PV degradation rate estimation using ordinary least squares (OLS)

Marios Theristis and Joshua S. Stein, Sandia National Laboratories

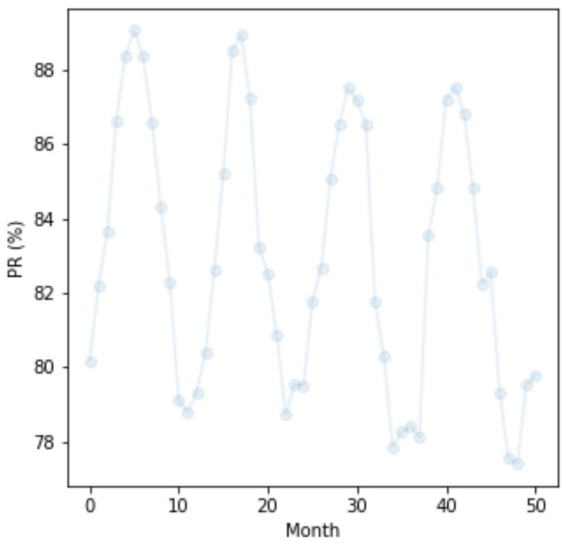
This notebook applies linear regression (LR) using ordinary least squares (OLS) to calculate the degradation rate (DR) on PV performance data.

In [1]:
```
import pandas as pd
import numpy as np
import statsmodels.api as sm
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
```

In [2]:
```
#reading the csv file that contains timeseries with operational and irradiance dat
a. In this example, we import the monthly performance ratio.
df = pd.read_csv(r'C:\...\Sample_data.csv', delimiter = ',' , parse_dates= ['Timest
amp'], dayfirst = False)
```

In [3]:
```
#plot the monthly PRs
fig, axs = plt.subplots(figsize=(5,5))
axs.plot(df.index, df.PR, 'o-', alpha = 0.1)
axs.set_ylabel('PR (%)');
axs.set_xlabel('Month')
```

Out[3]:
```
Text(0.5, 0, 'Month')
```



OLS is estimated using StatsModels. Month (x) and PR (y) need to be converted to NumPy arrays. An intercept (i.e. column of 1s) needs to be added before fitting the model (y = ax + b). Once the intercept (b) and slope (a) are extracted, the absolute and relative DR are calculated as follows:

DR_abs = resolution * slope

DR_rel = 100 *resolution* slope/intercept

Lower and upper confidence intervals are calculated for a confidence level of 95% (i.e. significance level, alpha = 0.05)

In [4]:
```
y = df.PR
x = df.Month
x, y = np.array(x), np.array(y)
```

In [5]:
```
x = sm.add_constant(x)
```

In [6]:
```
model = sm.OLS(y,x)
```

In [7]:
```
results = model.fit()
```

In [8]:
```
results.summary()
```

Out[8]:

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.075
Model:	OLS	**Adj. R-squared:**	0.057
Method:	Least Squares	**F-statistic:**	3.999
Date:	Mon, 16 Dec 2019	**Prob (F-statistic):**	0.0511
Time:	09:48:26	**Log-Likelihood:**	-135.29
No. Observations:	51	**AIC:**	274.6
Df Residuals:	49	**BIC:**	278.4
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	84.6269	0.967	87.525	0.000	82.684	86.570
x1	-0.0666	0.033	-2.000	0.051	-0.134	0.000

Omnibus:	25.217	**Durbin-Watson:**	0.344
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	4.238
Skew:	0.158	**Prob(JB):**	0.120
Kurtosis:	1.623	**Cond. No.**	57.2

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [9]:
```
#if monthly timeseries are used then resolution = 12, whereas daily timeseries shou
ld modify the value below to 365, etc.
resolution = 12
```

In [10]:
```
intercept, slope = results.params
```

In [11]:
```
#confidence level 95%
CI_abs = resolution*results.conf_int(alpha = 0.05)[1]
CIL_abs = CI_abs[1]
CIH_abs = CI_abs[0]
round(CIL_abs,2),round(CIH_abs,2)
```

Out[11]:
```
(0.0, -1.6)
```

In [12]:
```
DR_abs = round(resolution*slope, 2)
DR_abs
```

Out[12]:
```
-0.8
```

In [13]:
```
#confidence level 95%
CI_rel = 100*resolution*results.conf_int(alpha  = 0.05)[1]/intercept
CIL_rel = CI_rel[1]
CIH_rel = CI_rel[0]
round(CIL_rel,2),round(CIH_rel,2)
```

Out[13]:
```
(0.0, -1.89)
```

In [14]:
```
DR_rel = round(100*resolution*slope/intercept,2)
DR_rel
```

Out[14]:
```
-0.95
```

In [15]:
```
#using seaborn to plot the monthly PR and linear regression model fit
sns.regplot(y = df['PR'], x = df['Month'], data = df).set_ylabel("PR (%)")
plt.xlabel("Month")
plt.ylim(75, 90)
plt.title("Relative degradation rate of -0.95%/year on OLS model")
plt.show(fig)
```

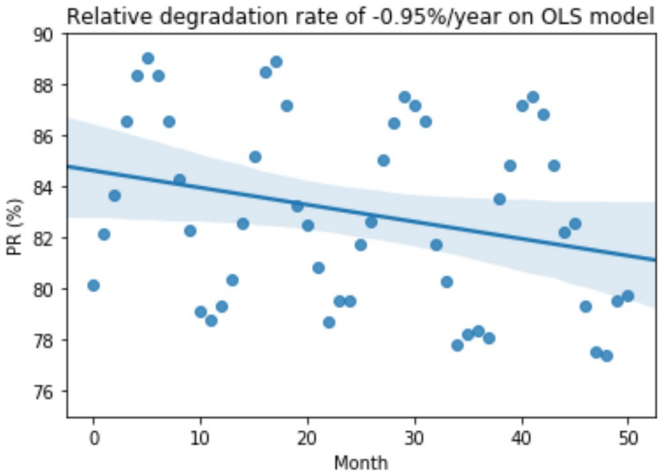


Table of results

In [16]:
```
Results = [['Relative DR', round(DR_rel,2) ], ['Relative CIL', round(CIL_rel,2)],
['Relative CIH', round(CIH_rel,2)], ['Absolute DR', round(DR_abs,2)], ['Absolute CI
L', round(CIL_abs,2)], ['Absolute CIH', round(CIH_abs,2)]]

# Create the pandas DataFrame
Results = pd.DataFrame(Results, columns = ['Parameter', 'Value (%/year)'])

Results
```

Out[16]:

	Parameter	Value (%/year)
0	Relative DR	-0.95
1	Relative CIL	0.00
2	Relative CIH	-1.89
3	Absolute DR	-0.80
4	Absolute CIL	0.00
5	Absolute CIH	-1.60