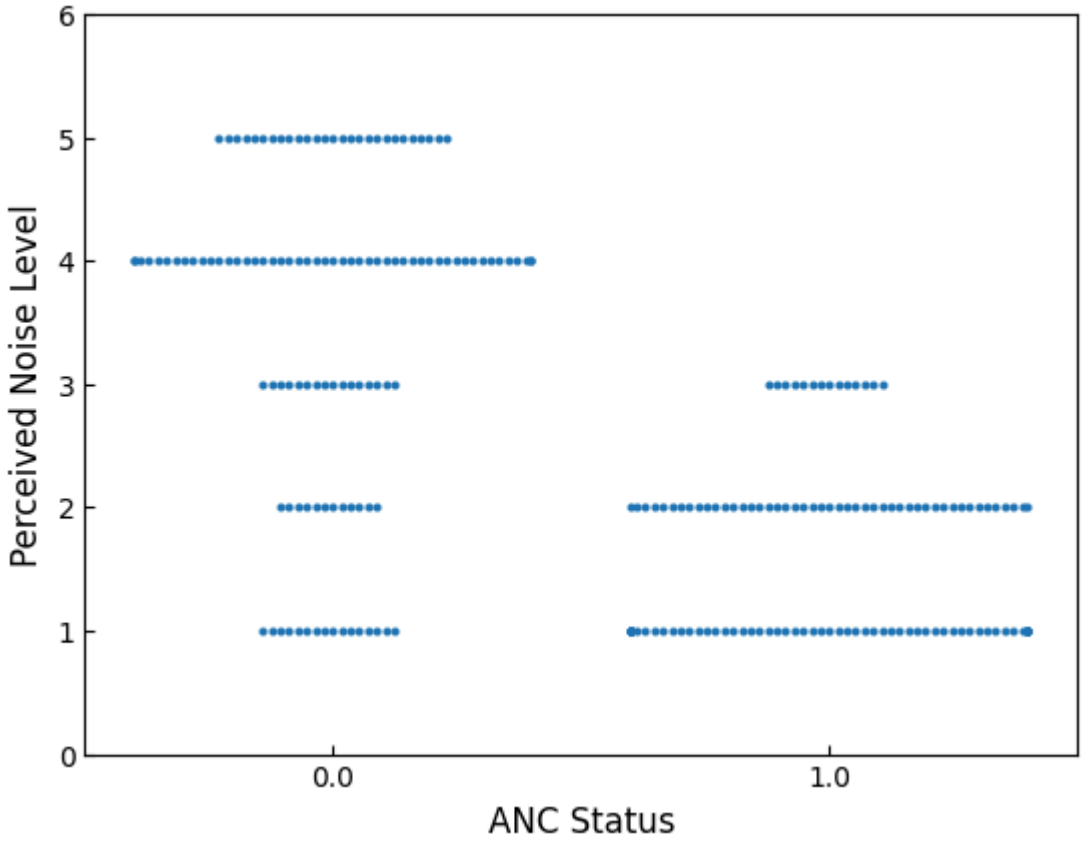
```
/Users/mrmalvicino/Desktop/GitHub/anc/myenv/lib/python3.12/site-package
s/seaborn/categorical.py:3399: UserWarning: 12.5% of the points cannot b
e placed; you may want to decrease the size of the markers or use stripp
lot.
  warnings.warn(msg, UserWarning)
```



$Y_1 \equiv$ Perceived Noise Level Dot Plot

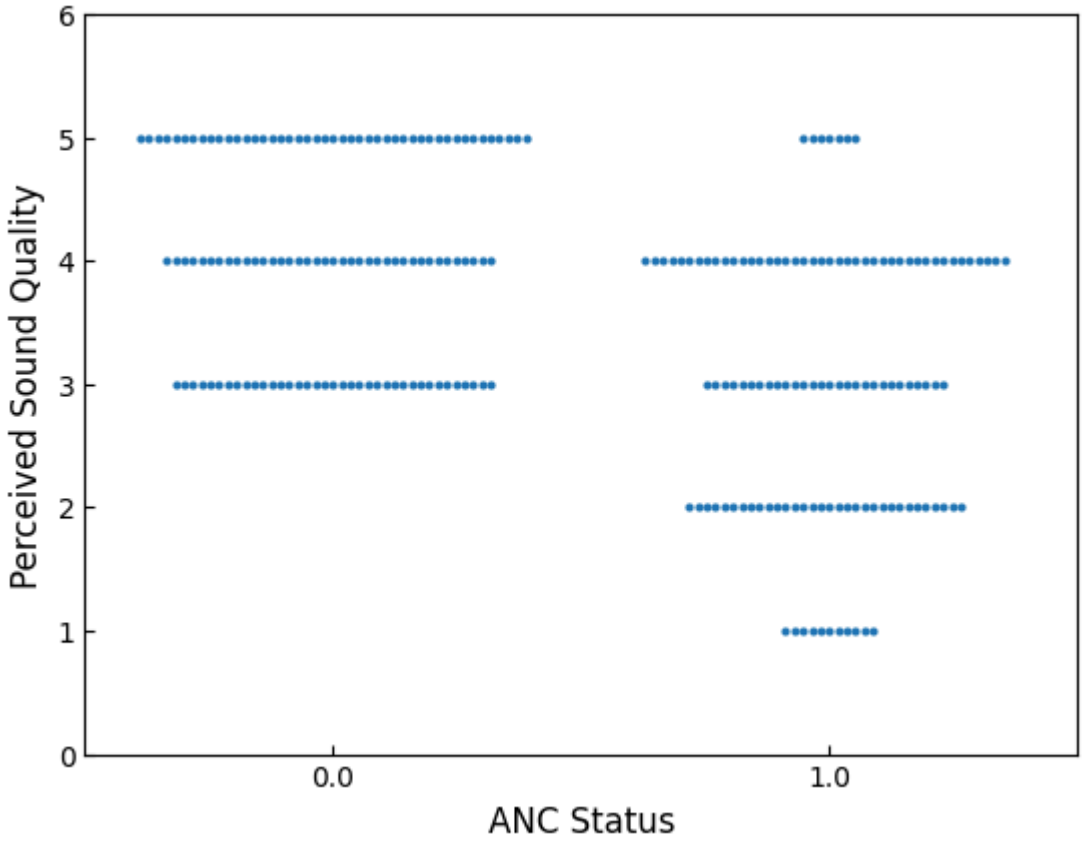
In [ ]:
```
sns.swarmplot(x = x_1, y = y_1, size = 3)
plt.tick_params(axis = 'both', direction = 'in')
plt.ylim(0, 6)
plt.xlabel('ANC Status', fontsize = 12)
plt.ylabel('Perceived Noise Level', fontsize = 12)
plt.show()
```

```
/Users/mrmalvicino/Desktop/GitHub/anc/myenv/lib/python3.12/site-packages/seaborn/categorical.py:3399: UserWarning: 13.3% of the points cannot be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



$Y_2 \equiv$ Perceived Sound Quality Dot Plot

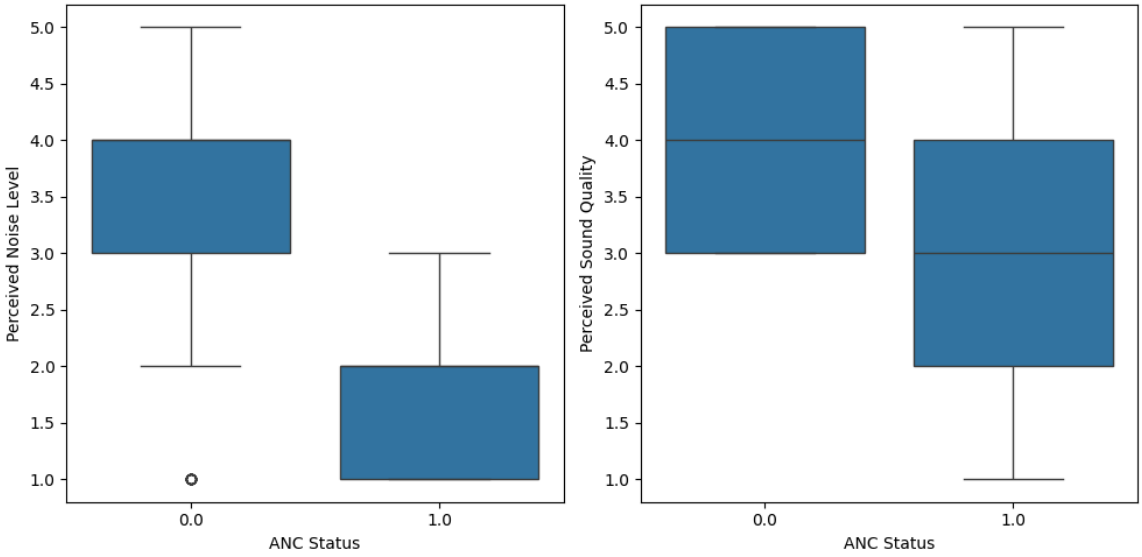
In [ ]:
```
sns.swarmplot(x = x_1, y = y_2, size = 3)
plt.tick_params(axis = 'both', direction = 'in')
plt.ylim(0, 6)
plt.xlabel('ANC Status', fontsize = 12)
plt.ylabel('Perceived Sound Quality', fontsize = 12)
plt.show()
```



Data Validation

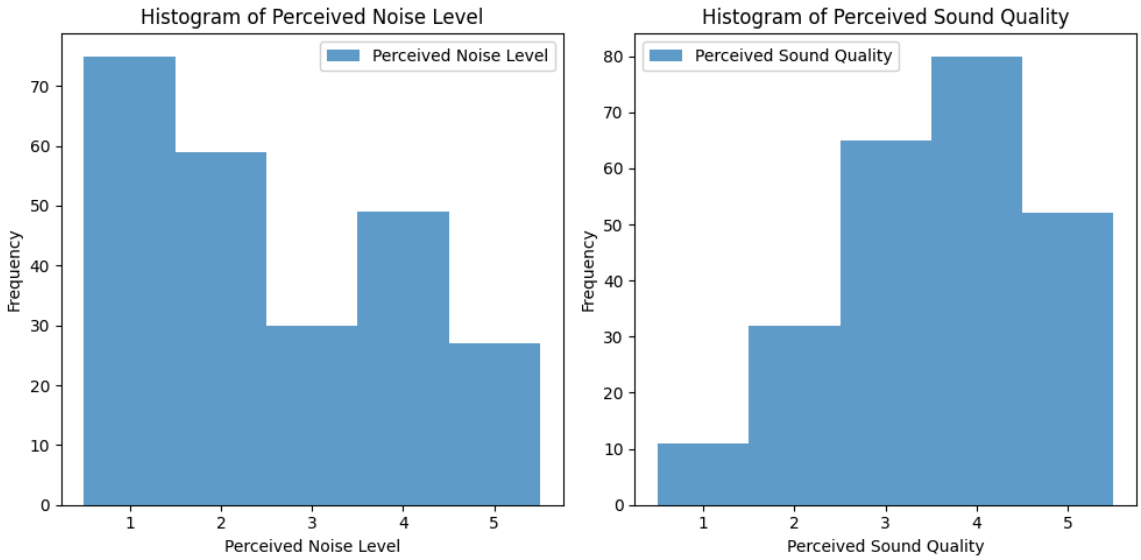
Box Plot

In [ ]:
```
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
sns.boxplot(x = x_1, y = y_1)
plt.xlabel('ANC Status')
plt.ylabel('Perceived Noise Level')

plt.subplot(1, 2, 2)
sns.boxplot(x = x_1, y = y_2)
plt.xlabel('ANC Status')
plt.ylabel('Perceived Sound Quality')

plt.tight_layout()
plt.show()
```



Histogram

In [ ]:
```
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.hist(y_1, bins=np.arange(0.5, 6.5), alpha=0.7, label='Perceived Noise
plt.xlabel('Perceived Noise Level')
plt.ylabel('Frequency')
plt.title('Histogram of Perceived Noise Level')
plt.legend()

plt.subplot(1, 2, 2)
plt.hist(y_2, bins=np.arange(0.5, 6.5), alpha=0.7, label='Perceived Sound
plt.xlabel('Perceived Sound Quality')
plt.ylabel('Frequency')
plt.title('Histogram of Perceived Sound Quality')
plt.legend()

plt.tight_layout()
plt.show()
```



Shapiro-Wilk Normality test

In [ ]:
```
from scipy import stats

statistic, p_value = stats.shapiro(x_2)

print("X2:")
print(f"Statistic: {statistic:.3f}")
print(f"p-value: {p_value:.3f}")

statistic, p_value = stats.shapiro(y_1)

print("\nY1:")
print(f"Statistic: {statistic:.3f}")
print(f"p-value: {p_value:.3f}")

statistic, p_value = stats.shapiro(y_2)

print("\nY2:")
print(f"Statistic: {statistic:.3f}")
print(f"p-value: {p_value:.3f}")
```

```
X2:
Statistic: 0.720
p-value: 0.000

Y1:
Statistic: 0.859
p-value: 0.000

Y2:
Statistic: 0.896
p-value: 0.000
```

Correlation and Regression

In [ ]:
```
import statsmodels.api as sm
```

Scatter Plot and Pearson Correlation for $Y_1(X_1)$

In [ ]:
```
# plt.scatter(x_1, y_1)
# plt.tick_params(axis = 'both', direction = 'in')
# plt.xlim(-1, 2)
# plt.ylim(0, 6)
# plt.xlabel('Variable X1 [ANC Status]', fontsize = 12)
# plt.ylabel('Variable Y1 [Noise score]', fontsize = 12)
# plt.show()

correlation_coefficient, p_value = stats.pearsonr(x_1, y_1)

print(f"r = {correlation_coefficient:.3f}")
print(f"p-value = {p_value:.3f}")
```

```
r = -0.668
p-value = 0.000
```

Scatter Plot and Pearson Correlation for $Y_2(X_1)$

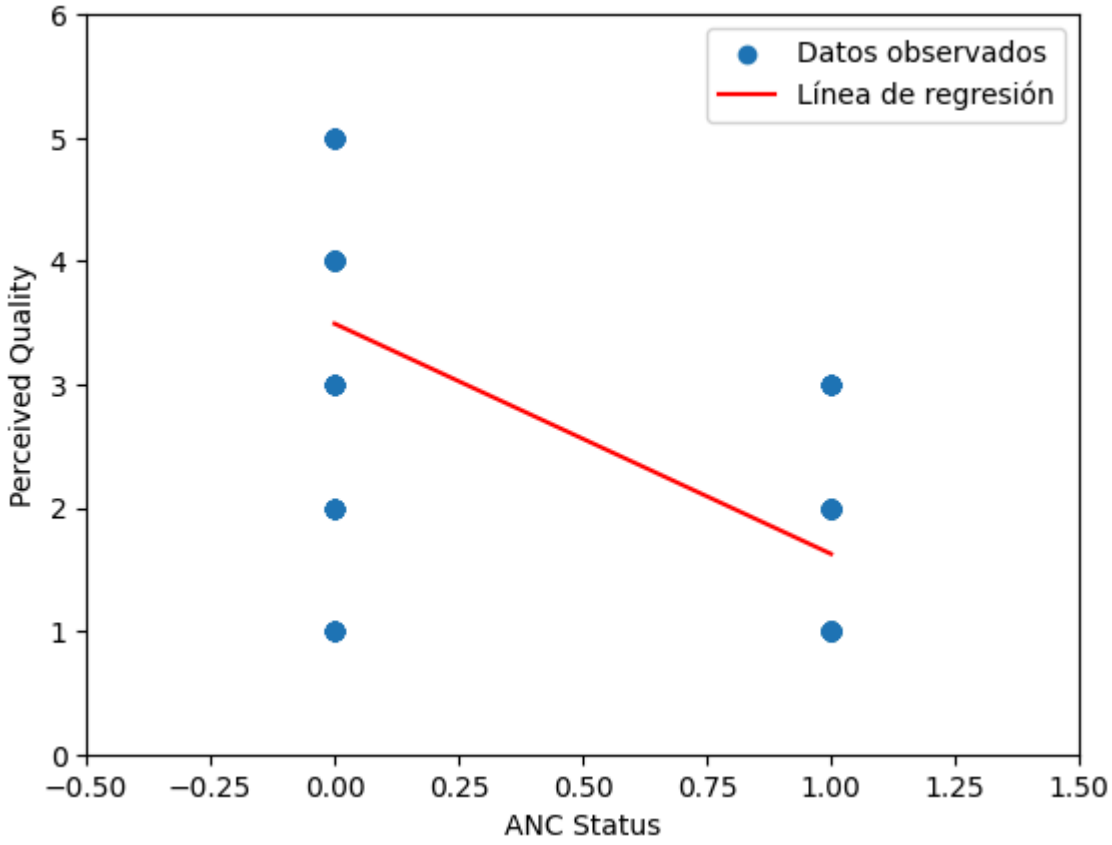
In [ ]:
```
# plt.scatter(x_1, y_2)
# plt.tick_params(axis = 'both', direction = 'in')
# plt.xlim(-1, 2)
# plt.ylim(0, 6)
# plt.xlabel('Variable X1 [ANC Status]', fontsize = 12)
# plt.ylabel('Variable Y2 [Quality score]', fontsize = 12)
# plt.show()

correlation_coefficient, p_value = stats.pearsonr(x_1, y_2)

print(f"r = {correlation_coefficient:.3f}")
print(f"p-value = {p_value:.3f}")
```

```
r = -0.475
p-value = 0.000
```

Linear Regression for $Y_1(X_1)$

In [ ]:
```
x_1_c = sm.add_constant(x_1)
model = sm.OLS(y_1, x_1_c).fit()
print(model.summary())
predictions = model.predict(x_1_c)

plt.scatter(x_1, y_1, label='Datos observados')
plt.plot(x_1, predictions, color='red', label='Línea de regresión')
plt.xlim(-0.5, 1.5)
plt.ylim(0, 6)
plt.xlabel('ANC Status')
plt.ylabel('Perceived Quality')
plt.legend()
plt.show()
```

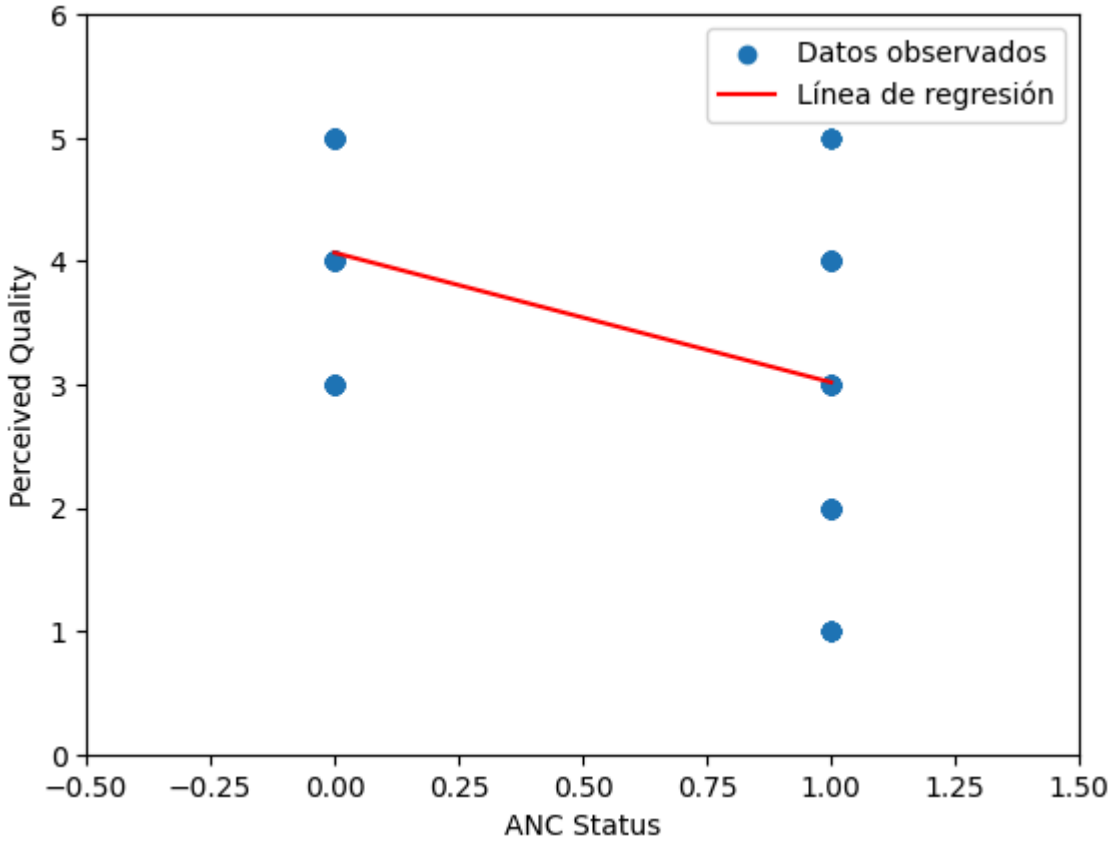
```
                            OLS Regression Results
==============================================================================
======
Dep. Variable:                        y   R-squared:
0.446
Model:                              OLS   Adj. R-squared:
0.443
Method:                   Least Squares   F-statistic:
191.3
Date:                  Fri, 07 Jun 2024   Prob (F-statistic):           2.
52e-32
Time:                          20:09:19   Log-Likelihood:                -
350.20
No. Observations:                   240   AIC:
704.4
Df Residuals:                       238   BIC:
711.4
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
======
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------
const          3.4917      0.095     36.587      0.000       3.304
3.680
x1            -1.8667      0.135    -13.831      0.000      -2.133
-1.601
==============================================================================
======
Omnibus:                       14.070   Durbin-Watson:
2.644
Prob(Omnibus):                  0.001   Jarque-Bera (JB):
15.121
Skew:                          -0.612   Prob(JB):                     0.
000521
Kurtosis:                       3.122   Cond. No.
2.62
==============================================================================
======

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is c
orrectly specified.
```



Linear Regression for $Y_2(X_1)$

In [ ]:
```
x_1_c = sm.add_constant(x_1)
model = sm.OLS(y_2, x_1_c).fit()
print(model.summary())
predictions = model.predict(x_1_c)

plt.scatter(x_1, y_2, label='Datos observados')
plt.plot(x_1, predictions, color='red', label='Línea de regresión')
plt.xlim(-0.5, 1.5)
plt.ylim(0, 6)
plt.xlabel('ANC Status')
plt.ylabel('Perceived Quality')
plt.legend()
plt.show()
```

```
                            OLS Regression Results
==============================================================================
======
Dep. Variable:                        y   R-squared:
0.225
Model:                              OLS   Adj. R-squared:
0.222
Method:                   Least Squares   F-statistic:
69.22
Date:                  Fri, 07 Jun 2024   Prob (F-statistic):           6.
86e-15
Time:                          20:09:19   Log-Likelihood:                -
334.09
No. Observations:                   240   AIC:
672.2
Df Residuals:                       238   BIC:
679.1
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
======
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------
const          4.0667      0.089     45.571      0.000       3.891
4.242
x1            -1.0500      0.126     -8.320      0.000      -1.299
-0.801
==============================================================================
======
Omnibus:                       32.242   Durbin-Watson:
2.236
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
9.837
Skew:                          -0.170   Prob(JB):
0.00731
Kurtosis:                       2.068   Cond. No.
2.62
==============================================================================
======

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is c
orrectly specified.
```


Datos observados
Línea de regresión
Perceived Quality
ANC Status
0
1
2
3
4
5
6
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50