APPENDIX

OLS Model: Soil Transmitted Helminthiasis against Body Mass Index

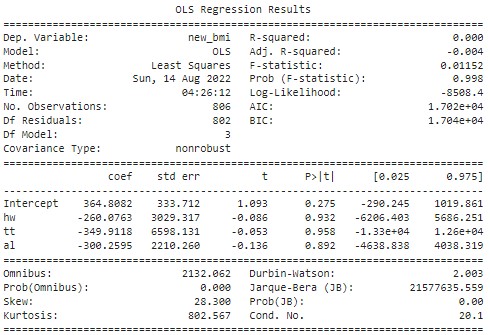
1. import statsmodels.formula.api as sm
2. norm = Normalizer ()
3. # Correlation between been infected with Soil Transmitted Helminthiasis and body mass index
4. new\_bmi = np. array(bmi) # changed the body mass index into a numpy array
5. hw = norm.fit\_transform (df.iloc[:, 23:24].values) #psac and sac hookworm
6. tt = norm.fit\_transform(df.iloc[:, 22:23].values) #psac ans sac Trichuris trichiura
7. al = norm.fit\_transform(df.iloc[:, 21:22].values) #psac and sac Ascaris trichiura
8. sth\_anem\_model = sm.ols('new\_bmi ~ hw + tt + al', data= df).fit()
9. print(sth\_anem\_model.summary())

Fig3: OLS Regression Model Result

**Model Equation:**

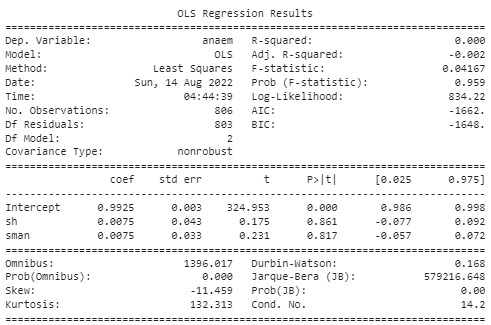
**BMI = (-260.07)hw + (-349.91.)TT + (-300.25)AL + 364.81**

***P-values*: 0.932 0.958 892 0.275**

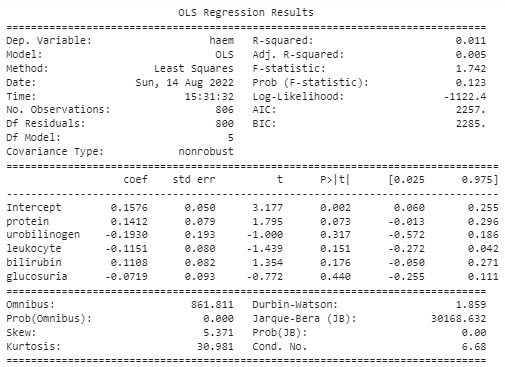
****

**Fig3: OLS Regression Model of body mass index against STH infection among PSAC and SAC, Borgou and Donga.**

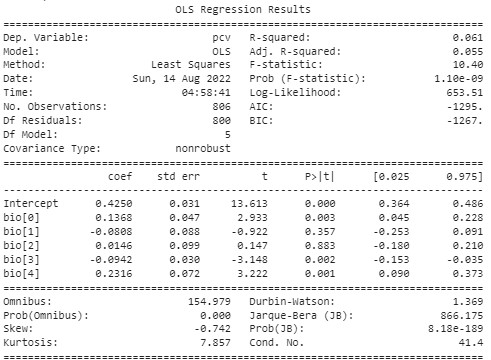
1. **Multiple Linear Regression: Schistosoma haematobium/ Schistosoma mansoni against Anemia (PCV)**
2. mms = MinMaxScaler()
3. norm = Normalizer()
4. sh = norm.fit\_transform(df.iloc[:,19:20].values) # Schistosoma  haematobium arrays
5. sman = norm.fit\_transform(df.iloc[:, 20:21].values) # Schistosoma mansoni arrays fro psac and sac
6. anaem = norm.fit\_transform(df.iloc[:,34:35].values) # This is an array of pcv2 : packedd cell volume of psac and sac.
7. # anaem.shape
8. prevalence1\_model = sm.ols('anaem ~ sh + sman', data=df).fit() # statistical model of schistosomiasis and anemia:
9. print(prevalence1\_model.summary()

****

1. **OLS Regression Model: Haematuria against Proteinuria, Leukocyturia, Bilirubinuria, Urobilinogen and Glucosuria.**
2. #Urinalysis: haematuria vs Proteinuria, leukocyturia, bil, Uro and Gluco
3. urine\_parameters = df.iloc[:,25:30].values
4. # urine parameters splitted below
5. protein = norm.fit\_transform(df.iloc[:,25:26].values)
6. urobilinogen =  norm.fit\_transform(df.iloc[:,26:27].values)
7. leukocyte =  norm.fit\_transform(df.iloc[:,27:28].values)
8. bilirubin =  norm.fit\_transform(df.iloc[:, 28:29].values)
9. glucosuria =  norm.fit\_transform(df.iloc[:,29:30].values)
10. #haematuria
11. haem = df.iloc[:, 18:19].values
12. haem = le.fit\_transform(haem) #transformed into float from partly string and int
13. urine\_parameters = mms.fit\_transform(urine\_parameters)
14. urine\_parameters = norm.fit\_transform(urine\_parameters)
15. # plt.scatter(urine\_parameters, haem)
16. # sns.regplot(urine\_parameters, haem)
17. model\_ols = sm.ols('haem ~ protein + urobilinogen + leukocyte + bilirubin + glucosuria', data=df).fit()
18. print(model\_ols.summary())
19. **See Figure below:**



1. **Linear Regression Model of Anthropometric Parameters against Anemia Status (PCV)**
2. from sklearn.model\_selection import train\_test\_split
3. from sklearn.metrics import r2\_score, mean\_squared\_error
4. lr = LinearRegression()
5. import statsmodels.formula.api as sm
6. #This is a combined biometrics to be used against anemia in combination with STH results
7. x\_biometrics = df.iloc[:, [7,8,9,10,11,34]].values  #'height(cm)','MUAC(cm)','weight(kg)', 'waist\_circ', pcv2
8. # from sklearn.preprocessing import MinMaxScaler
9. mms = MinMaxScaler()
10. x\_biometrics = mms.fit\_transform(x\_biometrics) #transforming the biometrics features to proper scaling.
11. bio = x\_biometrics[:,:5] #transformed biometrics indexed ,excluding the last which is the PCV2
12. pcv = x\_biometrics[:,5:]#the last column representing the pcv
13. sm.ols ('pcv ~ bio', data=df).fit() #fitting the model to the datasets
14. print (sm.ols('pcv ~ bio', data=df).fit().summary()) #Model summary



**Fig 5: OLS Regression Model: Anthropometrics against Packed Cell Volume**

1. **Linear Regression: Packed Cell Volume and Waist Circumference Model**
2. # Regression plot between PCV and Waist Circumference an extension of the statsmodel significance indication
3. pc = df[['PCV2']].values
4. wc\_train, wc\_test, pc\_train, pc\_test = train\_test\_split(wc, pc, test\_size = 0.2, random\_state=0)
5. modpc = lr.fit(wc\_train,pc\_train)
6. prediction = modpc.predict(wc\_test)
7. # plt.hist(prediction)
8. modpc.score(pc\_test, wc\_test) # model score is -0.00303671
9. r2\_score(pc\_test, wc\_test) # r2\_score is -12.01 this fitting is extremely bad.
10. lr.coef\_ # Regression coeficient is 0.18
11. # plt.scatter(pc\_train, wc\_train)
12. # plt.plot(pc\_train, lr.predict(pc\_train), color='red')
13. sns.regplot(pc,wc, color= 'red')
14. plt.title('Linear Regression Plot: PCV and Waist Circumference')
15. plt.xlabel('Waist Circumference')
16. plt.ylabel('Packed Cell Volume')

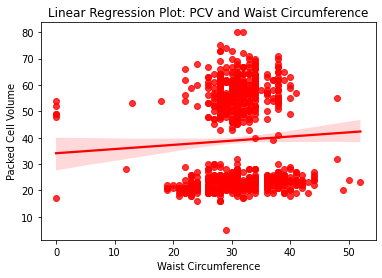
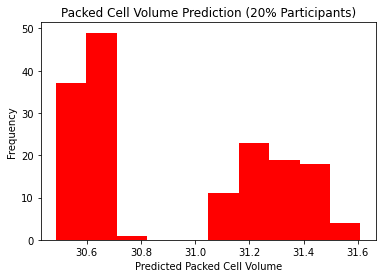


Fig: Regression Plot between packed cell Volume and Waist Circumference.

1. **Fig: Showing the trained Model Regression Plot.**
2. # Predicted Waist circumference
3. plt.hist(prediction, color='red')
4. plt.title('Packed Cell Volume Prediction (20% Participants)')
5. plt.xlabel('Predicted Packed Cell Volume')
6. plt.ylabel('Frequency')
7. prediction
8. mean\_squared\_error prediction, pc\_test) # Mean Squared Error is approx. 32.4

****

**Fig 6:Showing Packed Cell Volume Predicted Values for 162 PSAC and SAC,Borgou and Donga, Benin Republic.**

1. **ML Algorithm for OLS Regression Model Showing the Relationship between toilet types and STH infection.**
2. from sklearn.linear\_model import LogisticRegression #importing Logistic regression module form python library.
3. logreg = LogisticRegression()
4. # what are the determining factors behind having or not having toilet
5. x\_toil = toilet = df.iloc[:,13:14].values
6. x\_toil = le.fit\_transform(x\_toil).reshape(-1,1) # participants ot do not have toilet
7. x = norm.fit\_transform(x\_toil)
8. #These features are used for statsmodels to test if having/not having Toilet as a risk factor for STH in Borgou and Donga.
9. hw = norm.fit\_transform(df.iloc[:, 23:24].values).reshape(-1,1) #psac and sac hookworm
10. tt = norm.fit\_transform(df.iloc[:, 22:23].values).reshape(-1,1) #psac ans sac Trichuris trichiura
11. al = norm.fit\_transform(df.iloc[:, 21:22].values).reshape(-1,1) #psac and sac Ascaris trichiura
12. stats\_model5 = sm.ols(formula='x\_toil ~ hw + tt + al', data=df).fit()
13. print(stats\_model5.summary())
14. # Further subjecting the dataset to train\_test\_split to plot the line of best fit for the Linear regression
15. from sklearn.model\_selection import train\_test\_split
16. x\_train, x\_test, y\_train,y\_test = train\_test\_split(x\_toil,hw + tt + al, test\_size=0.2 , random\_state=0)
17. model5 = logreg.fit(x\_train,y\_train)
18. y\_pred = model5.predict(x\_test) #prediction of average sth infection intensity among psac and sac in ouake and bembereke
19. print('The Model prediction score is:', model5.score(x\_test,y\_test)\*100)
20. print ('The mean squared error is:', mean\_squared\_error(y\_test,y\_pred))
21. plt.scatter(x\_train,y\_train, marker='+', color='green')
22. #Prediction probability
23. Probability\_prediction = model5.predict\_proba(x\_test) # shows the probaility of each participant getting infected with either of the sth
24. # model5.predict\_log\_proba(x\_test) #logarithmic probability of being infected with either of the sth
25. Mean\_probability = Probability\_prediction.mean() #mean probability of having either of the sth
26. print('Mean probability of having either of STH is:', (Mean\_probability))
27. from sklearn.metrics import accuracy\_score,r2\_score
28. print ('R-Square Score is', r2\_score(y\_test,y\_pred))
29. #R-Squared score i.e. regression score.

**Fig: OLS Regression Model Showing the Relationship between toilet types and STH infection.**

1. **Toilet Types Model Summary:**

* **The Model prediction score is: 96.29629629629629**
* **The mean squared error is: 0.037037037037037035**
* **Mean probability of having either of the STH is: 0.3333333333333333**

From the available evidences, it can be concluded that having or not having toilet does not significantly determine been infected with STH in Borgou and Donga.

**Having Toilet or Not (Summary):**

* **The Model prediction score is**: 16.666666666666664
* **The mean squared error is**: 764.4949135514612
* **Mean probability of having either of STH is**: 3.125 %
* **Accuracy score is** -29.127584302745184 #Linear regression model badly fits to the data by giving a negative accuracy score.

**In order to check the dataset are normally distributed, it wa subjected to Shapiro’s Test for normality**

**#N.B. All the features that is < 0.05 are not normally distributed, thus the**

#if p-value > 0.05, features are normally distributed.

**So, from the Shapiro test it can be concluded that these features are not normally distributed, thus we use kruskal-Wallis’s test to check to differentiate between groups of data**

**KRUSKAL WALLIS TEST**

Testing for significant difference in features.

**kruskal.test(age ~ al):**

Kruskal-Wallis chi-squared = 2.8839, df = 5, *p-value = 0.7179*

**Linear Regression Model of Ascaris lumbricoides by Age**

1. lm(al ~ age)
2. Call:
3. lm(formula = al ~ age)
4. Coefficients:
5. (Intercept) (x) age
6. 1.50850 -0.03228

lm(hw ~ age)

1. Call:
2. lm(formula = hw ~ age)
3. Coefficients:
4. (Intercept) age
5. 0.04588 0.03898

lm(anaemia ~ age)

1. Call:
2. lm(formula = anaemia ~ age)
3. Coefficients:
4. (Intercept) age
5. 0.42029 -0.02761

**References**

1. WHO, 2020:
2. Ibikounle et al 2018
3. Aboki et al. 2019