Topic 5 HW

Zander Bonnet

3/20/2024

References:

Cardiovascular Disease dataset. (2019). Kaggle [Dataset]. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset/data.

COVID-19 in USA. (2021). Kaggle [Dataset]. https://www.kaggle.com/datasets/sudalairajkumar/covid19-in-usa?select=us_covid19_daily.csv.

Rogel-Salazar, J. (2023). Statistics and Data Visualization with Python. CRC Press.

Video : https://vimeo.com/925661752/efa69890a1?share=copy

In [1]:

```
import pandas as pd
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
import math
from sklearn.kernel_ridge import KernelRidge
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [2]:

```
data = pd.read_csv('/Users/zanderbonnet/Desktop/GCU/DSC_510/DataSets/cardio.csv'
#age in days
#height in cm
#weight in kg
#gender 1 = m, 2 = f
data.head()
```

Out[2]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

In [3]:

```
age = list(data['age'] / 365) # make age years
gender = [] #sets gender to strings
for gen in data['gender']:
    if gen == 1:
        gender.append("Male")
    elif gen == 2:
        gender.append("Female")
    else:
        gender.append('Missing')
height = list(data['height']) #get height
weight = list(data['weight']) #gets weight

hwa = pd.DataFrame({'Gender':gender,
                   'Age':age,
                   'Height':height,
                   'Weight':weight})
hwa.head()
```

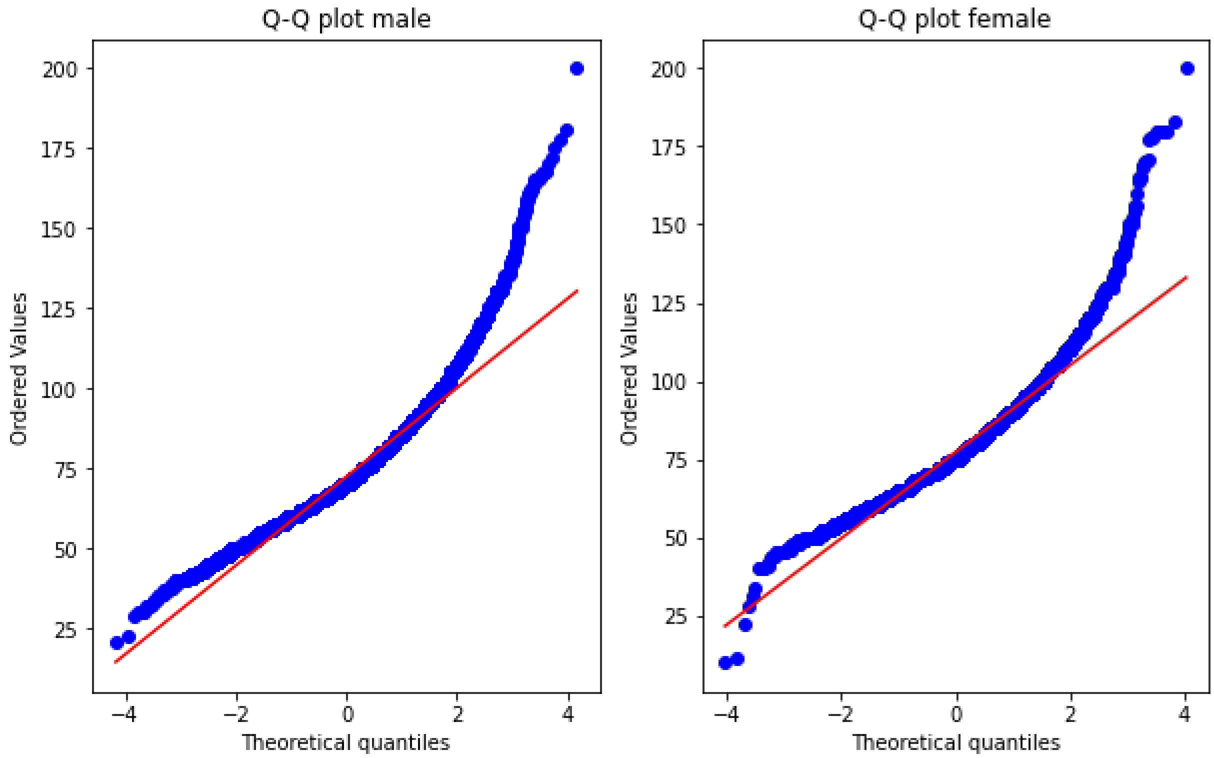
Out[3]:

	Gender	Age	Height	Weight
0	Female	50.391781	168	62.0
1	Male	55.419178	156	85.0
2	Male	51.663014	165	64.0
3	Female	48.282192	169	82.0
4	Male	47.873973	156	56.0

1. Hypothesis

In [4]:

```
#checks normality of variables
fig, (ax1,ax2) = plt.subplots(1,2, figsize = (10,6))
stats.probplot(hwa.loc[hwa['Gender'] == 'Male']['Weight'], dist = 'norm', plot =
ax1.title.set_text('Q-Q plot male')
stats.probplot(hwa.loc[hwa['Gender'] == 'Female']['Weight'], dist = 'norm', plot
ax2.title.set_text('Q-Q plot female')
plt.show()
```



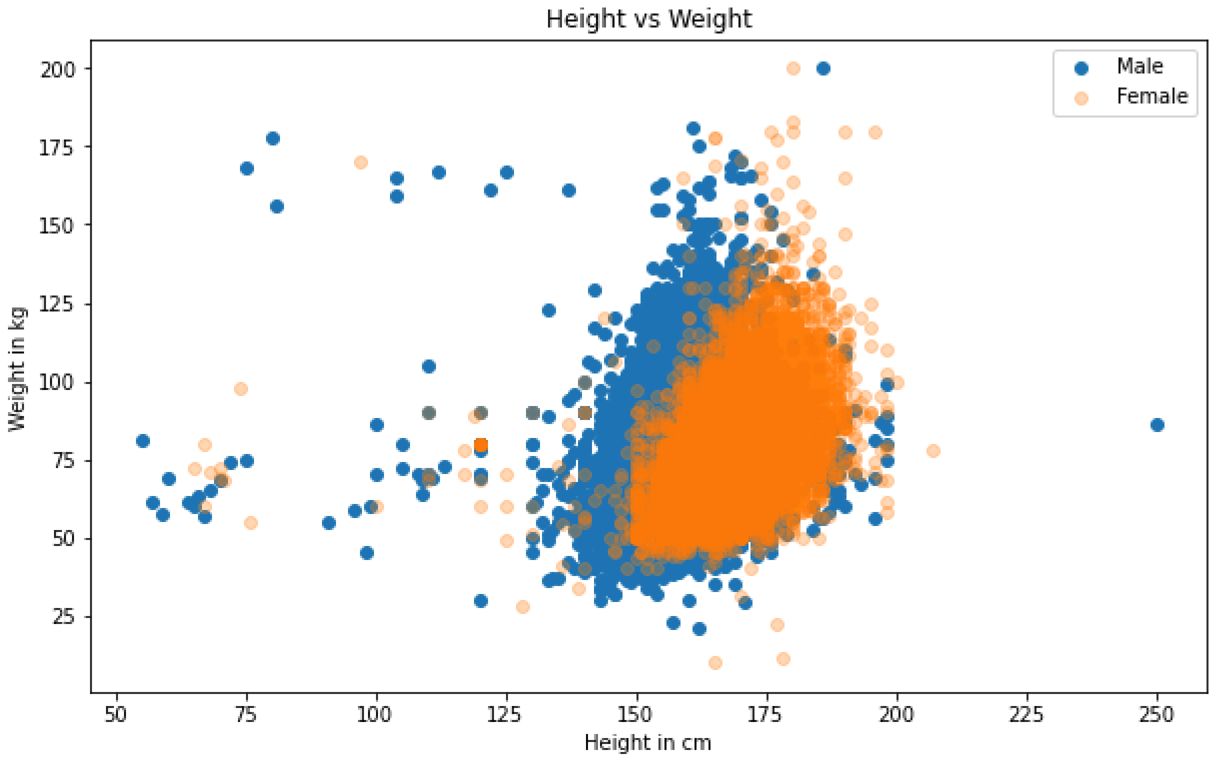
In [5]:

```
#calculates the statistic
mal = hwa.loc[hwa['Gender'] == 'Male']['Weight']
fem = hwa.loc[hwa['Gender'] == 'Female']['Weight']
tstat, pval = stats.ttest_ind(mal,fem)
if pval < .05: print(pval,'< 0.05: P-Value is Significant')
else: print(pval, '> 0.05: P-Value is Not Significant')
print(mal.mean(),fem.mean())
```

```
0.0 < 0.05: P-Value is Significant
72.56560509554139 77.25730690641602
```

In [6]:

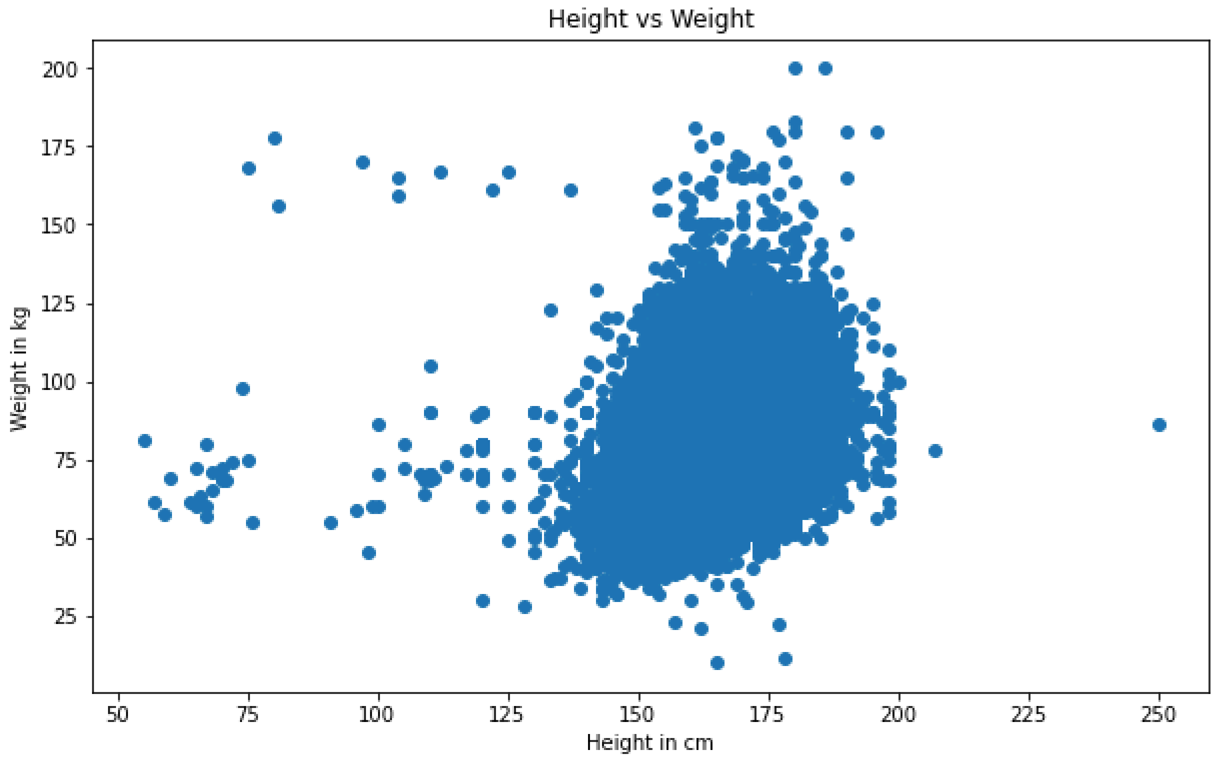
```
#Visualize the data
hei = hwa.loc[hwa['Gender'] == 'Male']['Height']
wei = hwa.loc[hwa['Gender'] == 'Male']['Weight']
plt.figure(figsize=(10,6))
plt.scatter(hei,wei, label = 'Male', )

hei = hwa.loc[hwa['Gender'] == 'Female']['Height']
wei = hwa.loc[hwa['Gender'] == 'Female']['Weight']
plt.scatter(hei,wei, label = 'Female', alpha = .3)

plt.xlabel('Height in cm')
plt.ylabel('Weight in kg')
plt.title('Height vs Weight')
plt.legend()
plt.show()
```



2. Correlation Coefficient

In [7]:

```
hei = hwa['Height']
wei = hwa['Weight']
plt.figure(figsize=(10,6))
plt.scatter(hei,wei)
plt.xlabel('Height in cm')
plt.ylabel('Weight in kg')
plt.title('Height vs Weight')
plt.show()
cor, pval = stats.pearsonr(hei,wei) #calculates the coef
print('Correlation coefficent: {:.3f}'.format(cor))
if pval < .05: print(pval,'< 0.05: P-Value is Significant')
else: print(pval, '> 0.05: P-Value is Not Significant')
```




```
Correlation coefficent: 0.291
0.0 < 0.05: P-Value is Significant
```

3. Linear Regression

In [8]:

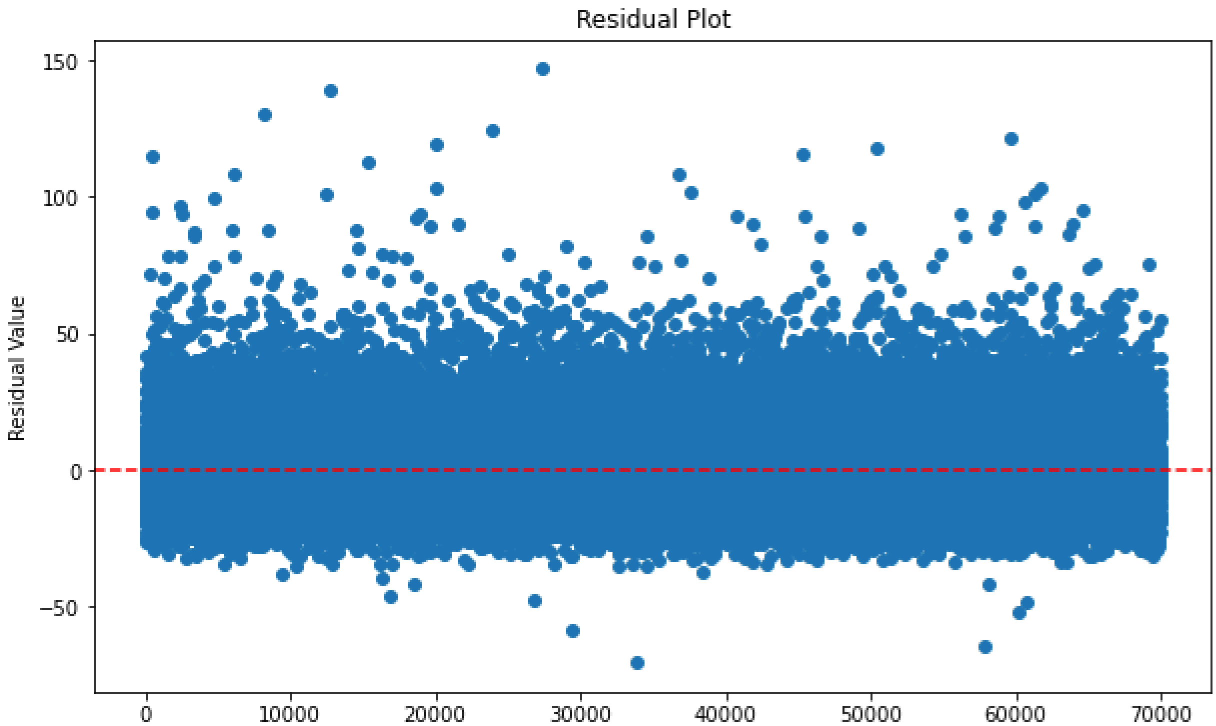
```python
results = smf.ols('Weight ~ Height', data = hwa).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Weight   R-squared:                       0.085
Model:                            OLS   Adj. R-squared:                  0.085
Method:                 Least Squares   F-statistic:                     6474.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):               0.00
Time:                        18:23:08   Log-Likelihood:            -2.8291e+05
No. Observations:               70000   AIC:                         5.658e+05
Df Residuals:                   69998   BIC:                         5.659e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -9.6483      1.043     -9.247      0.000     -11.693      -7.603
Height         0.5102      0.006     80.463      0.000       0.498       0.523
==============================================================================
Omnibus:                    16692.322   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            53181.354
Skew:                           1.215   Prob(JB):                         0.00
Kurtosis:                       6.511   Cond. No.                     3.30e+03
==============================================================================

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

[2] The condition number is large, 3.3e+03. This might indicate that there are strong multicollinearity or other numerical problems.

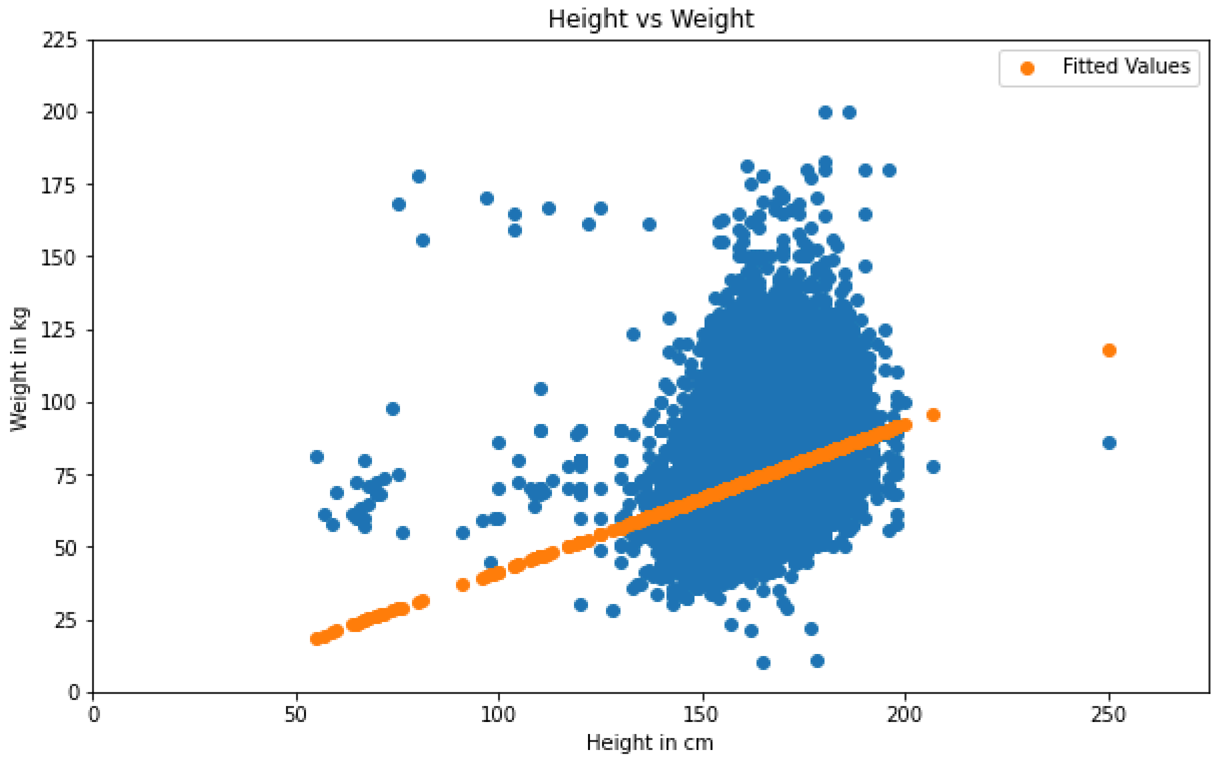
In [9]:

```
plt.figure(figsize=(10,6))
plt.scatter(hwa.index,results.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



In [10]:

```
hei = hwa['Height']
wei = hwa['Weight']

plt.figure(figsize=(10,6))
plt.scatter(hei,wei)
plt.xlabel('Height in cm')
plt.ylabel('Weight in kg')
plt.title('Height vs Weight')
plt.xlim(0,275)
plt.ylim(0,225)

plt.scatter(hwa['Height'],results.fittedvalues, label = 'Fitted Values')
plt.legend()
plt.show()
```



In [11]:

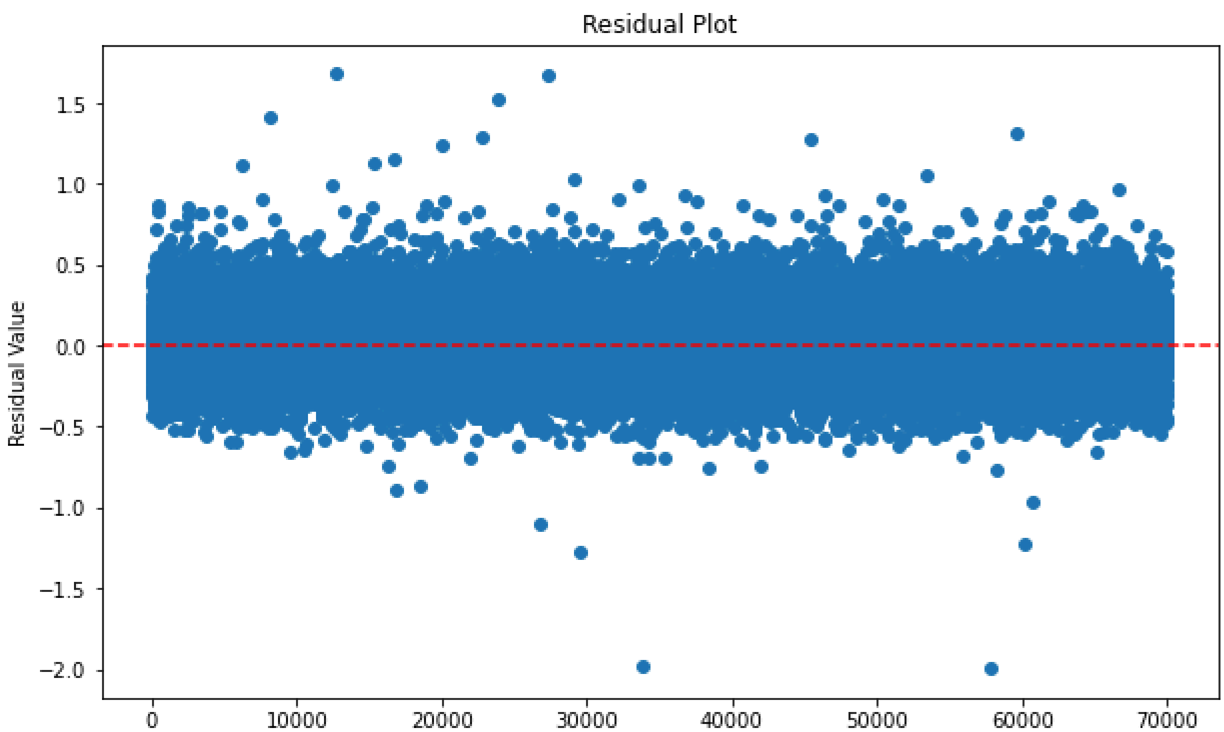
```
#takes the log of the numerical variables
loghwa = hwa.copy()
loghwa['Age'] = loghwa['Age'].apply(math.log)
loghwa['Height'] = loghwa['Height'].apply(math.log)
loghwa['Weight'] = loghwa['Weight'].apply(math.log)

results1 = smf.ols('Weight ~ Height', data = loghwa).fit()
print(results1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Weight   R-squared:                       0.089
Model:                            OLS   Adj. R-squared:                  0.089
Method:                 Least Squares   F-statistic:                     6857.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):               0.00
Time:                        18:23:09   Log-Likelihood:                 21256.
No. Observations:               70000   AIC:                        -4.251e+04
Df Residuals:                   69998   BIC:                        -4.249e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.2136      0.066    -18.262      0.000      -1.344      -1.083
Height         1.0788      0.013     82.807      0.000       1.053       1.104
==============================================================================
Omnibus:                     4861.571   Durbin-Watson:                   1.990
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            10879.145
Skew:                           0.451   Prob(JB):                         0.00
Kurtosis:                       4.708   Cond. No.                         521.
==============================================================================

Notes:
```

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

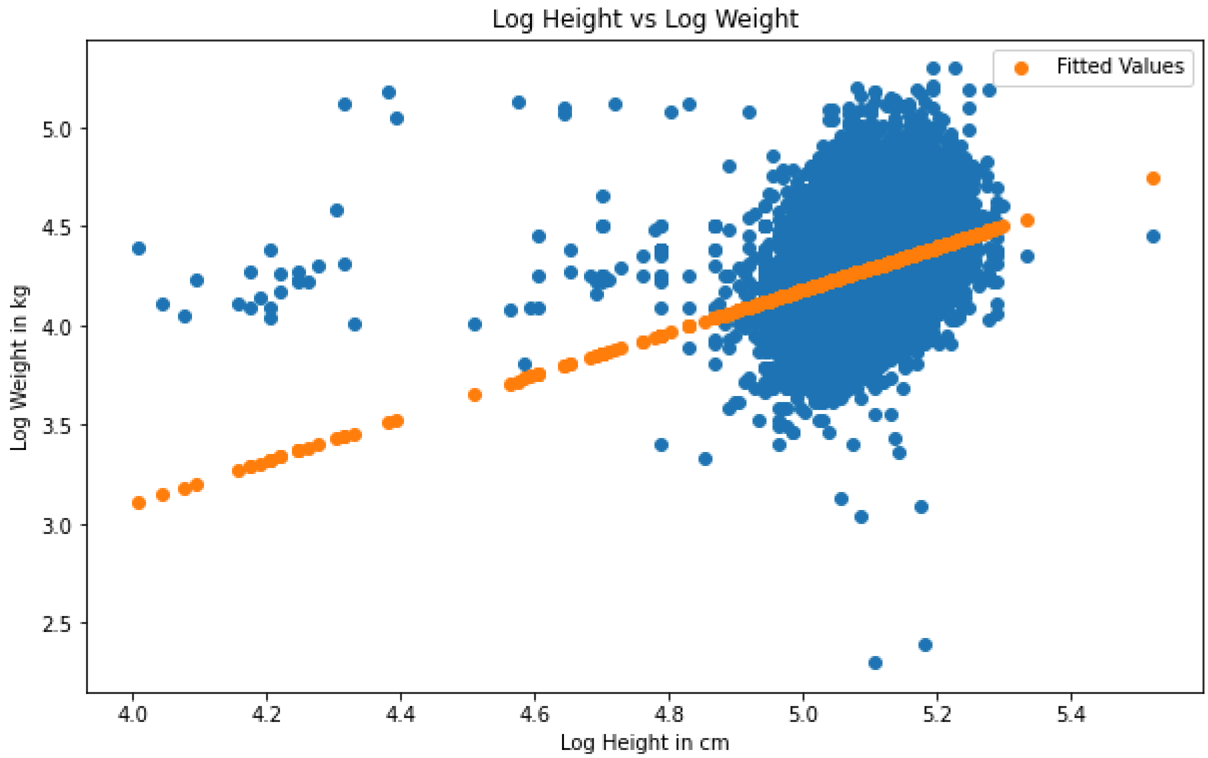
In [12]:

```
plt.figure(figsize=(10,6))
plt.scatter(loghwa.index,results1.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



In [13]:

```
hei = loghwa['Height']
wei = loghwa['Weight']

plt.figure(figsize=(10,6))
plt.scatter(hei,wei)
plt.xlabel('Log Height in cm')
plt.ylabel('Log Weight in kg')
plt.title('Log Height vs Log Weight')
plt.scatter(loghwa['Height'],results1.fittedvalues, label = 'Fitted Values')
plt.legend()
plt.show()
```



4. Multiple Regression Model

In [14]:

```
results2 = smf.ols('Weight ~ Height + Age', data = hwa).fit()
print(results2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Weight   R-squared:                       0.091
Model:                            OLS   Adj. R-squared:                  0.091
Method:                 Least Squares   F-statistic:                     3491.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):               0.00
Time:                        18:23:10   Log-Likelihood:            -2.8268e+05
No. Observations:               70000   AIC:                         5.654e+05
Df Residuals:                   69997   BIC:                         5.654e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -20.3302      1.152    -17.646      0.000     -22.588     -18.072
Height         0.5213      0.006     82.217      0.000       0.509       0.534
Age            0.1659      0.008     21.547      0.000       0.151       0.181
==============================================================================
Omnibus:                    17110.550   Durbin-Watson:                   1.993
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            56413.600
Skew:                           1.233   Prob(JB):                         0.00
Kurtosis:                       6.641   Cond. No.                     3.84e+03
==============================================================================

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

[2] The condition number is large, 3.84e+03. This might indicate that there are strong multicollinearity or other numerical problems.

In [15]:

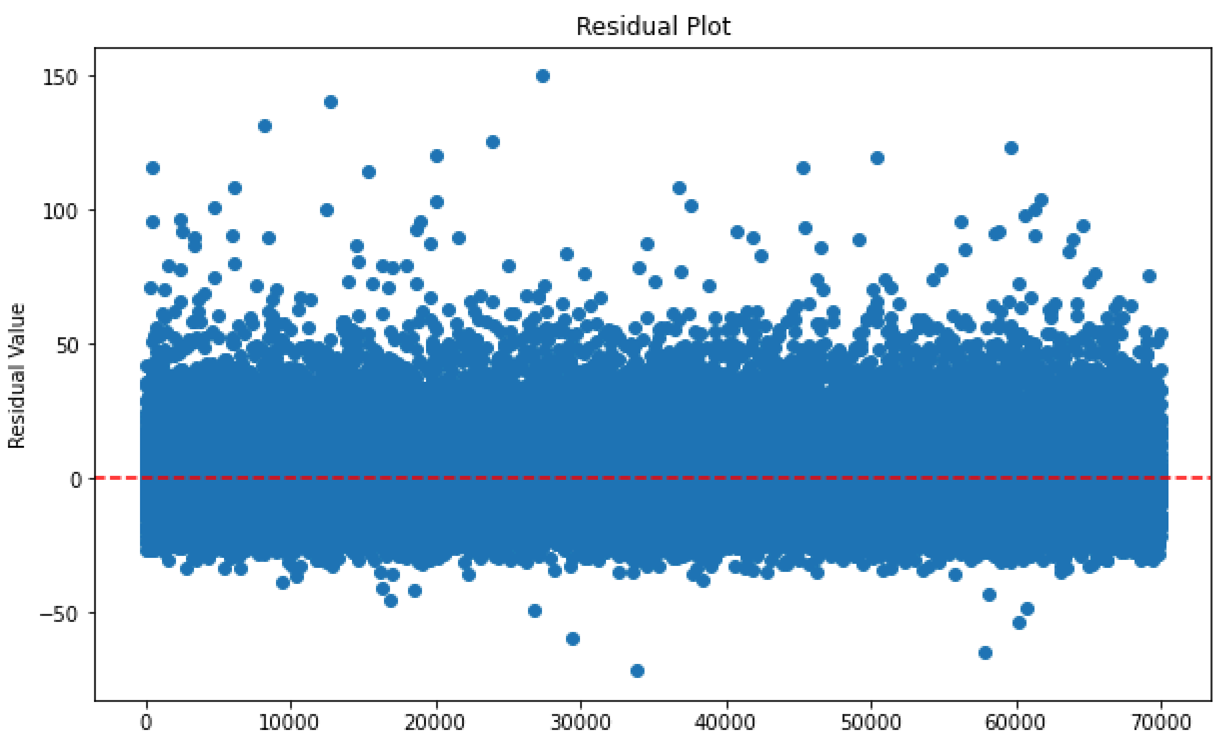
```
X = hwa[['Height', 'Weight', 'Age']]

# VIF dataframe
vif_data = pd.DataFrame()
vif_data["feature"] = X.columns

# calculating VIF for each feature

vif_data["VIF"] = [variance_inflation_factor(X.values, i) for i in range(len(X.c
vif_data
```

Out[15]:

	feature	VIF
0	Height	76.760870
1	Weight	30.186875
2	Age	52.074542

In [16]:

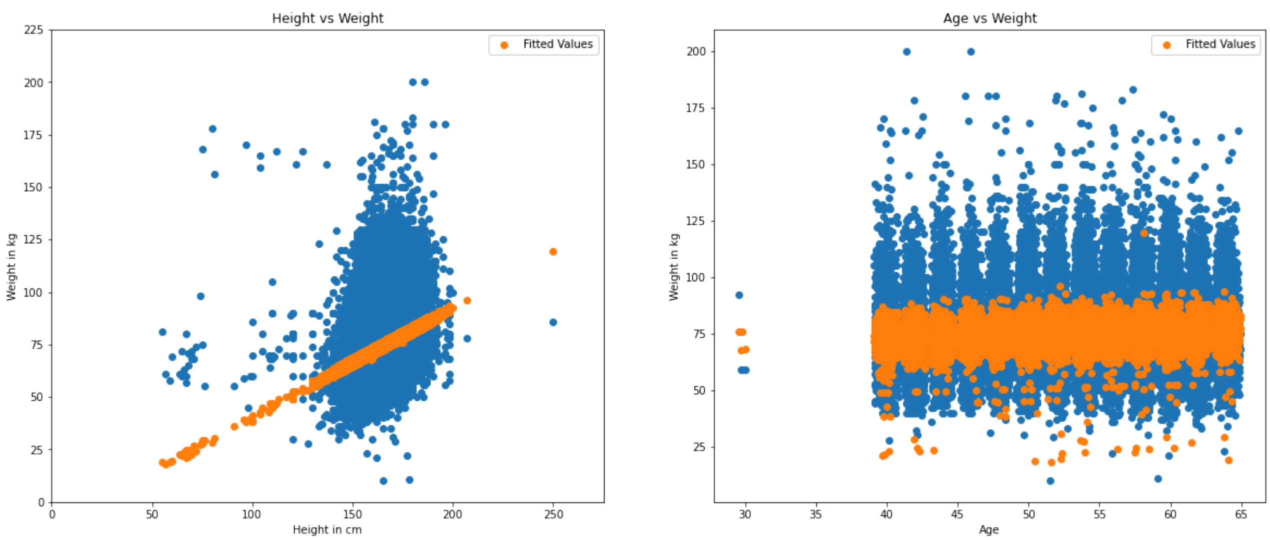
```
plt.figure(figsize=(10,6))
plt.scatter(hwa.index,results2.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



In [17]:

```
hei = hwa['Height']
wei = hwa['Weight']
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
ax1.scatter(hei,wei)
ax1.set_xlabel('Height in cm')
```

```python
ax1.set_ylabel('Weight in kg')
ax1.set_title('Height vs Weight')
ax1.set_xlim(0,275)
ax1.set_ylim(0,225)
ax1.scatter(hwa['Height'],results2.fittedvalues, label = 'Fitted Values')
ax1.legend()


ax2.scatter(hwa['Age'],wei)
ax2.set_xlabel('Age')
ax2.set_ylabel('Weight in kg')
ax2.set_title('Age vs Weight')
ax2.scatter(hwa['Age'],results2.fittedvalues, label = 'Fitted Values')
ax2.legend()
plt.show()
```



In [18]:

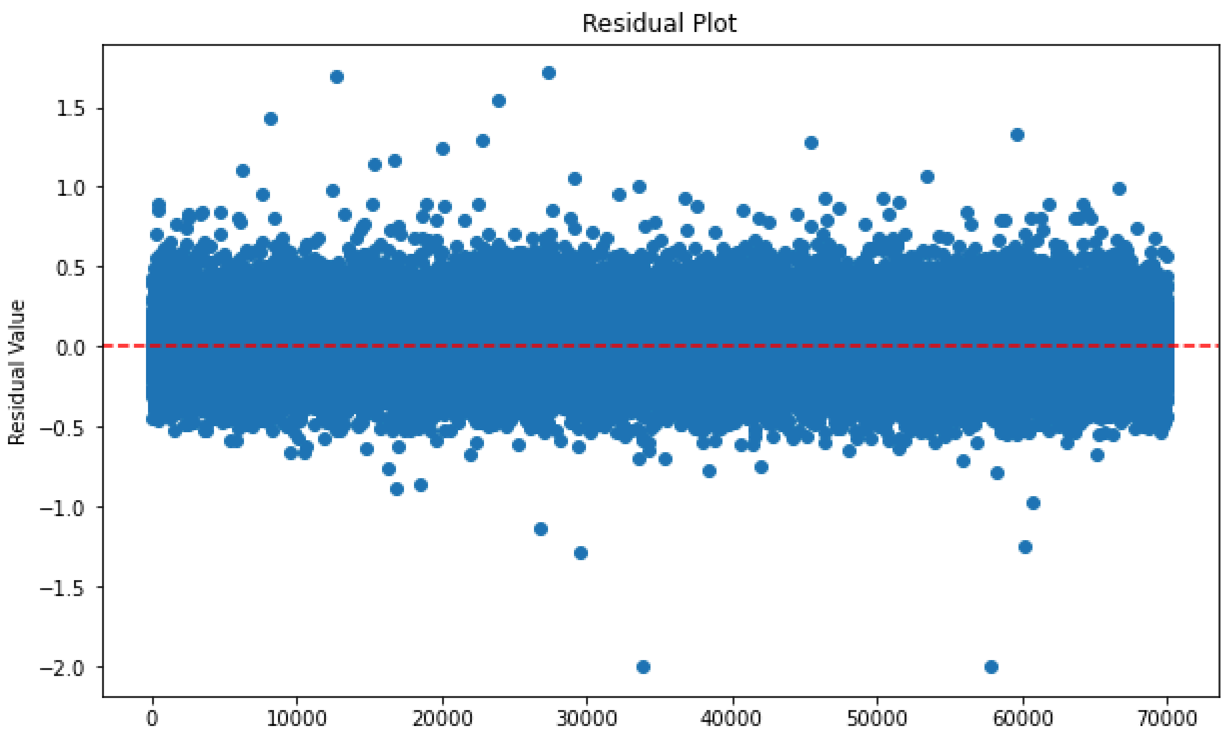
```python
loghwa = hwa.copy()
loghwa['Age'] = loghwa['Age'].apply(math.log)
loghwa['Height'] = loghwa['Height'].apply(math.log)
loghwa['Weight'] = loghwa['Weight'].apply(math.log)

results3 = smf.ols('Weight ~ Height + Age', data = loghwa).fit()
print(results3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Weight   R-squared:                       0.096
Model:                            OLS   Adj. R-squared:                  0.096
Method:                 Least Squares   F-statistic:                     3727.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):               0.00
Time:                        18:23:12   Log-Likelihood:                 21527.
No. Observations:               70000   AIC:                        -4.305e+04
Df Residuals:                   69997   BIC:                        -4.302e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -1.8100      0.071    -25.504      0.000      -1.949      -1.671
Height         1.1024      0.013     84.688      0.000       1.077       1.128
Age            0.1200      0.005     23.320      0.000       0.110       0.130
```

```
==============================================================================
Omnibus:                     5222.715   Durbin-Watson:                   1.990
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            12204.600
Skew:                           0.468   Prob(JB):                         0.00
Kurtosis:                       4.819   Cond. No.                         701.
==============================================================================

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

In [19]:

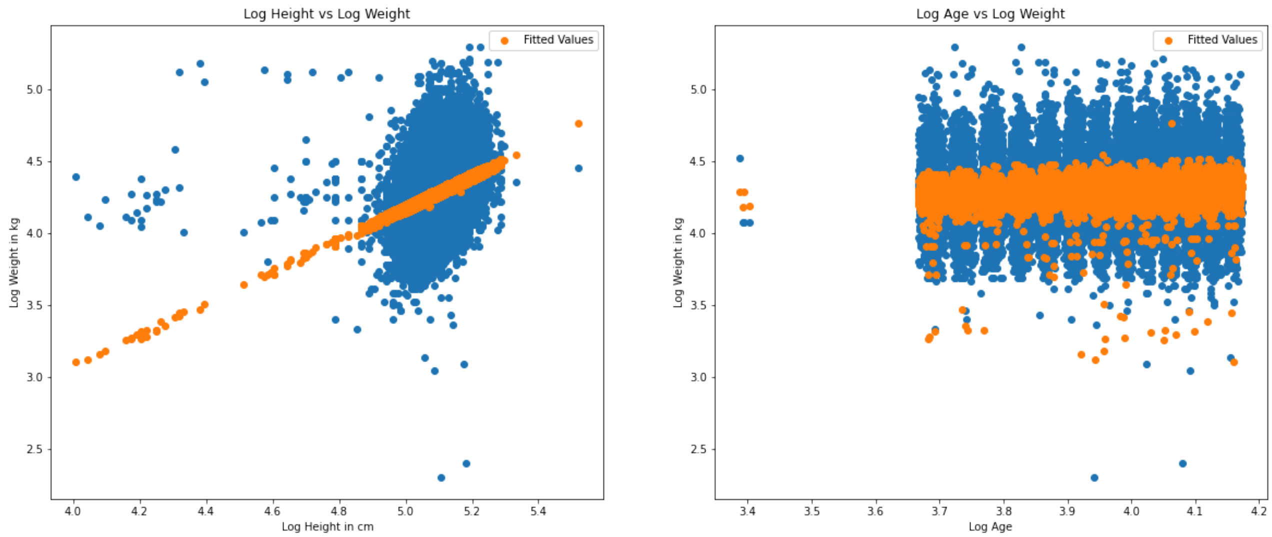
```
plt.figure(figsize=(10,6))
plt.scatter(loghwa.index,results3.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



In [20]:

```
hei = loghwa['Height']
wei = loghwa['Weight']
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,8))
ax1.scatter(hei,wei)
ax1.set_xlabel('Log Height in cm')
ax1.set_ylabel('Log Weight in kg')
ax1.set_title('Log Height vs Log Weight')
ax1.scatter(loghwa['Height'],results3.fittedvalues, label = 'Fitted Values')
ax1.legend()


ax2.scatter(loghwa['Age'],wei)
ax2.set_xlabel('Log Age')
ax2.set_ylabel('Log Weight in kg')
ax2.set_title('Log Age vs Log Weight')
ax2.scatter(loghwa['Age'],results3.fittedvalues, label = 'Fitted Values')
```

```
ax2.legend()
plt.show()
```



5. Solutions

In [21]:

```
cov = pd.read_csv('/Users/zanderbonnet/Desktop/GCU/DSC_510/DataSets/covidUS/us_c
cov = cov.sort_values(by = 'date', ignore_index=True)
cov['day'] = cov.index
cov = cov.fillna(0)
cov.head()
```

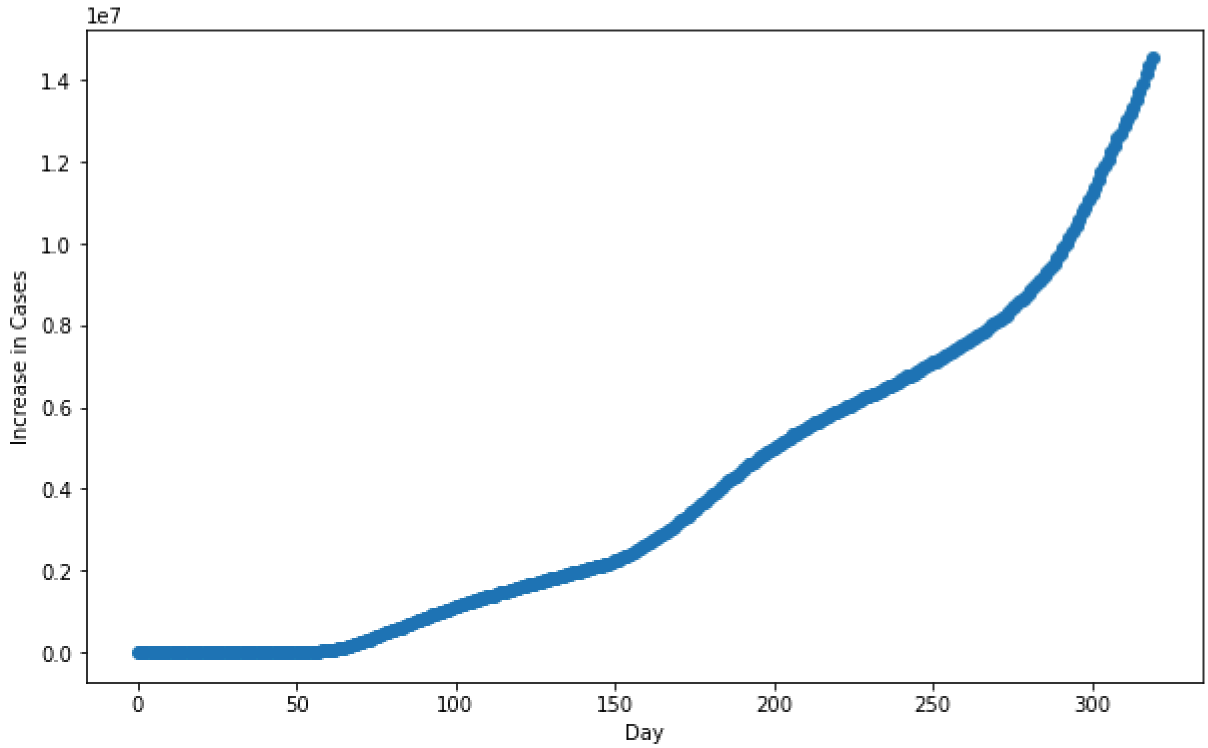
Out[21]:

	date	states	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	in
0	20200122	2	0	0	0.0	0.0	0.0	
1	20200123	2	0	0	0.0	0.0	0.0	
2	20200124	2	0	0	0.0	0.0	0.0	
3	20200125	2	0	0	0.0	0.0	0.0	
4	20200126	2	0	0	0.0	0.0	0.0	

5 rows × 26 columns

In [22]:

```
plt.figure(figsize = (10,6))
plt.scatter(cov['day'],cov['positive'])
plt.xlabel('Day')
plt.ylabel('Increase in Cases')
plt.show()
```



In [23]:

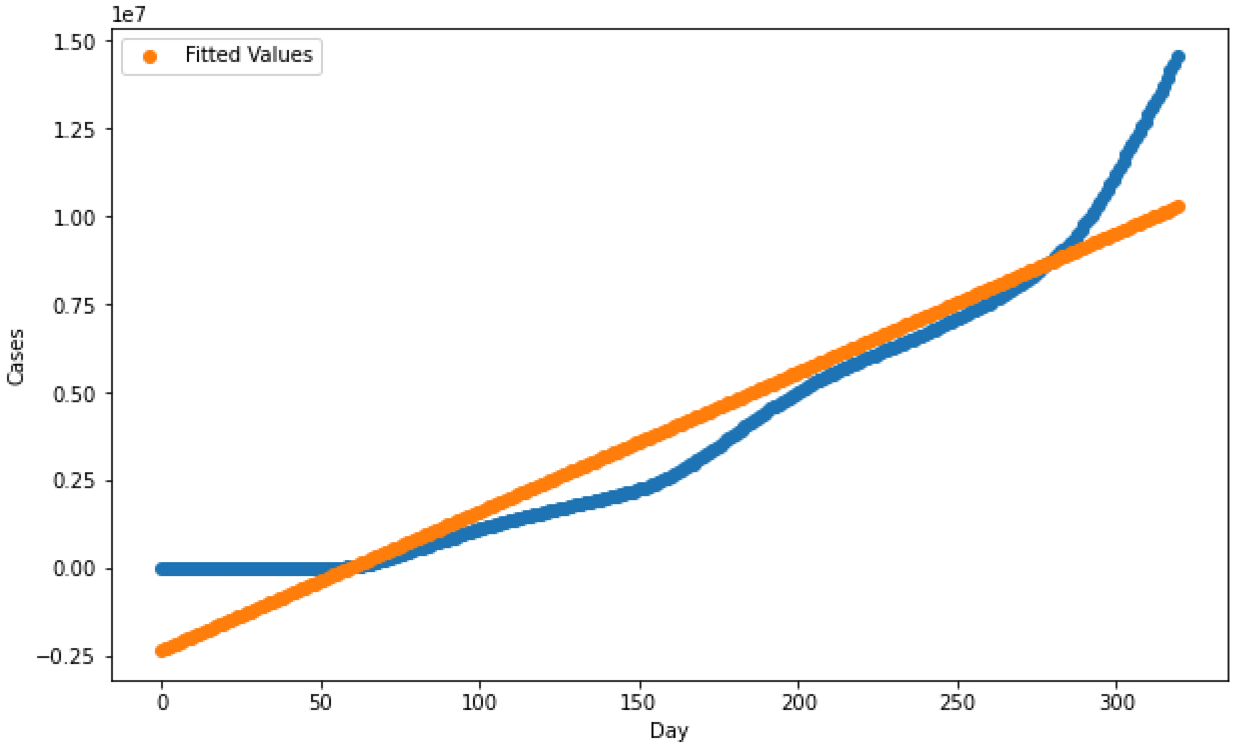
```
reg = smf.ols('positive ~ day', data = cov).fit()
print(reg.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               positive   R-squared:                       0.908
Model:                            OLS   Adj. R-squared:                  0.908
Method:                 Least Squares   F-statistic:                     3146.
Date:                Wed, 20 Mar 2024   Prob (F-statistic):          5.93e-167
Time:                        18:23:14   Log-Likelihood:                -4922.8
No. Observations:                 320   AIC:                             9850.
Df Residuals:                     318   BIC:                             9857.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept  -2.341e+06    1.3e+05    -18.023      0.000    -2.6e+06   -2.09e+06
day         3.953e+04    704.760     56.085      0.000    3.81e+04    4.09e+04
==============================================================================
Omnibus:                       77.294   Durbin-Watson:                   0.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              132.617
Skew:                           1.382   Prob(JB):                     1.59e-29
Kurtosis:                       4.518   Cond. No.                         368.
==============================================================================

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

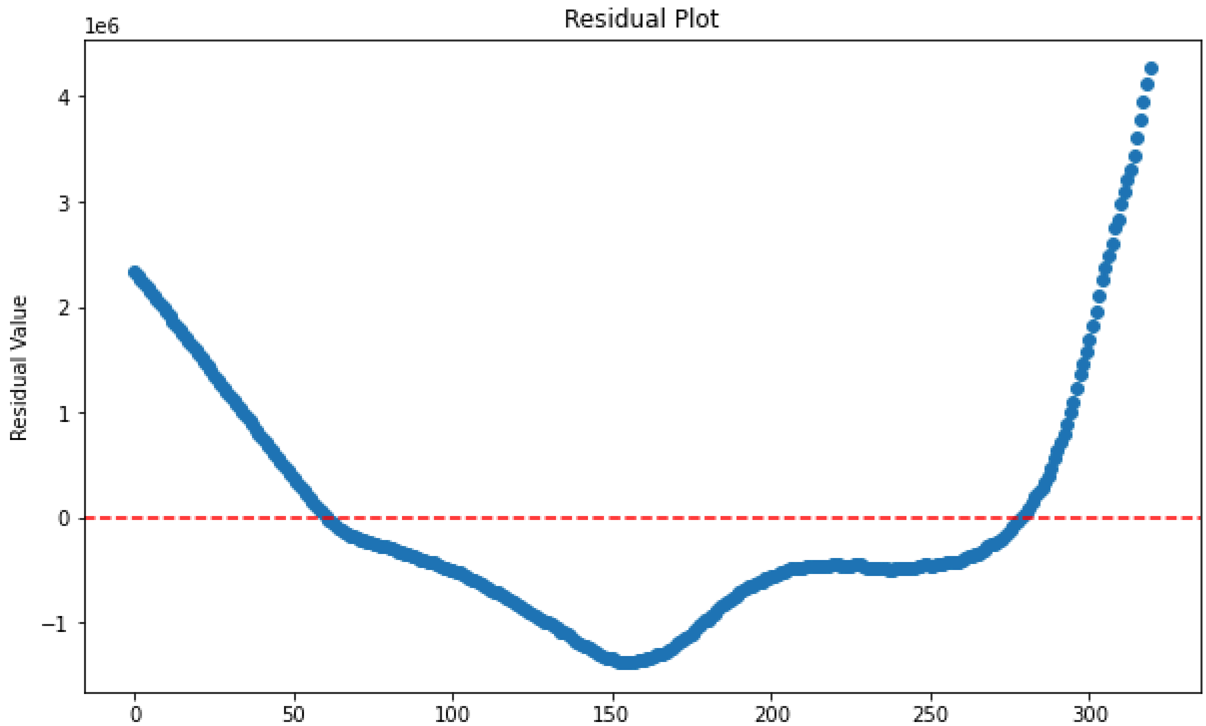
In [24]:

```
plt.figure(figsize = (10,6))
plt.scatter(cov['day'],cov['positive'])
plt.xlabel('Day')
plt.ylabel('Cases')
```

```
plt.scatter(cov['day'], reg.fittedvalues, label = 'Fitted Values')
plt.legend()
plt.show()
```



In [25]:
```
plt.figure(figsize=(10,6))
plt.scatter(cov['day'],reg.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



In [26]:

```
logpos = []
for num in cov['positive']:
    logpos.append(math.sqrt(num))
cov['logPos'] = logpos
cov.head()
```

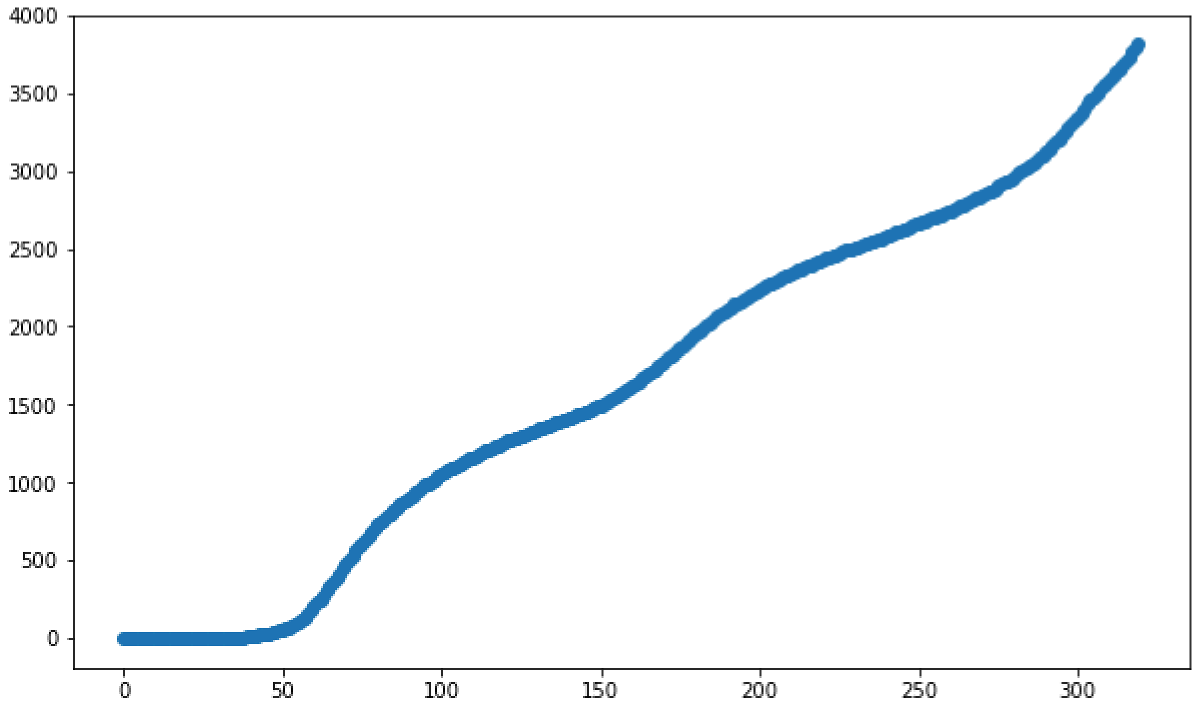
Out[26]:

	date	states	positive	negative	pending	hospitalizedCurrently	hospitalizedCumulative	in
0	20200122	2	0	0	0.0	0.0	0.0	
1	20200123	2	0	0	0.0	0.0	0.0	
2	20200124	2	0	0	0.0	0.0	0.0	
3	20200125	2	0	0	0.0	0.0	0.0	
4	20200126	2	0	0	0.0	0.0	0.0	

5 rows × 27 columns

In [27]:

```
plt.figure(figsize = (10,6))
plt.scatter(cov['day'],cov['logPos'])
plt.show()
```



In [28]:

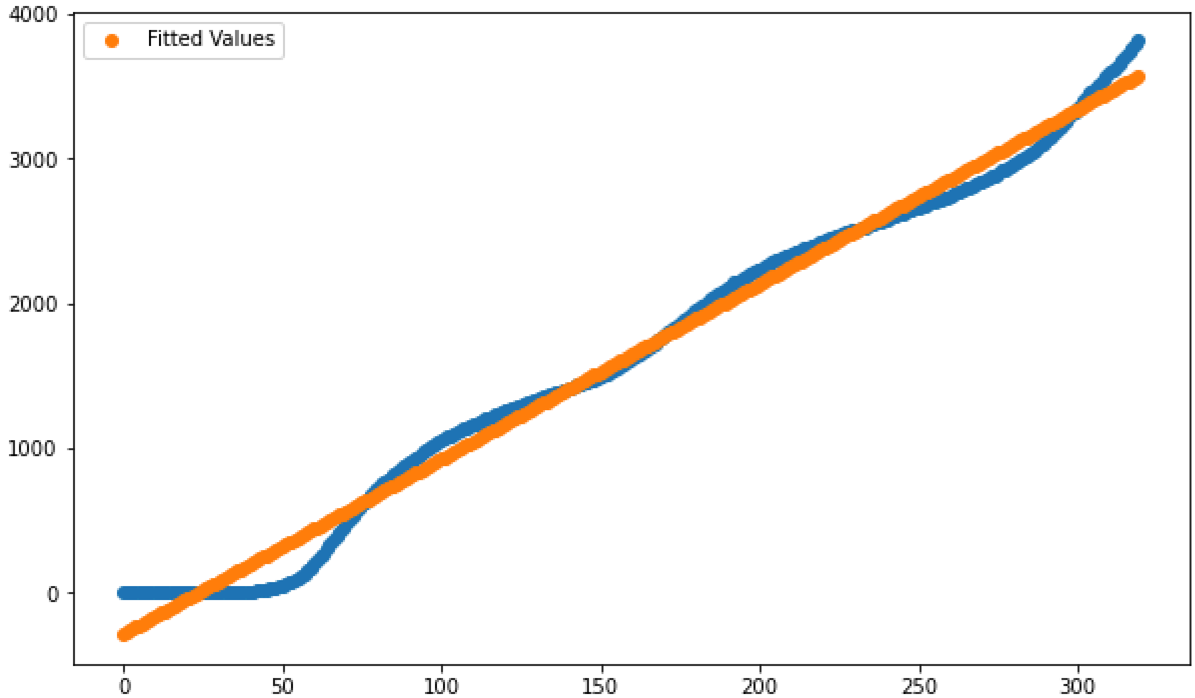
```
reg1 = smf.ols('logPos ~ day', data = cov).fit()
print(reg1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 logPos   R-squared:                       0.989
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                 2.863e+04
Date:                Wed, 20 Mar 2024   Prob (F-statistic):          1.35e-313
Time:                        18:23:14   Log-Likelihood:                -1979.7
No. Observations:                 320   AIC:                             3963.
Df Residuals:                     318   BIC:                             3971.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept   -283.5238     13.164    -21.538      0.000    -309.424    -257.624
day           12.0856      0.071    169.217      0.000      11.945      12.226
==============================================================================
Omnibus:                        8.559   Durbin-Watson:                   0.004
Prob(Omnibus):                  0.014   Jarque-Bera (JB):                8.783
Skew:                          -0.385   Prob(JB):                       0.0124
Kurtosis:                       2.745   Cond. No.                         368.
==============================================================================

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

In [29]:

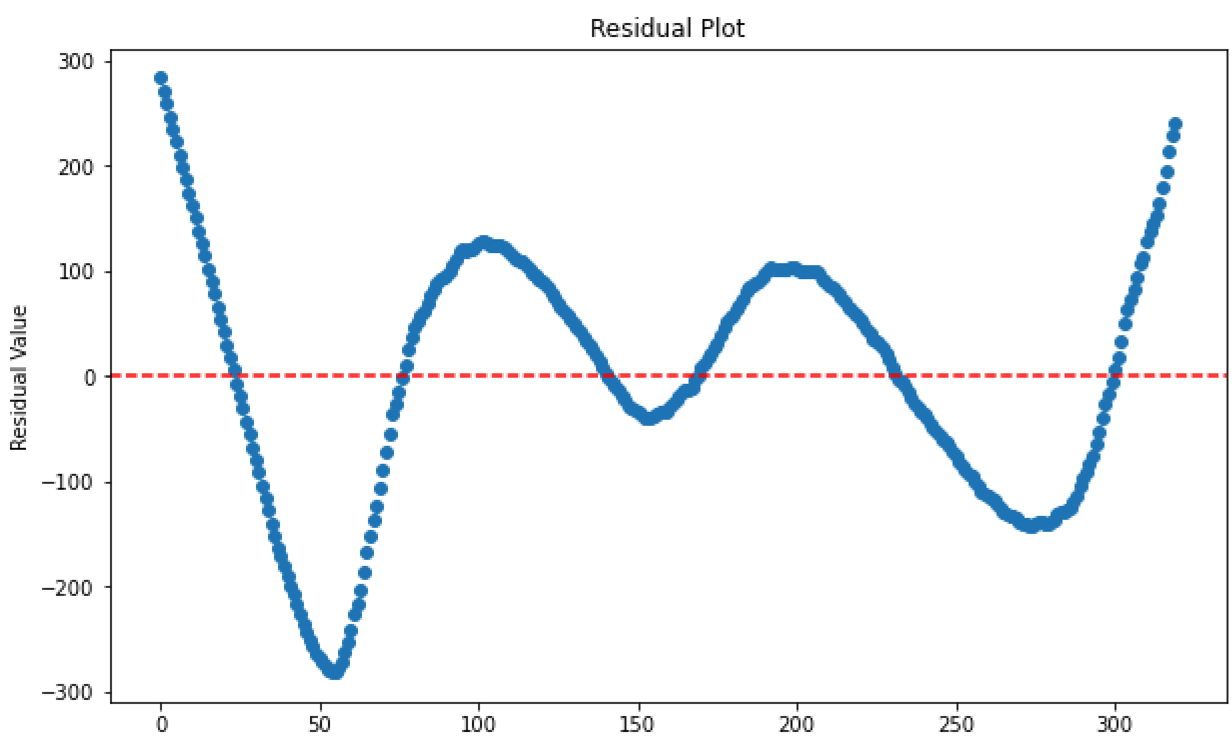
```
plt.figure(figsize = (10,6))
plt.scatter(cov['day'],cov['logPos'])

plt.scatter(cov['day'], reg1.fittedvalues, label = 'Fitted Values')
```

```
plt.legend()
plt.show()
```



In [30]:

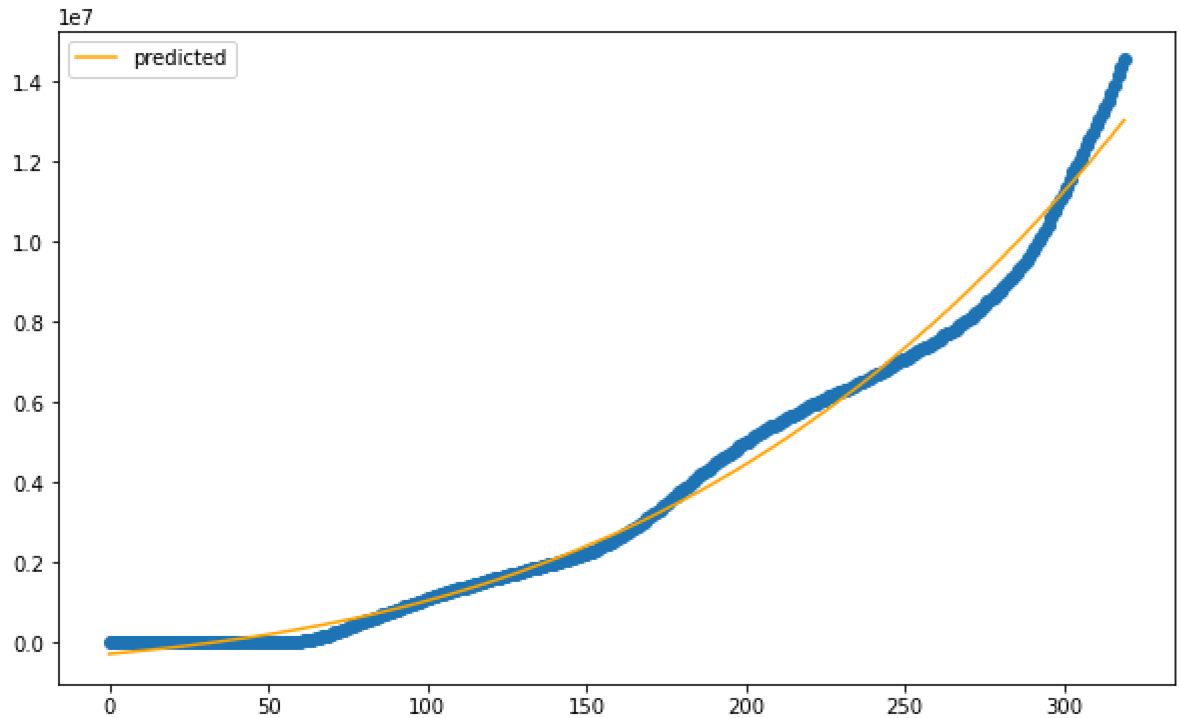
```
plt.figure(figsize=(10,6))
plt.scatter(cov['day'],reg1.resid)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.ylabel('Residual Value')
plt.show()
```



6. Non Linear Regression

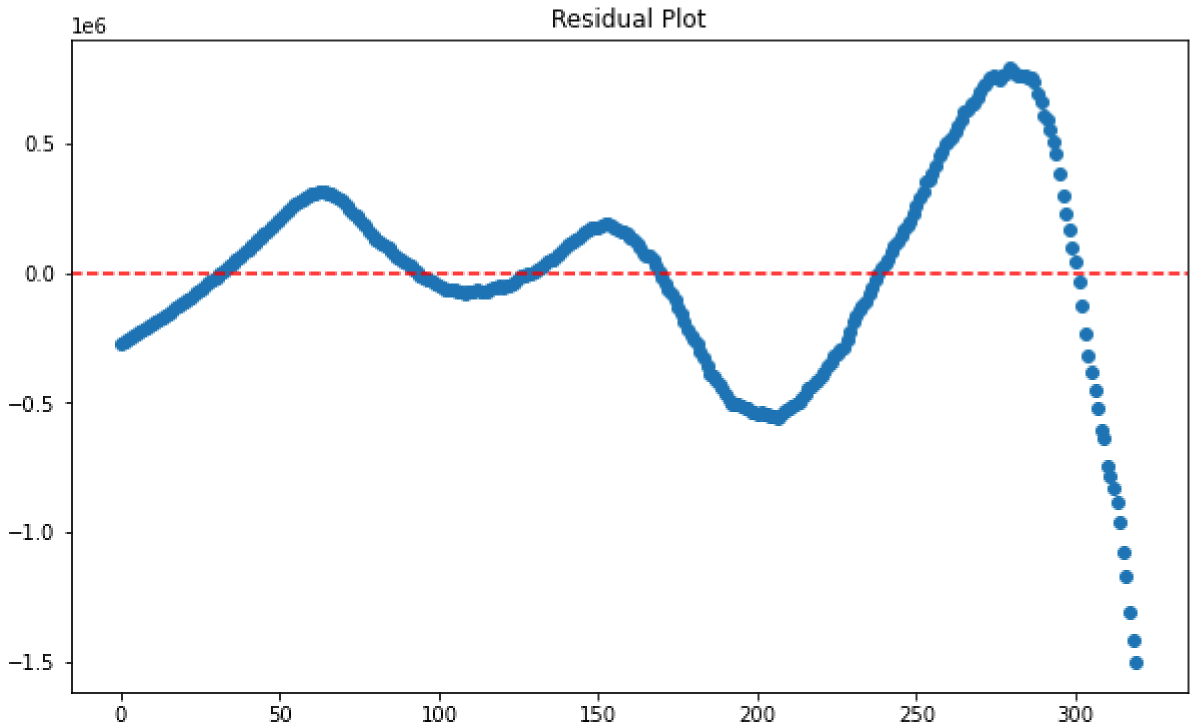
In [31]:

```
clf = KernelRidge(kernel='polynomial', gamma = .2)
d = np.array(cov['day']).reshape(-1,1)
clf.fit(d, cov['positive'])
pred = clf.predict(d)

plt.figure(figsize=(10,6))
plt.scatter(d,cov['positive'])
plt.plot(d,pred, label = 'predicted', color = 'orange')
plt.legend()
plt.show()
```



In [32]:

```
res = pred - cov['positive']
plt.figure(figsize=(10,6))
plt.scatter(cov['day'], res)
plt.axhline(0, linestyle = '--', color = 'red')
plt.title('Residual Plot')
plt.show()
```


Residual Plot
1e6
0.5
0.0
−0.5
−1.0
−1.5
0
50
100
150
200
250
300