Linear Regression

Colin Bennie, Caleb Hamblen, Sissi Shen, Justin Yang

When to use Linear Regression?

Strengths:

- Shows relationships between variables
- Relatively simple and easy to interpret
- Baseline for other more complex models
- Great at prediction and model selection

Limitations:

- Struggles with complex relationships(non-linear)
- Multicollinearity
- Sensitive to outliers

Life Expectancy Data

```python
df.columns
```

```
Index(['Country', 'Year_Cohort', 'Life_expectancy', 'Adult_Mortality',
       'infant_deaths', 'Alcohol', 'Hepatitis_B', 'Measles', 'BMI', 'Polio',
       'Total_expenditure', 'Diphtheria', 'HIV_AIDS', 'GDP', 'Population',
       'Schooling', 'region', 'sub_region'],
      dtype='object')
```

```python
df = pd.read_csv('Data/Life_Expectancy_Grouped.csv')
df.head()
```

	Country	Year_Cohort	Life_expectancy	Adult_Mortality	infant_deaths	Alcohol	Hepatitis_B	Measles	BMI	Polio	Total_ex
0	Afghanistan	2000-2003	56.066667	204.666667	87.666667	0.0100	64.000000	4015.333333	13.000000	37.333333	
1	Afghanistan	2004-2007	57.275000	293.500000	84.500000	0.0225	65.000000	1223.250000	14.475000	46.000000	
2	Afghanistan	2008-2011	58.675000	280.500000	75.500000	0.0150	65.250000	2365.500000	16.450000	65.250000	
3	Afghanistan	2012-2015	61.075000	268.500000	65.250000	0.0100	64.500000	1215.750000	18.350000	48.250000	
4	Albania	2000-2003	73.233333	15.666667	1.000000	4.0900	96.333333	14.000000	46.933333	97.333333	

Main Question: Linear Regression

- Given this dataset with multiple predictors and our dependent variable "Life_expectancy", fit a linear regression model and interpret the results.

VIF Check to determine Multicollinearity

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

y, X = dmatrices("Life_expectancy ~  C(region) + C(Year_Cohort) + Adult_Mortality + infant_deaths + Alcohol + \
Hepatitis_B + Measles + BMI + Polio + Total_expenditure + Diphtheria + HIV_AIDS + GDP + Population + Schooling",
                 data=df, return_type='dataframe')

vif = pd.DataFrame()
vif["VIF Factor"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
vif["features"] = X.columns
print(vif)
```

	VIF Factor	features
0	65.997770	Intercept
1	1.572435	C(Year_Cohort)[T.2004-2007]
2	1.662678	C(Year_Cohort)[T.2008-2011]
3	1.810202	C(Year_Cohort)[T.2012-2015]
4	2.517837	C(region)[T.Americas]
5	2.219348	C(region)[T.Asia]
6	4.203763	C(region)[T.Europe]
7	1.634417	C(region)[T.Oceania]
8	2.830165	Adult_Mortality
9	2.542112	infant_deaths
10	2.856947	Alcohol
11	1.727525	Hepatitis_B
12	1.743388	Measles
13	2.306060	BMI
14	3.888790	Polio
15	1.297069	Total_expenditure
16	4.030447	Diphtheria
17	2.120223	HIV_AIDS
18	1.624895	GDP
19	1.740964	Population
20	3.218422	Schooling

Code for Multiple Regression and output

```
model_full = smf.ols(
    'Life_expectancy ~ C(region) + C(Year_Cohort) + Adult_Mortality + infant_deaths + Alcohol + Hepatitis_B + Measles + \
    BMI + Polio + Total_expenditure + Diphtheria + HIV_AIDS + GDP + Population + Schooling',
    data = df).fit()
model_full.summary()
```

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.879
Model:	OLS	**Adj. R-squared:**	0.876
Method:	Least Squares	**F-statistic:**	240.1
Date:	Sun, 08 Oct 2023	**Prob (F-statistic):**	1.07e-286
Time:	18:20:42	**Log-Likelihood:**	-1750.4
No. Observations:	680	**AIC:**	3543.
Df Residuals:	659	**BIC:**	3638.
Df Model:	20		
Covariance Type:	nonrobust		

Multiple Regression Interpretation

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	53.4288	1.005	53.181	0.000	51.456	55.402
C(Year_Cohort)[T.2004-2007]	0.0472	0.358	0.132	0.895	-0.656	0.750
C(Year_Cohort)[T.2008-2011]	0.3139	0.368	0.852	0.394	-0.409	1.037
C(Year_Cohort)[T.2012-2015]	0.4880	0.384	1.270	0.205	-0.267	1.243
C(region)[T.Americas]	4.2379	0.502	8.442	0.000	3.252	5.224
C(region)[T.Asia]	2.9562	0.418	7.079	0.000	2.136	3.776
C(region)[T.Europe]	3.8057	0.627	6.069	0.000	2.574	5.037
C(region)[T.Oceania]	2.0010	0.672	2.978	0.003	0.682	3.320
Adult_Mortality	-0.0259	0.002	-12.924	0.000	-0.030	-0.022
infant_deaths	-0.0009	0.002	-0.546	0.585	-0.004	0.002
Alcohol	-0.0886	0.053	-1.681	0.093	-0.192	0.015
Hepatitis_B	-0.0235	0.008	-2.927	0.004	-0.039	-0.008
Measles	-2.044e-05	1.85e-05	-1.103	0.271	-5.68e-05	1.6e-05
BMI	0.0351	0.011	3.232	0.001	0.014	0.056
Polio	0.0549	0.013	4.102	0.000	0.029	0.081
Total_expenditure	0.0388	0.066	0.585	0.559	-0.091	0.169
Diphtheria	0.0418	0.013	3.187	0.002	0.016	0.068
HIV_AIDS	-0.3242	0.041	-7.947	0.000	-0.404	-0.244
GDP	9.294e-05	1.37e-05	6.802	0.000	6.61e-05	0.000
Population	-1.086e-10	3.27e-09	-0.033	0.974	-6.53e-09	6.31e-09
Schooling	0.8390	0.069	12.093	0.000	0.703	0.975

Model: Life expectancy = 53.4228 + $0.0472 * \text{Year\_Cohort}_{2004\text{-}2007}$ + $0.3139 * \text{Year\_Cohort}_{2008\text{-}2011}$ + ...+0.8390*Schooling

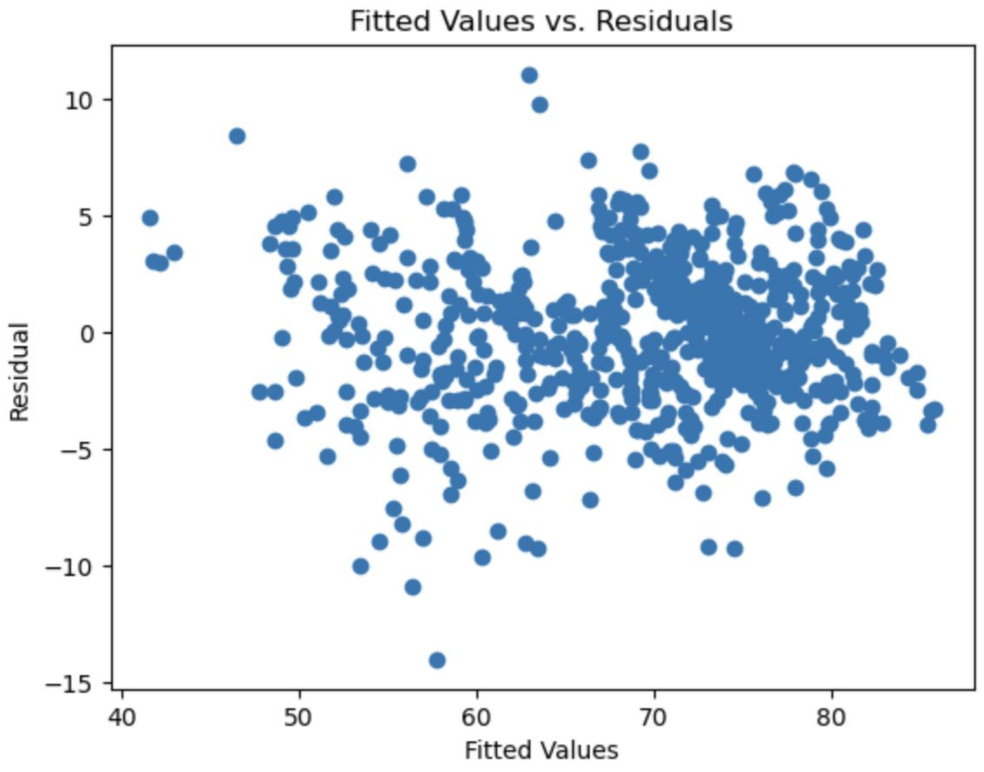
- Baseline levels: Year_Cohort 2000-2003, region: Africa
- P-values ≤ 0.05

Follow Up Question: Model Diagnostic Checks

- Given the model, how can we assess if it's a good model? What are some ways we can improve the strength and reliability of the model?

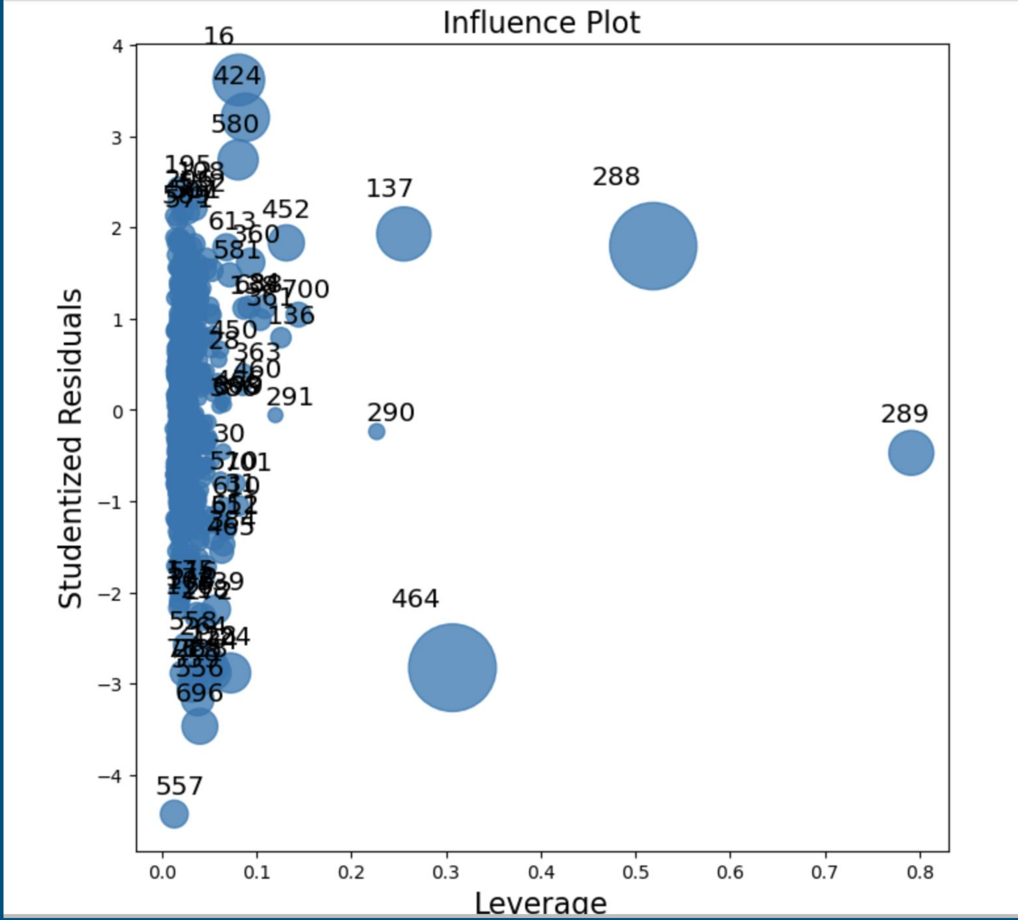
Heteroscedasticity

```
p = model_full.fittedvalues
res = model_full.resid
plt.scatter(p,res)
plt.xlabel("Fitted Values")
plt.ylabel("Residual")
plt.title("Fitted Values vs. Residuals")
```



Influential Points

```python
# Influence Point
infl = model_full.get_influence()
fig, ax = plt.subplots(figsize=(8,8))
fig=sm.graphics.influence_plot(
    model_full, ax=ax, criterion="cooks")
```



Influential Points

```python
n = df.shape[0]
p = len(model_full.params)

# External Studentized Residuals
seuil_stud = scipy.stats.t.ppf(0.975,df=n-p-1)
reg_studs=infl.resid_studentized_external
atyp_stud = np.abs(reg_studs) > seuil_stud
df_resid = pd.DataFrame({"index": df.index[atyp_stud], "resid": reg_studs[atyp_stud]})

# Cook's distance
inflsum=infl.summary_frame()
reg_cook=inflsum.cooks_d
atyp_cook = np.abs(reg_cook) >= 4/n
df_cook = pd.DataFrame({"index": df.index[atyp_cook], "cook": reg_cook[atyp_cook]})
```

```python
infl_cook = pd.merge(
    df_resid, df_cook,
    on="index", how="inner")
infl_cook
```

	index	resid	cook
0	16	3.619346	0.054354
1	76	-2.887665	0.008873
2	108	2.339580	0.009238
3	124	-2.884458	0.030792
4	128	-2.814932	0.020737
5	218	-2.250043	0.009279

Fit the model again

```
# Delete the points that are high in both criterion
del_index = list(infl_cook['index'])
df_final = df.drop(del_index)

# Fit the model again
model_full_new = smf.ols(
    'Life_expectancy ~ C(region) + C(Year_Cohort) + Adult_Mortality + infant_deaths + Alcohol + Hepatitis_B + \
    Measles + BMI + Polio + Total_expenditure + Diphtheria + HIV_AIDS + GDP + Population + Schooling',
    data = df_final).fit()
model_full_new.summary()
```

Model Interpretation

OLS Regression Results

Dep. Variable:	Life_expectancy	**R-squared:**	0.908
Model:	OLS	**Adj. R-squared:**	0.905
Method:	Least Squares	**F-statistic:**	325.8
Date:	Mon, 09 Oct 2023	**Prob (F-statistic):**	0.00
Time:	15:27:58	**Log-Likelihood:**	-1655.5
No. Observations:	682	**AIC:**	3353.
Df Residuals:	661	**BIC:**	3448.
Df Model:	20		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	56.3511	0.942	59.836	0.000	54.502	58.200
C(region)[T.Americas]	3.7125	0.441	8.419	0.000	2.847	4.578
C(region)[T.Asia]	2.3394	0.365	6.406	0.000	1.622	3.057
C(region)[T.Europe]	3.4130	0.545	6.259	0.000	2.342	4.484
C(region)[T.Oceania]	2.0480	0.599	3.421	0.001	0.872	3.224
C(Year_Cohort)[T.2004-2007]	-0.0255	0.309	-0.082	0.934	-0.633	0.582
C(Year_Cohort)[T.2008-2011]	0.4537	0.317	1.432	0.153	-0.168	1.076
C(Year_Cohort)[T.2012-2015]	0.5176	0.327	1.584	0.114	-0.124	1.159
Adult_Mortality	-0.0325	0.002	-17.048	0.000	-0.036	-0.029
infant_deaths	-0.0019	0.001	-1.336	0.182	-0.005	0.001
Alcohol	-0.0734	0.045	-1.621	0.105	-0.162	0.016
Hepatitis_B	-0.0163	0.007	-2.334	0.020	-0.030	-0.003
Measles	1.028e-05	1.85e-05	0.557	0.578	-2.6e-05	4.65e-05
BMI	0.0307	0.009	3.243	0.001	0.012	0.049
Polio	0.0456	0.012	3.963	0.000	0.023	0.068
Total_expenditure	0.0297	0.055	0.538	0.591	-0.079	0.138
Diphtheria	0.0366	0.011	3.235	0.001	0.014	0.059
HIV_AIDS	-0.2351	0.038	-6.112	0.000	-0.311	-0.160
GDP	7.952e-05	1.16e-05	6.826	0.000	5.66e-05	0.000
Population	-3.641e-10	2.83e-09	-0.129	0.898	-5.92e-09	5.19e-09
Schooling	0.7845	0.062	12.680	0.000	0.663	0.906

Link to Github

https://github.com/cwbennie/comms_code_demo23