```
import matplotlib.pyplot as plt  # To visualize
import pandas as pd  # To read data
from sklearn.linear_model import LinearRegression
import numpy

def get_filter(dataframe, column_to_filter, filter_value):
    data_filter = dataframe[column_to_filter] == filter_value
    filtered_dataset = dataframe[data_filter]
    return filtered_dataset

def get_linear_regression(dataset, x, y, use_log):
    X = dataset[x].values  # values converts it into a numpy array
    Y = dataset[y].values
    if use_log:
        FX = numpy.zeros(len(X))
        FY = numpy.zeros(len(Y))
        for idx, x in enumerate(X):
            FX[idx] = numpy.log10(float(x))
        for idx, y in enumerate(Y):
            FY[idx] = numpy.log10(y)
        FX2 = numpy.reshape(FX, (len(X),1))
        FY2 = numpy.reshape(FY, (len(Y),1))
    else:
        FX2=X.reshape(-1, 1)
        FY2=Y.reshape(-1, 1)

    linear_regressor = LinearRegression()  # create object for the class
    linear_regressor.fit(FX2, FY2)  # perform linear regression
    Y_pred = linear_regressor.predict(FX2)  # make predictions
    return FX2, FY2, Y_pred, linear_regressor.intercept_, linear_regressor.coef_


def plot_results(datas_and_fits, xlabel, ylabel, title, plot_fit):
    plt.scatter(datas_and_fits[0], datas_and_fits[1])
    if plot_fit: plt.plot(datas_and_fits[0], datas_and_fits[2], color='red')
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.savefig('Plots/'+ title+'.jpg')
    plt.close()

def plot_multiple(datas_and_fits, x_label, y_label, plot_fit):
    for d in datas_and_fits:
        test = d[0]
        plt.scatter(test[0],test[1])
        if plot_fit:
            plt.plot(test[0], test[2], label=d[1], color='red')

    plt.legend()
    plt.title(x_label + ' vs ' + y_label)
    plt.xlabel(x_label)
    plt.ylabel(y_label)

    plt.savefig('Plots/Combined.jpg')
    plt.close()


# Get data from CSV
path = 'Data/vocab_cs_mod.csv'
```

```
dataframe = pd.read_csv(path)

# raw unfiltered
linear_fit = get_linear_regression(dataframe, 'k', 'N', False)
plot_results(linear_fit, 'k', 'N_k', 'Plot of count of words (N_k) appearing k
times', False)

# unfiltered log (bad fit)
log_linear_fit = get_linear_regression(dataframe, 'k', 'N', True)
plot_results(log_linear_fit, 'k', 'N_k', 'Unfiltered log-linear fit of word
frequencies', True)

# filtered log
filtered_dataframe = dataframe.query('k<(10**5)')
filtered_linear_fit = get_linear_regression(filtered_dataframe, 'k', 'N', True)
plot_results(filtered_linear_fit, 'k', 'N_k', 'Filtered log-linear fit of word
frequencies', True)

# mean and s.d
avg = numpy.sum(dataframe['N']* dataframe['k'])/numpy.sum(dataframe['N'])
variance = numpy.sum((dataframe['N']*(dataframe['k']-
avg)**2))/numpy.sum(dataframe['N'])

print('avg: ' + str(avg))
print('sigma: ' + str(numpy.sqrt(variance)))
```