Appendices

Appendix A1 – Original Dataset

country	inc_level	life_exp	adult_m	HepB	BMI	alcohol	pm2.5	GDP	hlth_exc
Afghanistan	Low	62.7	245	65	23.0	0.2	59.9	583.0551	60.1
Albania	Upper_middle	76.4	96	98	26.8	7.5	18.2	4683.5192	265.9
Algeria	Upper_middle	76.4	95	91	25.6	0.9	34.5	4827.7243	291.6
Angola	Lower_middle	62.6	238	55	23.0	6.4	28.4	3533.8652	108.6
Antigua and Barbuda	High	75.0	120	92	26.8	7	18.0	13286.0098	657.2
Argentina	Upper_middle	76.9	111	92	27.8	9.8	11.7	10206.2496	997.9
Armenia	Lower_middle	74.8	116	94	26.5	5.5	32.9	3932.5546	366.0
Australia	High	82.9	61	94	27.4	10.6	7.3	55731.4956	4934.0
Austria	High	81.9	62	87	26.0	11.6	13.1	48077.8784	4536.1
Azerbaijan	Upper_middle	73.1	118	97	27.0	0.8	18.5	5861.5120	368.3
Bahamas	High	75.7	159	94	28.1	4.4	19.0	26439.0536	1685.5
Bahrain	High	79.1	57	99	25.0	1.9	69.0	22291.4042	1190.0
Bangladesh	Lower_middle	72.7	130	97	21.7	0	58.6	1029.5782	31.8
Barbados	High	75.6	100	97	28.0	9.6	22.4	16518.5530	1160.2
Belarus	Upper_middle	74.2	161	96	27.2	11.2	19.3	6372.4018	351.8
Belgium	High	81.2	72	97	26.7	12.1	13.0	45457.8993	4228.3
Belize	Upper_middle	70.5	179	95	28.9	6.8	20.9	4304.2028	301.2
Benin	Low	61.1	242	82	23.4	3	30.4	837.1990	31.3
Bhutan	Lower_middle	70.6	207	98	23.9	0.6	35.4	2803.3282	91.1
Bolivia	Lower_middle	71.5	182	87	26.3	4.8	23.3	2457.6376	197.3
Bosnia and Herzegovina	Upper_middle	77.3	92	78	26.2	6.4	29.7	5382.8013	431.2
Botswana	Upper_middle	66.1	249	95	23.8	8.4	20.9	7485.0421	389.4
Brazil	Upper_middle	75.1	143	89	26.6	7.8	11.8	10868.6534	780.4
Brunei Darussalam	High	76.4	98	99	27.4	0.4	5.8	31430.9626	812.2
Bulgaria	Upper_middle	74.8	135	91	27.1	12.7	20.8	7966.8820	572.0
Burkina Faso	Low	60.3	255	91	22.2	8.2	36.3	664.0217	33.4
Burundi	Low	60.1	290	94	21.7	7.5	35.6	219.2066	24.3
Cabo Verde	Lower_middle	73.2	122	96	24.3	5.7	31.6	3477.0594	145.8
Cambodia	Lower_middle	69.4	170	93	22.1	6.7	24.9	1078.2055	69.6
Cameroon	Lower_middle	58.1	341	85	24.1	8.9	65.4	1498.2786	63.6
Canada	High	82.8	63	62	27.3	8.9	6.7	50407.3413	4507.6
Central African Republic	Low	53.0	412	47	22.6	3.3	51.2	325.7205	16.6
Chad	Low	54.3	360	41	21.5	1.6	50.8	874.7671	35.6
Chile	High	79.5	87	95	28.1	9.3	23.1	14958.5622	1102.0
China	Upper_middle	76.4	80	99	24.0	7.2	51.0	6894.4645	425.6
Colombia	Upper_middle	75.1	137	91	26.2	5.8	17.2	7541.2214	374.2
Comoros	Low	63.9	225	91	24.4	0.9	18.6	1350.1461	58.5
Congo, Dem. Rep.	Low	60.5	256	79	21.8	2.6	37.4	407.5637	19.8
Congo, Rep.	Lower_middle	64.3	261	71	23.0	7.8	36.4	2771.3973	58.8
Costa Rica	Upper_middle	79.6	97	97	27.3	4.8	16.7	9592.1117	929.1
Cote d'Ivoire	Lower_middle	54.6	398	82	23.6	8.4	23.9	1547.5526	75.5
Croatia	Upper_middle	78.3	88	93	28.1	8.9	17.6	14713.5808	852.1
Cuba	Upper_middle	79.0	92	99	25.9	6.1	21.6	6469.8667	826.3
Cyprus	High	80.7	55	97	27.1	10.8	17.1	28549.4697	1563.3

country	inc_level	life_exp	adult_m	HepB	BMI	alcohol	pm2.5	GDP	hlth_exc
Czechia	High	79.2	81	96	27.9	14.4	15.6	21894.1127	1284.0
Denmark	High	81.2	65	-	25.7	10.4	10.3	60670.2372	5497.4
Djibouti	Lower_middle	63.8	245	68	23.8	0.5	41.0	-	81.7
Dominican Republic	Upper_middle	73.5	160	80	26.4	6.9	13.3	6917.6049	396.7
Ecuador	Upper_middle	76.5	114	84	27.1	4.4	15.5	5209.5592	530.1
Egypt	Lower_middle	70.5	165	95	29.4	0.4	79.6	2725.5558	156.6
El Salvador	Lower_middle	73.7	177	93	27.4	3.8	23.8	3402.6009	283.2
Equatorial Guinea	Upper_middle	59.5	305	19	24.0	11.3	49.1	12320.5246	280.0
Eritrea	Low	65.0	252	95	20.5	1.3	41.1	-	30.9
Estonia	High	77.8	119	93	27.0	11.6	7.0	18094.5873	1112.0
Ethiopia	Low	65.5	219	73	20.5	2.9	34.0	511.1874	24.3
Fiji	Upper_middle	69.9	186	99	27.7	3	10.5	4195.9681	174.9
Finland	High	81.4	70	-	26.6	10.7	6.5	45983.3133	4005.5
France	High	82.9	71	90	25.6	12.6	12.4	41968.9816	4026.1
Gabon	Upper_middle	66.4	221	75	24.9	11.5	37.8	9553.4952	197.9
Gambia	Low	61.9	262	95	23.6	3.8	32.3	520.6801	31.9
Georgia	Lower_middle	72.6	160	92	28.1	9.8	24.0	4083.9982	280.9
Germany	High	81.0	69	87	27.4	13.4	11.9	45923.0076	4591.8
Ghana	Lower_middle	63.4	241	93	23.8	2.7	31.1	1659.4842	79.6
Greece	High	81.2	66	96	27.7	10.4	16.4	22687.6003	1504.8
Grenada	Upper_middle	73.4	140	96	26.9	9.3	21.8	8681.0955	460.3
Guatemala	Lower_middle	73.2	162	80	26.3	2.5	24.2	3100.9790	224.4
Guinea	Low	59.8	262	45	22.7	1.3	22.2	764.4043	25.1
Guinea-Bissau	Low	59.8	269	87	23.2	4.8	26.5	584.0792	39.5
Guyana	Upper_middle	66.2	264	97	26.3	6.3	21.6	3783.5420	184.5
Haiti	Low	63.5	243	58	25.1	5.8	14.7	729.3605	53.6
Honduras	Lower_middle	75.2	145	97	26.5	4	21.5	2144.8694	176.6
Hungary	High	76.0	126	-	27.9	11.4	16.3	14997.2005	893.7
Iceland	High	82.4	55	-	26.5	9.1	5.9	48995.1685	4375.4
India	Lower_middle	68.8	178	88	21.8	5.7	68.0	1862.4304	63.3
Indonesia	Lower_middle	69.3	176	79	23.2	0.8	16.4	3974.7317	111.8
Iran. Islamic Rep.	Upper_middle	75.7	80	99	26.1	1	34.4	6733.9135	366.0
Iraq	Upper_middle	69.8	174	63	28.3	0.4	60.1	5725.7218	154.5
Ireland	High	81.5	62	95	27.8	13	8.7	69892.3156	4757.1
Israel	High	82.3	58	95	27.4	3.8	19.4	33677.4619	2756.1
Italy	High	82.8	54	94	26.3	7.5	15.7	34318.4968	2700.4
Jamaica	Upper_middle	76.0	131	98	27.7	4.2	13.6	4794.7575	294.3
Japan	High	84.2	51	-	22.8	8	11.8	47660.8930	3732.6
Jordan	Lower_middle	74.3	111	98	28.5	0.7	31.7	3258.4932	257.4
Kazakhstan	Upper_middle	71.1	181	82	26.4	7.7	14.5	10582.7009	379.1
Kenya	Lower_middle	66.7	219	89	23.2	3.4	25.8	1143.3643	70.1
Kiribati	Lower_middle	66.1	197	81	30.0	0.4	10.9	1743.9354	108.4
Kuwait	High	74.8	79	99	29.5	0	58.9	35250.9146	1168.8
Kyrgyzstan	Lower_middle	71.4	162	96	26.2	6.2	17.4	1043.9511	92.1
Laos	Lower_middle	65.8	193	82	22.7	10.4	25.5	1642.7301	53.0
Latvia	High	75.0	154	98	27.4	12.9	14.4	14733.8394	783.8
Lebanon	Upper_middle	76.3	96	78	27.5	1.5	30.7	7083.8173	645.2
Lesotho	Lower_middle	52.9	483	93	24.5	5	28.1	1352.2023	90.9
Liberia	Low	62.9	230	79	24.0	5.8	17.0	545.0555	69.3
Libya	Upper_middle	71.9	150	97	27.9	0	41.7	5848.9374	-
Lithuania	High	75.0	155	95	27.0	15	12.3	15945.5235	923.3
Luxembourg	High	82.4	56	94	26.8	13	10.4	108600.9349	6236.0

country	inc_level	life_exp	adult_m	HepB	BMI	alcohol	pm2.5	GDP	hlth_exp
Macedonia, FYR	Upper_middle	75.9	96	94	27.0	8.1	33.0	5246.0885	295.4
Madagascar	Low	66.1	216	77	21.2	1.9	22.5	5246.0885	21.0
Malawi	Low	64.2	255	84	22.8	3.7	21.9	416.0027	34.2
Malaysia	Upper_middle	75.3	123	98	25.7	0.9	17.3	481.4491	385.6
Maldives	Upper_middle	78.4	53	99	25.1	2.7	7.7	11038.8693	943.9
Mali	Low	58.0	270	66	22.9	1.3	29.0	8559.3509	42.3
Malta	High	81.5	55	97	27.7	8.1	14.0	745.8707	2304.2
Mauritania	Lower_middle	63.9	202	74	24.6	0	41.7	26064.1714	53.6
Mauritius	Upper_middle	74.8	145	72	25.4	3.6	13.5	1294.2548	506.1
Mexico	Upper_middle	76.6	127	93	27.9	6.5	20.9	9833.6212	534.8
Micronesia	Lower_middle	69.6	164	76	29.1	2.5	10.5	9867.9308	394.9
Moldova	Lower_middle	71.5	167	90	27.4	15.2	16.5	2070.6344	186.4
Monaco	High	89.5	-	99	-	-	12.2	191586.6000	3315.8
Mongolia	Lower_middle	69.8	211	99	26.1	7.4	49.5	2845.0116	152.5
Montenegro	Upper_middle	76.8	99	75	26.9	8	19.3	7487.3867	382.1
Morocco	Lower_middle	76.0	69	99	25.9	0.6	31.1	3204.8647	159.8
Mozambique	Low	60.1	336	80	22.6	2.4	18.4	514.9632	28.3
Myanmar	Lower_middle	66.8	195	90	22.7	4.8	34.6	1408.0142	59.1
Namibia	Upper_middle	63.7	296	85	23.8	9.8	21.0	6024.3406	423.1
Nepal	Low	70.2	151	87	22.4	2	99.5	686.3179	44.4
Netherlands	High	81.6	59	93	26.1	8.7	12.1	52267.7318	4746.0
New Zealand	High	82.2	66	92	28.2	10.7	5.8	37528.4583	3553.6
Nicaragua	Lower_middle	75.5	146	98	26.9	5.2	19.0	1944.0010	162.9
Niger	Low	59.8	250	80	21.8	0.5	73.0	392.1710	25.7
Nigeria	Lower_middle	55.2	352	42	23.0	13.4	46.3	2455.9186	97.3
North Korea	Low	71.9	132	96	24.2	3.9	31.0	-	-
Norway	High	82.5	55	-	27.1	7.5	7.8	90316.9723	7464.1
Oman	High	77.0	96	99	26.2	0.8	36.2	16962.3488	636.5
Pakistan	Lower_middle	66.5	159	75	23.8	0.3	56.2	1179.4140	38.0
Panama	Upper_middle	78.0	111	86	26.9	7.9	12.0	11116.7158	920.6
Papua New Guinea	Lower_middle	65.9	224	66	25.8	1.2	11.5	2390.9316	77.3
Paraguay	Upper_middle	74.2	146	92	26.5	7.2	11.7	5129.3917	321.3
Peru	Upper_middle	75.9	124	89	26.6	6.3	29.0	6094.7382	323.0
Philippines	Lower_middle	69.3	194	86	23.1	6.6	18.7	2752.1091	126.9
Poland	High	77.8	111	95	27.3	11.6	21.5	15099.3493	796.7
Portugal	High	81.5	76	98	26.2	12.3	8.1	22444.0711	1721.7
Qatar	High	78.1	62	98	29.2	2	91.7	66419.3479	2029.5
Romania	Upper_middle	75.2	135	90	27.5	12.7	15.4	10236.8603	442.4
Russian Federation	Upper_middle	71.9	203	97	26.7	11.7	14.7	11279.6253	523.8
Rwanda	Low	68.0	198	98	22.2	9	40.7	738.8057	56.7
Saint Lucia	Upper_middle	75.6	147	95	30.0	9.9	21.2	8356.9892	481.6
St.Vincent and the Grenadines	Upper_middle	72.0	169	99	27.2	8.2	21.4	6681.2836	284.0
Samoa	Upper_middle	75.1	110	55	31.9	2.5	10.9	3783.9088	222.9
Sao Tome and Principe	Lower_middle	68.7	191	96	24.5	6.8	25.2	1284.6884	159.9
Saudi Arabia	High	74.8	89	98	28.5	0.2	86.7	21380.4527	1194.1
Senegal	Low	66.8	185	93	22.5	0.7	39.7	1389.8882	36.1
Serbia	Upper_middle	76.3	103	91	26.5	11.1	24.7	5852.3843	491.3
Seychelles	High	73.3	163	97	26.9	12	18.6	13598.3363	491.8
Sierra Leone	Low	53.1	389	84	22.7	5.7	20.6	453.6639	106.7
Singapore	High	82.9	51	96	23.8	2	18.3	53353.8395	2280.3

country	inc_level	life_exp	adult_m	HepB	BMI	alcohol	pm2.5	GDP	hlth_exp
Slovakia	High	77.4	104	96	27.0	11.5	18.0	19275.0868	1108.4
Slovenia	High	80.9	72	-	27.0	12.6	16.4	24460.3979	1771.6
Solomon Islands	Lower_middle	71.1	146	99	26.2	1.4	11.5	1502.8871	152.1
Somalia	Low	55.4	316	42	22.9	0	28.0	-	-
South Africa	Upper_middle	63.6	301	66	27.0	9.3	24.3	7520.5332	470.8
South Korea	High	82.7	61	98	23.9	10.2	24.7	25484.0376	2012.7
South Sudan	Low	58.6	321	26	-	-	40.9	-	28.0
Spain	High	83.1	56	97	26.6	10	9.8	31505.6122	2353.9
Sri Lanka	Lower_middle	75.3	131	99	23.2	4.3	15.1	3768.5168	117.9
Sudan	Lower_middle	65.1	224	93	24.2	0.5	46.8	1923.9953	151.8
Suriname	Upper_middle	71.8	180	91	26.6	5.1	25.8	8005.1809	577.5
Sweden	High	82.4	52	76	26.4	9.2	6.1	56473.0228	5600.1
Switzerland	High	83.3	49	-	25.7	11.5	10.4	76682.6448	9818.0
Syrian Arab Republic	Lower_middle	63.8	301	50	27.8	0.3	37.4	-	-
Tajikistan	Lower_middle	70.8	123	97	25.6	3.3	42.8	968.1593	63.0
Tanzania	Low	63.9	261	97	23.0	9.4	25.1	867.0381	31.7
Thailand	Upper_middle	75.5	147	99	24.6	8.3	26.6	5910.4546	217.1
Timor-Leste	Lower_middle	68.6	150	76	21.1	2.1	18.2	2967.9911	71.7
Togo	Low	60.6	265	89	23.4	3.1	31.2	637.7843	36.6
Tonga	Upper_middle	73.4	133	78	31.9	1.5	10.2	3792.2901	221.4
Trinidad and Tobago	High	71.8	172	97	27.9	8.4	22.4	15765.4698	1146.2
Tunisia	Lower_middle	76.0	91	98	26.4	1.9	35.7	4269.1398	258.0
Turkey	Upper_middle	76.4	104	98	27.9	2	41.2	14117.4333	454.6
Turkmenistan	Upper_middle	68.2	191	98	26.2	5.4	24.2	6986.8559	405.1
UAE	High	77.2	74	99	28.6	3.8	37.2	41449.6814	1402.0
Uganda	Low	62.5	288	85	22.4	9.5	48.7	663.2220	46.1
UK	High	81.4	67	-	27.5	11.5	10.6	42039.7364	4355.8
Ukraine	Lower_middle	72.5	180	26	26.9	8.6	19.4	2905.8574	125.0
Uruguay	High	77.1	114	95	27.6	10.8	8.7	14042.6390	1281.3
USA	High	78.5	114	93	29.1	9.8	7.6	52319.1634	9535.9
Uzbekistan	Lower_middle	72.3	131	99	26.5	2.7	28.9	1961.4822	133.9
Vanuatu	Lower_middle	72.0	129	81	26.0	1	11.0	2873.6929	99.0
Venezuela	Upper_middle	74.1	157	84	26.7	5.6	16.8	-	973.0
Viet Nam	Lower_middle	76.3	125	96	21.9	8.4	30.1	1735.2910	116.7
Yemen	Lower_middle	65.3	221	71	23.4	0.1	44.3	754.0579	72.0
Zambia	Lower_middle	62.3	286	91	22.2	4.8	23.8	1629.5903	69.4
Zimbabwe	Low	61.4	334	90	23.4	4.8	19.1	1063.6013	94.3

Appendix A2 – Weights

Country	Weight	Country	Weight
Afghanistan	0.87294	Czechia	2.11018
Albania	1.66625	Denmark	1.66850
Algeria	1.80422	Djibouti	1.11401
Angola	0.89367	Dominican Republic	1.70096
Antigua and Barbuda	1.21874	Ecuador	1.80086
Argentina	2.02280	Egypt	1.38771
Armenia	0.99311	El Salvador	1.12867
Australia	1.84452	Equatorial Guinea	1.64209
Austria	1.67918	Eritrea	0.73756
Azerbaijan	2.08367	Estonia	1.67510
Bahamas	1.32303	Ethiopia	0.71267
Bahrain	1.07332	Fiji	1.99339
Bangladesh	1.01468	Finland	1.71024
Barbados	1.37355	France	1.73855
Belarus	1.97277	Gabon	1.66571
Belgium	1.75265	Gambia	0.72349
Belize	1.85245	Georgia	1.22906
Benin	0.76651	Germany	2.18144
Bhutan	1.07961	Ghana	0.95035
Bolivia	1.04065	Greece	1.45929
Bosnia and Herzegovina	1.55579	Grenada	1.70721
Botswana	1.40503	Guatemala	1.15009
Brazil	1.73552	Guinea	0.91400
Brunei Darussalam	1.77945	Guinea-Bissau	0.71625
Bulgaria	2.22203	Guyana	1.53710
Burkina Faso	0.66877	Haiti	0.86313
Burundi	0.64487	Honduras	1.07670
Cabo Verde	0.88712	Hungary	1.57789
Cambodia	0.84454	Iceland	1.63699
Cameroon	0.83552	India	0.74779
Canada	1.74077	Indonesia	1.16923
Central African Republic	0.71373	Iran, Islamic Rep.	1.80849
Chad	0.77505	Iraq	2.13081
Chile	1.37033	Ireland	2.19922
China	1.26968	Israel	1.34368
Colombia	1.60427	Italy	1.24032
Comoros	0.95704	Jamaica	1.85839
Congo, Dem.Rep.	0.72758	Japan	1.15993
Congo, Rep.	0.85706	Jordan	1.41809
Costa Rica	1.74257	Kazakhstan	1.67083
Cote d'Ivoire	0.90369	Kenya	0.92852
Croatia	1.86057	Kiribati	1.87942
Cuba	1.54120	Kuwait	1.62463
Cyprus	1.41718	Kyrgyzstan	1.04463
Country	Weight	Country	Weight
Laos	0.97954	Qatar	1.28760
Latvia	1.74418	Romania	2.34229
Lebanon	1.98527	Russian Federation	2.00952
Lesotho	0.87766	Rwanda	0.67555
Liberia	0.78219	Saint Lucia	2.18592
Libya	2.13529	St.Vincent and the Grenadines	1.66717

Country	Weight	Country	Weight
Lithuania	2.20333	Samoa	3.07054
Luxembourg	2.41943	Sao Tome and Principe	0.91532
Macedonia, FYR	1.59919	Saudi Arabia	1.46254
Madagascar	0.78759	Senegal	0.82896
Malawi	0.75326	Serbia	1.86401
Malaysia	1.97037	Seychelles	1.51150
Maldives	1.83375	Sierra Leone	0.70565
Mali	0.83722	Singapore	1.17272
Malta	1.48590	Slovakia	1.47706
Mauritania	1.11872	Slovenia	1.65633
Mauritius	1.76294	Solomon Islands	1.31047
Mexico	1.72092	Somalia	0.96670
Micronesia	1.49827	South Africa	1.74440
Moldova	2.02052	South Korea	1.13439
Monaco	1.79690	South Sudan	0.71790
Mongolia	0.90784	Spain	1.43456
Montenegro	1.73180	Sri Lanka	0.94586
Morocco	1.22760	Sudan	1.07670
Mozambique	0.80871	Suriname	1.57018
Myanmar	0.84180	Sweden	1.80439
Namibia	1.50792	Switzerland	3.08962
Nepal	0.68510	Syrian Arab Republic	1.43717
Netherlands	1.45285	Tajikistan	0.97823
New Zealand	1.86412	Tanzania	0.74051
Nicaragua	1.08274	Thailand	1.42628
Niger	0.77944	Timor-Leste	0.99855
Nigeria	1.22496	Togo	0.74961
North Korea	0.75246	Tonga	3.16608
Norway	2.08419	Trinidad and Tobago	1.29005
Oman	1.33181	Tunisia	1.12032
Pakistan	1.09915	Turkey	1.79138
Panama	1.78790	Turkmenistan	1.52033
Papua New Guinea	1.35283	UAE	1.24800
Paraguay	1.72052	Uganda	0.68502
Peru	1.55323	UK	1.83451
Philippines	0.90474	Ukraine	1.23844
Poland	1.49215	Uruguay	1.62159
Portugal	1.67381	USA	3.37401
Country	Weight		
Uzbekistan	1.10025		
Vanuatu	1.38000		
Venezuela	1.67881		
Viet Nam	0.84587		
Yemen	1.13222		
Zambia	0.85193		
Zimbabwe	0.73810		

Appendix A3 – R Code

```
rm(list = ls())
cat("\f")

setwd("")

################
# 0. PACKAGES  #
################

if(!require(Amelia)) install.packages("Amelia")              ; library(Amelia)
if(!require(VIM)) install.packages("VIM")                    ; library(VIM)
if(!require(missForest)) install.packages("missForest")      ; library(missForest)
if(!require(stargazer)) install.packages("stargazer")        ; library(stargazer)
if(!require(ggplot2)) install.packages("ggplot2")            ; library(ggplot2)
if(!require(ggpubr)) install.packages("ggpubr")              ; library(ggpubr)
if(!require(ggrepel)) install.packages("ggrepel")            ; library(ggrepel)
if(!require(ggridges)) install.packages("ggridges")          ; library(ggridges)
if(!require(moments)) install.packages("moments")            ; library(moments)
if(!require(arm)) install.packages("arm")                    ; library(arm)
if(!require(car)) install.packages("car")                    ; library(car)
if(!require(olsrr)) install.packages("olsrr")                ; library(olsrr)
if(!require(lmtest)) install.packages("lmtest")              ; library(lmtest)
if(!require(rcompanion)) install.packages("rcompanion")      ; library(rcompanion)
if(!require(nortest)) install.packages("nortest")            ; library(nortest)
if(!require(colorspace)) install.packages("colorspace")      ; library(colorspace)
if(!require(broom)) install.packages("broom")                ; library(broom)
if(!require(dplyr)) install.packages("dplyr")                ; library(dplyr)
if(!require(RcmdrMisc)) install.packages("RcmdrMisc")        ; library(RcmdrMisc)
if(!require(caret)) install.packages("caret")                ; library(caret)
if(!require(tidyverse)) install.packages("tidyverse")        ; library(tidyverse)
if(!require(rgl)) install.packages("rgl")                    ; library(rgl)
if(!require(boot)) install.packages("boot")                  ; library(boot)
if(!require(dotwhisker)) install.packages("dotwhisker")      ; library(dotwhisker)

###########
# 1. DATA  #
###########

dt <- read.csv2("LifeExp DATA (very new) 2016.csv", na.strings="")

# Data cleaning
str(dt)

dt$inc_level <- ordered(dt$inc_level, levels = c("Low", "Lower_middle", "Upper_middle", "High")) # Reorder by
income level

# Other variables to numeric
dt <- cbind(dt[, c(1:2)],
        data.frame(lapply(dt[, -c(1:2)], function(x) as.numeric(as.character(x)))))

## Country as row names
rownames(dt) <- dt$country
dt <- dt[, -1] # Omit variable "country"

str(dt)
summary(dt)

####################
# 2. MISSING DATA  #
####################

# Total proportion of missing values
sum(is.na(dt)) / (dim(dt)[1]*dim(dt)[2]) # 1.5 % of values missing
```

```
# Count of missing values per each variable
sort(sapply(dt,function(x)sum(is.na(x))), decreasing = TRUE)

# # Pattern in missing values
# missmap(dt, col=c("indianred1", "peachpuff"))
# ## 2 % of all values missing (out of 183*9)

# Proportion of missing values
( mice_plot <- aggr(dt,col=c("peachpuff", "indianred1"),
                    numbers = TRUE,
                    sortVars = TRUE,
                    labels = names(dt),
                    cex.axis = .7 ) )
## 90.2 % obs with no missing values
## 9.8 % obs with missing combinations of 1-3 variables

## 5 % of 'HepB' values missing
## 4 % of 'GDP_pc' values missing

#####################
# 3. IMPUTING DATA  #
#####################

# Impute missForest
set.seed(713)
fit_missForest <- missForest(dt,
                             maxiter = 10,
                             ntree = 200,
                             replace = TRUE,
                             variablewise = FALSE,
                             verbose = TRUE)

# Save imputed values
# Complete data with logical variables: TRUE (imputed), FALSE (observed))
# ...for plots construction

dt_missForest <- cbind(fit_missForest$ximp,
                       inc_level_imp = ifelse(is.na(dt[,'inc_level']), TRUE, FALSE),
                       life_exp_imp = ifelse(is.na(dt[,'life_exp']), TRUE, FALSE),
                       adult_m_imp = ifelse(is.na(dt[,'adult_m']), TRUE, FALSE),
                       HepB_imp = ifelse(is.na(dt[,'HepB']), TRUE, FALSE),
                       BMI_imp = ifelse(is.na(dt[,'BMI']), TRUE, FALSE),
                       alcohol_imp = ifelse(is.na(dt[,'alcohol']), TRUE, FALSE),
                       pm2.5_imp = ifelse(is.na(dt[,'pm2.5']), TRUE, FALSE),
                       GDP_pc_imp = ifelse(is.na(dt[,'GDP_pc']), TRUE, FALSE),
                       hlth_exp_pc_imp = ifelse(is.na(dt[,'hlth_exp_pc']), TRUE, FALSE) )

dt_new <- dt_missForest[,-c(10:18)]

# Comparison of descriptive stats before & after imputation

# Variables with missing values
stargazer(dt[,c("adult_m", "HepB", "BMI", "alcohol", "pm2.5", "GDP_pc", "hlth_exp_pc")],
        type = "text",
        summary.stat = c("min", "p25", "median", "p75", "max", "sd"), out = "DescStat1.csv")
# Variables with imputed values
stargazer(dt_new[,c("adult_m", "HepB", "BMI", "alcohol", "pm2.5", "GDP_pc", "hlth_exp_pc")],  # Statistics: variables
with missing values
        type = "text",
        summary.stat = c("min", "p25", "median", "p75", "max", "sd"), out = "DescStat2.csv")

#############################################################
# 4. DEFINITION OF RESPONSE & EXPLANATORY VARIABLES #
#############################################################

( s1 <- ggplot(data = dt_new, aes(x = adult_m, y = life_exp)) +
   geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
```

```
  geom_smooth(col = "blue", alpha = 0.3, size = 0.6, span = 0.85) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
  labs(x = "Adult mortality", y = "Life expectancy", col = "Income level") +
  theme_light() +
  theme(axis.text = element_text(size = 25),
      axis.title = element_text(size = 25),
      legend.text  = element_text(size = 25),
      legend.title = element_text(size = 25))+
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
              labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s2 <- ggplot(data = dt_new, aes(x = HepB, y = life_exp)) +
  geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
  geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 0.85) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
  labs(x = "Hepatitis B (immunization)", y = "Life expectancy", col = "Income level") +
  theme_light()  +
  theme(axis.text = element_text(size = 25),
      axis.title = element_text(size = 25),
      legend.text  = element_text(size = 25),
      legend.title = element_text(size = 25))+
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
              labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s3 <- ggplot(data = dt_new, aes(x = BMI, y = life_exp)) +
  geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
  geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 1.2) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
  labs(x = "BMI", y = "Life expectancy", col = "Income level") +
  theme_light() +
  theme(axis.text = element_text(size = 25),
      axis.title = element_text(size = 25),
      legend.text  = element_text(size = 25),
      legend.title = element_text(size = 25))+
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
              labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s4 <- ggplot(data = dt_new, aes(x = alcohol, y = life_exp)) +
  geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
  geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 0.85) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
  labs(x = "Alcohol", y = "Life expectancy", col = "Income level") +
  theme_light() +
  theme(axis.text = element_text(size = 25),
      axis.title = element_text(size = 25),
      legend.text  = element_text(size = 25),
      legend.title = element_text(size = 25))+
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
              labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s5 <- ggplot(data = dt_new, aes(x = pm2.5, y = life_exp)) +
  geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
  geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 0.85) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
  labs(x = "Particulate matter, PM 2.5", y = "Life expectancy", col = "Income level") +
  theme_light() +
  theme(axis.text = element_text(size = 25),
      axis.title = element_text(size = 25),
      legend.text  = element_text(size = 25),
      legend.title = element_text(size = 25))+
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
              labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s6 <- ggplot(data = dt_new, aes(x = GDP_pc, y = life_exp)) +
  geom_jitter(aes(col = inc_level),  alpha = 0.8, size = 5) +
  geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 0.85) +
  geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
```

```
   labs(x = "GDP per capita", y = "Life expectancy", col = "Income level") +
   theme_light() +
   theme(axis.text = element_text(size = 25),
        axis.title = element_text(size = 25),
        legend.text  = element_text(size = 25),
        legend.title = element_text(size = 25))+
   scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
                  labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

( s7 <- ggplot(data = dt_new, aes(x = hlth_exp_pc, y = life_exp)) +
   geom_jitter(aes(col = inc_level), alpha = 0.8, size = 5) +
   geom_smooth(col ="blue", alpha = 0.3, size = 0.6, span = 0.85) +
   geom_smooth(method = "lm", se = FALSE, col = " black", size = 0.6) +
   labs(x = "Health expenditures per capita", y = "Life expectancy", col = "Income level") +
   theme_light() +
   theme(axis.text = element_text(size = 25),
        axis.title = element_text(size = 25),
        legend.text  = element_text(size = 25),
        legend.title = element_text(size = 25)) +
   scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1"),
                   labels = c("Low", "Lower-middle", "Upper-middle", "High")) )

png(file = "3.GDP & Hlth.png", width = 1800, height = 500)
ggarrange(s6,s7,
        ncol = 2,
        nrow = 1,
        common.legend = TRUE,
        legend = "top")
dev.off()

png(file = "4.Adult.png", width = 1300, height = 500); s; dev.off()

png(file = "5.HepB.png", width = 1300, height = 500); s2; dev.off()

png(file = "6.BMI.png", width = 1300, height = 500); s3; dev.off()

png(file = "7.Alco & PM25.png", width = 1800, height = 500)
ggarrange(s4,s5,
        ncol = 2,
        nrow = 1,
        common.legend = TRUE,
        legend = "top")
dev.off()

#############################
# 5. DESCRIPTIVE STATISTICS #
#############################

summary(dt$life_exp[dt$inc_level=="Low"])
summary(dt$life_exp[dt$inc_level=="Lower_middle"])
summary(dt$life_exp[dt$inc_level=="Upper_middle"])
summary(dt$life_exp[dt$inc_level=="High"])

sd(dt$life_exp[dt$inc_level=="Low"])
sd(dt$life_exp[dt$inc_level=="Lower_middle"])
sd(dt$life_exp[dt$inc_level=="Upper_middle"])
sd(dt$life_exp[dt$inc_level=="High"])

skewness(dt$life_exp[dt$inc_level=="Low"])
skewness(dt$life_exp[dt$inc_level=="Lower_middle"])
skewness(dt$life_exp[dt$inc_level=="Upper_middle"])
skewness(dt$life_exp[dt$inc_level=="High"])

kurtosis(dt$life_exp[dt$inc_level=="Low"])
kurtosis(dt$life_exp[dt$inc_level=="Lower_middle"])
kurtosis(dt$life_exp[dt$inc_level=="Upper_middle"])
kurtosis(dt$life_exp[dt$inc_level=="High"])
```

```
# Function detecting outlier
is_outlier <- function(x) {
  return(x < quantile(x, 0.25) - 1.5 * IQR(x) | x > quantile(x, 0.75) + 1.5 * IQR(x))}

p_inc <- dt_new %>%
  tibble::rownames_to_column(var = "outlier") %>%
  group_by(inc_level) %>%
  mutate(is_outlier = ifelse(is_outlier(life_exp), life_exp, as.numeric(NA)))

p_inc$outlier[which(is.na(p_inc$is_outlier))] <- as.numeric(NA)

# Ridgeline and boxplot by income level

life1 <- ggplot(dt_new, aes(x=life_exp, y=inc_level, fill=life_exp)) +
  geom_density_ridges(scale = 1.5, aes(fill=inc_level), quantile_lines = TRUE, quantiles = 2) +
  labs(x = "Life expectancy", y = "Income level") +
  scale_fill_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1")) +
  guides(fill=FALSE) +
  theme_light()+
  theme(axis.text = element_text(size = 9),
       axis.title = element_text(size = 9))

life2 <- ggplot(p_inc, aes(x = inc_level, y = life_exp, fill = inc_level)) +
  geom_boxplot(show.legend = FALSE) +
  geom_text_repel(aes(label = outlier),
                hjust = -.1,
                size = 10,direction = "y") +
  labs(x = "Income level",
      y = "Life expectancy") +
  scale_fill_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1")) +
  theme_light()+
  theme(axis.text = element_text(size = 25),
       axis.title = element_text(size = 25))

   png(file = "2.Exp by income.png", width = 1800, height = 860)
   life2
   dev.off()

ggarrange(life1, life2,
         ncol = 2,
         nrow = 1,
         common.legend = TRUE,
         legend = "top")

( lifehist <- qplot(data = dt_new,
                 x = life_exp, y=(..count..)/(sum(..count..)),
                 fill = ..x..,
                 colour = I("black"), size = I(0.1), alpha = I(0.9),
                 geom = "histogram", bins = 50) +
   geom_vline(xintercept = mean(dt_new$life_exp), lty=1) +
   geom_vline(xintercept = median(dt_new$life_exp), lty=2) +
   labs(x = "Life expectancy at birth", y = "Proportion") +
   scale_y_continuous(labels=scales::percent_format(accuracy = 1), limits = c(0,.09)) +
   scale_x_continuous(limits = c(50, 91)) +
   scale_fill_gradientn(colours=c("turquoise2", "tan2", "olivedrab2", "indianred2"), guide = FALSE) +
   theme_minimal() +
   theme(plot.title = element_text(size = 25),
        axis.title.x = element_text(size = 25),
        axis.title.y = element_text(size = 25),
        axis.text.x = element_text(size = 25),
        axis.text.y = element_text(size = 25)) +
   annotate("text", x = 55, y = 0.08, label = paste("Skewness = -0.473"), size= 8) +
   annotate("text", x = 55, y = 0.07, label = paste("Kurtosis = 2.557"), size= 8) )

( lifehist_log <- qplot(data = dt_new,
                 x = log(life_exp), y=(..count..)/(sum(..count..)),
```

```
                fill = ..x..,
                colour = I("black"), size = I(0.1), alpha = I(0.9),
                geom = "histogram", bins = 50) +
    geom_vline(xintercept = mean(log(dt_new$life_exp)), lty=1) +
    geom_vline(xintercept = median(log(dt_new$life_exp)), lty=2) +
    labs(x = "Logarithm of life expectancy at birth", y = "Proportion") +
    scale_y_continuous(labels=scales::percent_format(accuracy = 1), limits = c(0,.09)) +
    scale_x_continuous() +
    scale_fill_gradientn(colours=c("turquoise2", "tan2", "olivedrab2", "indianred2"), guide = FALSE) +
    theme_minimal() +
    theme(plot.title = element_text(size = 25),
        axis.title.x = element_text(size = 25),
        axis.title.y = element_text(size = 25),
        axis.text.x = element_text(size = 25),
        axis.text.y = element_text(size = 25)) +
    annotate("text", x = 4.1, y = 0.08, label = paste("Skewness = -0.698"), size= 8) +
    annotate("text", x = 4.1, y = 0.07, label = paste("Kurtosis = 2.885"), size= 8) )

png(file = "Life hist.png", width = 1800, height = 600)
ggarrange(lifehist, lifehist_log, ncol=2)
dev.off()

#########
# 6. OLS  #
#########

dt_new <- within(dt_new, inc_level <- relevel(factor(inc_level, ordered=FALSE), ref = "High"))
dt_new$GDP <- dt_new$GDP_pc/1000
dt_new$hlth_exp <- dt_new$hlth_exp_pc/1000

### Model with RAW polynomials
mod_raw=lm(data=dt_new,
         life_exp ~ inc_level +
           adult_m +
           HepB +
           BMI + I(BMI^2) +
           alcohol + I(alcohol^2) +
           log(pm2.5)  +
           GDP + I(GDP^2) +
           hlth_exp + I(hlth_exp^2))

summary(mod_raw)

X_raw = model.matrix(mod_raw)
X_raw_poly = cor(X_raw[,c(7:10,12:15)])
colnames(X_raw_poly) <- c("BMI", "$ BMI^2", "alcohol", "$ alcohol^2", "GDP", "$ GDP^2", "hlth_exp", "$
hlth_exp^2")
rownames(X_raw_poly) <- c("BMI", "$ BMI^2", "alcohol", "$ alcohol^2", "GDP", "$ GDP^2", "hlth_exp", "$
hlth_exp^2")

### Model with ORTHOGONAL polynomials
logpm25 <- log(dt_new$pm2.5)
BMI_v <- poly(dt_new$BMI, 2, raw = FALSE)
alcohol_v <- poly(dt_new$alcohol, 2, raw = FALSE)
GDP_v <- poly(dt_new$GDP, 2, raw = FALSE)
hlth_exp_v <- poly(dt_new$hlth_exp, 2, raw = FALSE)

# mod=lm(data=dt_new,
#        life_exp ~ inc_level +
#          adult_m +
#          HepB +
#          poly(BMI, 2, raw = FALSE) +
#          poly(alcohol, 2, raw = FALSE) +
#          log(pm2.5)  +
#          poly(GDP, 2, raw = FALSE) +
#          poly(hlth_exp, 2, raw = FALSE))
```

```
mod=lm(data=dt_new,
      life_exp ~ inc_level +
        adult_m +
        HepB +
        BMI_v +
        alcohol_v +
        logpm25  +
        GDP_v +
        hlth_exp_v)

write.csv(summary(mod), file = "Summary (OLS orth).csv")

X = model.matrix(mod)
X_poly = cor(X[,c(7:10,12:15)])
colnames(X_poly) <- c("BMI", "$ BMI^2", "alcohol", "$ alcohol^2", "GDP", "$ GDP^2", "hlth_exp", "$ hlth_exp^2")
rownames(X_poly) <- c("BMI", "$ BMI^2", "alcohol", "$ alcohol^2", "GDP", "$ GDP^2", "hlth_exp", "$ hlth_exp^2")

# Compare correlations between second order RAW and ORTHOGONAL polynomials

png(file = "8.Corrplot.png", width = 1600, height = 600)
par(mfrow=c(1,2), oma=c(0,0,4,0))

corrplot(X_raw_poly, method="color", type = "upper", diag = FALSE,
        tl.cex = 1.3, tl.col = "black", tl.srt = 90,
        number.font = 1, number.cex = 1.7,
        col = colorRampPalette(c("turquoise1","white","indianred1"))(200),
        addCoef.col = "black",
        addgrid.col = "gray45",
        number.digits = 3)

corrplot(X_poly, method="color", type = "upper", diag = FALSE,
        tl.cex = 1.3, tl.col = "black", tl.srt = 90,
        number.font = 1.5, number.cex = 1.7,
        col =  colorRampPalette(c("turquoise1","white","indianred1"))(200),
        addCoef.col = "black",
        addgrid.col = "gray45",
        number.digits = 3)

par(mfrow=c(1,1))
dev.off()

# Variance Inflation Factor
VIF <- cbind(ols_vif_tol(mod_raw)$VIF,
           ols_vif_tol(mod)$VIF)
rownames(VIF) <- ols_vif_tol(mod)$Variables
colnames(VIF) <- c("Ordinary polynomial", "Orthogonal polynomial")

write.csv(round(VIF,4), file = "VIF.csv")

# Analysis of Variance
anova(mod)

sum( (fitted(mod) - mean(dt_new$life_exp))^2) # SSE (Explained)
sum( (dt_new$life_exp - fitted(mod))^2) # SSR (Residual)
sum( (dt_new$life_exp - mean(dt_new$life_exp))^2) # SST (Total)

sum( (fitted(mod) - mean(dt_new$life_exp))^2) / 14 # Mean Square (Explained)
sum( (dt_new$life_exp - fitted(mod))^2) / (183-14-1) # Mean Square (Residual)

pf(338.6833, 14, 168, lower.tail = FALSE) # p-value

##########################
# 6.1 OLS residual diagnostics #
##########################
```

```
#-------
# RESET (Regression Specification Error Test)
#-------
alpha=0.05

resettest(mod,
        power = 2:3,
        type = "fitted")
## Reject H0: model suffers from misspesification

mod_RESET <- lm(data = dt_new,
             life_exp ~ inc_level +
               adult_m +
               HepB +
               BMI_v +
               alcohol_v +
               logpm25  +
               GDP_v +
               hlth_exp_v +
               I(fitted(mod)^2) + I(fitted(mod)^3))

F_mod_RESET <- ( (summary(mod_RESET)$r.squared - summary(mod)$r.squared)/2 ) /
  ( (1-summary(mod_RESET)$r.squared) / (183-14-3) )

crit_value_mod_RESET <- qf(1-alpha, 2, 183-14-3) # Critical value

F_mod_RESET >= crit_value_mod_RESET # Rejection region

pf(F_mod_RESET, 2, 183-14-3, lower.tail = FALSE) # p-value

#-------
# Normality tests
#-------
write.csv(rbind(shapiro.test(mod$residuals),
lillie.test(mod$residuals),
cvm.test(mod$residuals),
ad.test(mod$residuals) ), file = "10.OLS norm.csv")

plot(mod,which=2)

( qq <- ggplot(as.data.frame(mod$residuals), aes(sample = mod$residuals)) +
  geom_qq(aes(colour = ..sample..),size = 10, show.legend = FALSE) +
  geom_qq_line() +
  scale_color_gradientn(colours = c("turquoise2", "tan2", "olivedrab2", "indianred2"), guide = FALSE) +
  labs(x = "Theoretical quantiles", y = "Sample quantiles") +
  theme_minimal() +
  theme(plot.title = element_text(size = 7),
       axis.title.x = element_text(size = 25),
       axis.title.y = element_text(size = 25),
       axis.text.x = element_text(size = 25),
       axis.text.y = element_text(size = 25)) )

png(file = "9.QQ residuals.png", width = 1300, height = 500)
qq
dev.off()

#-------
# Heteroskedasticity tests
#-------

# Breusch-Pagan test
ols_test_breusch_pagan(mod)
qchisq(alpha, 14, lower.tail = F)

# Modified White's test
```

```
# Step 1.
u_sq <- (mod$residuals)^2 # Squared residuals --> response
y_hat <- mod$fitted.values # Fitted 'life_exp' --> explanatory
y_hat_sq <- (mod$fitted.values)^2 # Squared fitted 'life_exp' --> explanatory

# Step 2.
eq <- lm(u_sq ~ y_hat + y_hat_sq) # Regress squared residuals on fitted & squared fitted 'life_exp'

R_sq_u <- summary(eq)$r.squared # R-squared

# Step 3.
n <- nrow(dt_new) # Number of observations

# F-statistic
F_stat = (R_sq_u/2) / ((1-R_sq_u)/(n-2-1)) # F-statistic
crit_valueF <- qf(1-alpha, 2, n-3) # Critical value

F_stat >= crit_valueF # Rejection region --> Reject H0: heteroskedastic

pf(F_stat, 2, n-3, lower.tail = FALSE) # p-value --> Reject H0: heteroskedastic

# LM statistic
LM_stat = n* R_sq_u # LM
crit_valueChiSq <- qchisq(alpha, 2, lower.tail = F) # Critical value

LM_stat >= crit_valueChiSq # Rejection region --> Reject H0: heteroskedastic

pchisq(LM_stat, 2, lower.tail = F) # p-value --> Reject H0: heteroskedastic

( heter <- ggplot(aes(x=fitted(mod), y=resid(mod)), data=dt_new) +

   theme_light()  +
   theme(axis.text = element_blank(),
        axis.title = element_text(size = 25),
        legend.text = element_text(size = 25),
        legend.title = element_text(size = 25),
        legend.position = "top")  +
  geom_jitter(show.legend= T, aes(col = inc_level), size = 5, alpha = 0.8) +
   geom_smooth(col ="red", alpha = 0, size = 0.6, span = .5) +
   geom_text_repel(aes(label=ifelse(resid(mod) > 2, as.character(rownames(dt_new)), "")), size=7.5) +
   geom_text_repel(aes(label=ifelse(resid(mod) < -2, as.character(rownames(dt_new)), "")), size=7.5) +
   geom_hline(yintercept = 0, lty=2) +
   geom_hline(yintercept = 2, lty=2) +
   geom_hline(yintercept = -2, lty=2) +
  scale_color_manual(values = c("turquoise3", "tan2", "olivedrab3", "indianred2"),
                 labels = c("Low", "Lower-middle", "Upper-middle", "High")) +

  labs(x = "Fitted values", y = "Residuals",  col = "Income level")  )

png("10.Heterosk.png", width=1700, height = 600)
heter
dev.off()

##########
# 7. FGLS #
##########

u_sq <- (mod$residuals)^2 # Squared residuals --> response

# Steps 1-2. (Squared residuals 'u_sq' obtained from Modified White's test above)
log_u_sq <- log(u_sq) # Logarithm of squared residuals

# Step 3. Regress 'log_u_sq' on explanatory variables
g_mod  <- lm(data=dt_new,
           log_u_sq ~ inc_level +
             adult_m +
             HepB +
```

```
            BMI_v +
            alcohol_v +
            logpm25  +
            GDP_v +
            hlth_exp_v)

g_hat <- g_mod$fitted.values

# Step 4. Exponentiate fitted values 'g_hat'
w_hat <- exp(g_hat)

#############
# 7.1 Weights  #
#############

W <- diag(w_hat)

W_inverse <- solve(W)
P <- sqrt(W_inverse)

head(w_hat, 1)
W_inverse[1,1]
P[1,1]

### RESPONSE: q
q = P %*% dt_new$life_exp
names(q) <- rownames(dt_new)

### EXPLANATORY: Q
Q = P %*% model.matrix(mod)
rownames(Q) <- rownames(dt_new)

### TRANSFORMED DATA FRAME
dt_tr <- as.data.frame(cbind(q, Q))
rownames(dt_tr) <- rownames(dt_new)
colnames(dt_tr) <- c("life_exp",
                "Intercept_non_const",
                "inc_levelLow", "inc_levelLower_middle", "inc_levelUpper_middle",
                "adult_m",
                "HepB",
                "BMI_v1", "BMI_v2",
                "alcohol_v1", "alcohol_v2",
                "logpm25",
                "GDP_v1", "GDP_v2",
                "hlth_exp_v1", "hlth_exp_v2")

cbind(colnames(model.matrix(mod)), colnames(dt_tr[,-1])) # Check matching names

#---------------
# Plot weights distribution by group
#---------------

weight <- as.vector(diag(P)) # Save weights = 1/sqrt(w_hat)

weights <- as.data.frame(cbind(dt$inc_level, weight))
rownames(weights) <- rownames(dt)

colnames(weights)[1] <- "inc_level"

weights$inc_level <- as.factor(weights$inc_level)
levels(weights$inc_level) <- c("Low", "Lower-middle", "Upper-middle", "High")

# Detecting outliers

w_outliers <- weights %>%
  tibble::rownames_to_column(var = "outlier") %>%
  group_by(inc_level) %>%
```

```
  mutate(is_outlier = ifelse(is_outlier(weight), weight, as.numeric(NA)))

w_outliers$outlier[which(is.na(w_outliers$is_outlier))] <- as.numeric(NA)

# (w2 <- ggplot(w_outliers, aes(x=weight, y=inc_level, fill=weight)) +
#     geom_density_ridges(scale = 1.5, aes(fill=inc_level)) +
#     labs(x = "Weight", y = "Income level") +
#     scale_fill_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1")) +
#     guides(fill=FALSE) +
#     geom_vline(xintercept = 1, col = "red", lty = 2) +
#     theme_light()+
#     theme(axis.text = element_text(size = 7),
#           axis.title = element_text(size = 7)) )

(w3 <- ggplot(data = w_outliers, aes(y = weight, x=inc_level)) +
   geom_boxplot(show.legend = FALSE, aes(fill = inc_level), outlier.alpha = 0.6) + coord_flip() +
   labs(x = "Income level", y = "Weight") +
   geom_hline(yintercept = 1, col = "red", lty = 2) +
   theme_light() +
   geom_text_repel(aes(label = outlier),
                   size = 10,
                   direction = "y") +
   theme(axis.text = element_text(size =25),
         axis.title = element_text(size = 25)) +
   scale_fill_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1")))

png("11.Weights.png", width = 1800, height = 600)
w3
dev.off()

# ggarrange(w2, w3,
#           ncol = 2,
#           nrow = 1,
#           common.legend = TRUE,
#           legend = "top")

w_mean <- rbind(mean(weights$weight[weights$inc_level=="High"]),
                mean(weights$weight[weights$inc_level=="Upper-middle"]),
                mean(weights$weight[weights$inc_level=="Lower-middle"]),
                mean(weights$weight[weights$inc_level=="Low"]))

w_median <- rbind(median(weights$weight[weights$inc_level=="High"]),
                  median(weights$weight[weights$inc_level=="Upper-middle"]),
                  median(weights$weight[weights$inc_level=="Lower-middle"]),
                  median(weights$weight[weights$inc_level=="Low"]))

w_mm <- cbind(w_mean, w_median)
rownames(w_mm) <- c("High", "Upper-middle", "Lower-middle", "Low")
colnames(w_mm) <- c("Mean weight", "Median weight")
w_mm

write.csv(round(w_mm,4), file = "Weights.csv")

###########
# 8. FWLS  #
###########

mod2 = lm(data=dt_tr, life_exp ~ 0 + .)

summary(mod2)

anova(mod2)

t(coefficients(mod2)) %*% t(Q) %*% q # SSE (Explained)
t(q) %*% q - t(coefficients(mod2)) %*% t(Q) %*% q # SSR (Residual)
t(q)%*%q # SST (Total)
```

```
(t(coefficients(mod2)) %*% t(Q) %*% q) / 15 # Mean Square (Explained)
(t(q) %*% q - t(coefficients(mod2)) %*% t(Q) %*% q) / (183-14-1) # Mean Square (Residual)

pf(153838.1/3.285219, 14, 168, lower.tail = FALSE) # p-value

sqrt(( t(q) %*% q - t(coefficients(mod2)) %*% t(Q) %*% q   )/168) # Residual SE

(t(coefficients(mod2)) %*% t(Q) %*% q)/ (t(q)%*%q) # Multiple R-squared

1-(1-0.9997609) * (182/168) # Adusted R-squared

anova(lm(life_exp ~ 0, dt_tr) , mod2) # The same result

write.csv(round(cbind(summary(mod2)$coefficients[,1:4]),4),
        file = "FWLS coef.csv")

############################
# 8.1 FWLS residual diagnostics #
############################

#-------
# RESET (Regression Specification Error Test)
#-------
alpha=0.05

resettest(mod2,
        power = 2:3,
        type = "fitted")
## Fail to reject H0: model does not suffer from misspesification

mod2_RESET <- lm(data = dt_tr,
              life_exp ~ 0 + .
              + I(fitted(mod2)^2) + I(fitted(mod2)^3))

F_mod2_RESET <- ( (summary(mod2_RESET)$r.squared - summary(mod2)$r.squared)/2 ) /
  ( (1-summary(mod2_RESET)$r.squared) / (183-14-3) )

crit_value_mod2_RESET <- qf(1-alpha, 2, 183-14-3) # Critical value

F_mod2_RESET >= crit_value_mod2_RESET # Acceptance region --> Fail to reject H0: model does not suffer from
misspesification

pf(F_mod2_RESET, 2, 183-14-3, lower.tail = FALSE) # p-value --> Fail to reject H0: model does not suffer from
misspesification

#-------
# Normality tests
#-------

shapiro.test(mod2$residuals)
lillie.test(mod2$residuals)
cvm.test(mod2$residuals)
ad.test(mod2$residuals)

#-------
# Heteroskedasticity tests
#-------

# Breusch-Pagan test
ols_test_breusch_pagan(mod2)

# Modified White's test

# Step 1.
u_sq2 <- (mod2$residuals)^2 # Squared residuals --> response
y_hat2 <- mod2$fitted.values # Fitted 'life_exp' --> explanatory
y_hat_sq2 <- (mod2$fitted.values)^2 # Squared fitted 'life_exp' --> explanatory
```

```
# Step 2.
eq2 <- lm(u_sq2 ~ y_hat2 + y_hat_sq2) # Regress squared residuals on fitted & squared fitted 'life_exp'

R_sq_u2 <- summary(eq2)$r.squared # R-squred

# Step 3.
n <- nrow(dt_new) # Number of observations

# F-statistic
F_stat2 = (R_sq_u2/2) / ((1-R_sq_u2)/(n-2-1)) # F-statistic
crit_valueF2 <- qf(1-alpha, 2, n-3) # Critical value

F_stat2 >= crit_valueF2 # Rejection region

pf(F_stat2, 2, n-3, lower.tail = FALSE) # p-value

# LM statistic
LM_stat2 = n * R_sq_u2 # LM
crit_valueChiSq2 <- qchisq(alpha, 2, lower.tail = F) # Critical value

LM_stat2 >= crit_valueChiSq2 # Rejection region

pchisq(LM_stat2, 2, lower.tail = F) # p-value

#################
# 8. OLS vs FWLS #
#################

bonfCL_95 <- 1-0.05/15

bonfCL_90 <- 1-0.1/15

summary(mod)
summary(mod2)

cbind(round(summary(mod)$coefficients[,1:2],4),
     round(summary(mod2)$coefficients[,1:2],4))

write.csv( cbind(round(confint(mod, level = .9), 3), round(confint(mod2, level = .9), 3), # OLS, FWLS: individual 90%
CI
              round(confint(mod, level = bonfCL_90), 3), round(confint(mod2, level = bonfCL_90), 3)), # OLS, FWLS:
Bonferroni 90% CI
              file = "Conf Int.csv")

#-------
# Confidence intervals
#-------

# Omit intercepts
m1_df <- tidy(mod, conf.int = TRUE ,conf.level = bonfCL_90)  %>%
  filter(term != "(Intercept)") %>%
  mutate(model = "OLS 90%")

m2_df <- tidy(mod2, conf.int = TRUE, conf.level = bonfCL_90) %>%
  filter(term != "Intercept_non_const") %>%
  mutate(model = "FWLS 90%")

two_models <- rbind(m1_df,m2_df)

brackets <- list(c(" ", "Low", "Upper-middle"),
              c(" ", "adult_m", "adult_m"),
              c(" ", "HepB", "HepB"),
              c(" ", "BMI", "BMI^2"),
              c(" ", "alcohol", "log(pm2.5)"),
              c(" ","GDP", "hlth_exp^2"))
```

```
# Bonferroni simultaneous 90% CI: (1-0.1/15)
png("12.Conf Int plot.png", width = 1800, height = 1000)

{dwplot(two_models,
       dot_args = list(size = 5),
       whisker_args = list(size = 1.3),
       vline = geom_vline(xintercept = 0, colour = "grey60", linetype = 2)) %>%
   relabel_predictors(c(inc_levelLow = "Low",
                        inc_levelLower_middle = "Lower-middle",
                        inc_levelUpper_middle = "Upper-middle",
                        adult_m = "adult_m",
                        HepB = "HepB",
                        BMI_v1 = "BMI",
                        BMI_v2 = "BMI^2",
                        alcohol_v1 = "alcohol",
                        alcohol_v2 = "alcohol^2",
                        logpm25 = "log(pm2.5)",
                        GDP_v1 = "GDP",
                        GDP_v2 = "GDP^2",
                        hlth_exp_v1 = "hlth_exp",
                        hlth_exp_v2 = "hlth_exp^2")) +
   xlab("Coefficient Estimate") +
   scale_color_manual(values = c("indianred3","turquoise3")) +
   theme_classic() +
   theme(legend.position = c(.8,.8),
         legend.justification = c(0, 0),
         axis.title = element_text(size = 25),
         axis.text = element_text(size = 25),
         legend.background = element_rect(colour="grey80"),
         legend.title = element_blank(),
         legend.text = element_text(size = 25))} %>%
  add_brackets(brackets)
dev.off()

#-------
# Confidence regions
#-------
png("13.Ellipse.png", width = 1600, height = 700)

par( mar = c(5,5,4,2))
confidenceEllipse(mod2, which.coef = c("alcohol_v1", "GDP_v1"),
              lwd = 1.3,
              fill=T, fill.alpha = 0,
              grid = FALSE,
              col = "white", levels = .99,
              ylab = "GDP", xlab = "Alcohol",
              cex.axis = 2.5,
              cex.lab = 2.5)
confidenceEllipse(mod2, which.coef = c("alcohol_v1", "GDP_v1"),
              fill=T,fill.alpha = 0.2,
              lwd = 1.3,
              col = "turquoise2", levels = .95,
              add= TRUE)
confidenceEllipse(mod2, which.coef = c("alcohol_v1", "GDP_v1"),
              fill=T,fill.alpha = 0.4,
              lwd = 1.3,
              col = "turquoise3", levels = .90,
              add = TRUE)
confidenceEllipse(mod2, which.coef = c("alcohol_v1", "GDP_v1"),
              fill=T,fill.alpha = .4,
              lwd = 1.3,
              col = "turquoise4", levels = .5,
              add = TRUE)

ci = confint(mod2, level = bonfCL_95)
for(j in 1:2) abline(v=ci[9,j], lty =2, lwd = 1.5)
for(j in 1:2) abline(h=ci[12,j], lty =2, lwd = 1.5)
```

```
legend(x = 3.138003598, y =5.59, legend = c("95 %", "90 %", "50 %"),
       pch = c(16, 16, 16),
       col = c("turquoise2", "turquoise3", "turquoise4"),
       cex = 2.5)

text(y = -4.426624, x = 7.640379,
     labels = "(7.6404; -4.4266)",
     pos = 3,
     cex = 2.5)

points(y = -4.426624, x = 7.640379,
       col = "white", pch = 16, cex = 1.5)

dev.off()

( vcov2 = vcov(mod2)[c(9,12), c(9,12)] )

solve(vcov2)
round(solve(vcov(mod2)[c(9,12), c(9,12)]),4)

qf(.95,2,168) * 2 * vcov2

##################################
# 9. Outliers & Influential Observations  #
##################################

#-------
# Leverage: Hat Values
#-------

hv <- as.data.frame(hatvalues(mod2)) # Hat values
h_bar <- mean(hatvalues(mod2)) # Mean leverage

hv$warn <- ifelse(hv[, 'hatvalues(mod2)'] > 3*h_bar, 'x3',              # Exceeding threshold 3*mean leverage
                  ifelse(hv[, 'hatvalues(mod2)'] > 2*h_bar, 'x2', '-' ))   # Exceeding threshold 2*mean leverage

hv[hv$warn != "-",]
lev <- hv[hv$warn != "-",]
lev[order(lev),] # Sort in ascending order

write.csv(lev[order(lev, decreasing = T),], file = "Hat-values.csv")

# Plot hat-values

png("14.Hat-values.png", width=1800, height = 800)

par( mar = c(5,5,4,2))

plot(hv[,1], type = "h", lwd= 2.5,
     xlab = "Country index",
     ylab = "Hat-value",
     col = "turquoise3",
     cex.lab = 2,
     cex.axis = 2,
     xlim = c(2,182))

legend(-5.1,.962, legend = c(paste(round(3*h_bar,4)), paste(round(2*h_bar,4)), paste(round(h_bar,4))),
       lty = c(1, 2, 2), col = c("indianred3","indianred3", "black"),
       cex = 2)

text(x = c(24,52,77,95,96,103,108,109,133,139,159,166,174,176),
     y = hv[c(24,52,77,95,96,103,108,109,133,139,159,166,174,176), 1],
     labels = names(which(hatvalues(mod2) > 2*h_bar)),
     cex = 2, col = "indianred3",
     pos = 3,
     offset = 0.1)
```

```
abline(h = 2*h_bar, col="indianred3", lty = 2, lwd = 1.5)
abline(h = 3*h_bar, col="indianred3", lwd = 1.5)
abline(h=h_bar, lty=2, lwd = 1.5)

dev.off()

#-------
# Outliers: Externally Studentized Residuals
#-------

rst <- rstudent(mod2) # Externally studentized residuals
names(rst) <- rownames(dt_tr)

which(rst > 2) # Threshold |2|
which(rst < -2)
rst[160]

write.csv( cbind(sort(round(abs(rst[c(14,19,51,86,101,146,159,160,170)]), 4),decreasing = T) ),
        file = "Extern stud.csv")

# Ext. stud. residuals for points with leverage > 2(p/n)
rst[c('Ukraine','Iraq', 'Qatar','Lithuania','Malta')]

# Ext. stud. residuals for points with leverage > 3(p/n)
rst[c('Luxembourg','Moldova','Brunei Darussalam','Equatorial Guinea',
    'Tonga','Samoa','Switzerland','USA',
    'Monaco')]

# Plot ext.stud. residuals

png("15.Extern stud.png", width=1800, height = 800)

par( mar = c(5,5,4,2))
{plot(x = seq(1, 183, by = 1),
    y = rst,
    type = "h", lwd = 2.5,
    xlab = "Country index",
    ylab = "Externally Studentized Residual",
    col = "turquoise3",
    cex.lab = 2,
    cex.axis = 2,
    ylim = c(-4, 4))

text(x = c(19, 51, 146, 159, 160),
    y = rst[c(19, 51, 146, 159, 160)],
    labels = c("Bhutan", "El Salvador", "Singapore",
            "Switzerland", "Syrian Arab Republic"),
    cex = 2, col = "indianred3",
    pos = 3,
    offset = 0.15)

text(x = c(14, 86, 101, 170),
    y = rst[c(14, 86, 101, 170)],
    labels = c("Barbados", "Kiribati",  "Maldives", "Turkmenistan"),
    cex = 2, col = "indianred3",
    pos = 1,
    offset = 0.15)

abline(h = c(-3, 0, 3), col = c("indianred3", "black", "indianred3"), lwd= 1.5)
abline(h = c(-2, 2), col = c("indianred3", "indianred3"), lty = 2, lwd= 1.5)}
axis(2, at=c(-3,-1,1,3), cex.axis = 2)

dev.off()

outlierTest(mod2)
# 2*pt(3.3466, 183-14-2, lower.tail = F)
# 2*pt(3.3466, 183-14-2, lower.tail = F) * 183
```

```
Syria <- rep.int(0,183)
Syria[160] <- 1

outtest <- lm(data=dt_tr, life_exp ~ 0 + . + Syria)
summary(outtest)

Switzerland <- rep.int(0,183)
Switzerland[159] <- 1

outtest <- lm(data=dt_tr, life_exp ~ 0 + . + Switzerland)
summary(outtest)

#-------
# Cook's Distance
#-------

cookD <- cooks.distance(mod2) # Cook's D

round(head(sort(cookD, decreasing = T),15), 4)

( cookD_crit <- qf(0.5, 15, 183-15) ) # Critical value

which(cookD > cookD_crit) # Monaco is influential
which(cookD > 4/(183-14-1))
write.csv(cbind(round(sort(cookD[cookD > 4/(183-14-1)], decreasing = TRUE), 4)),
        file = "Cook D.csv")

( ( (rstandard(mod2)[109])^2 ) / 15 ) * (hatvalues(mod2)[109] / (1-hatvalues(mod2)[109])) # Cook's D for Monaco

mod2_woMonaco <- update(mod2,
                    subset = rownames(dt_tr) != "Monaco")

mod2_woSwitzerland <- update(mod2,
                        subset = rownames(dt_tr) != "Switzerland")

mod2_woMonacoSwitz <- update(mod2,
                    subset = !(rownames(dt_tr) %in%  c("Monaco","Switzerland")))

compcoef_MonacoSwitz <- compareCoefs(mod2, mod2_woMonaco, mod2_woSwitzerland, mod2_woMonacoSwitz,
                                 se = F)


colnames(compcoef_MonacoSwitz) <- c("All in",
                                 "Monaco out", "Switzerland out", "Both out")
write.csv(round(compcoef_MonacoSwitz, 4), file = "Comp Coefs.csv")

cookD_dt <- as.data.frame(cbind(dt$inc_level, cookD, hv[,1], rst))
rownames(cookD_dt) <- rownames(dt)
colnames(cookD_dt)[c(1,3)] <- c("inc_level", "hv")

cookD_dt$inc_level <- as.factor(cookD_dt$inc_level)
levels(cookD_dt$inc_level) <- c("Low", "Lower-middle", "Upper-middle", "High")

png("16.Cook's D.png", width = 1800, height = 700)
ggplot(cookD_dt, aes(x = hv, y = rst, col = inc_level, size = cookD, fill=inc_level)) +
  geom_point(alpha = 0.6) +
  geom_vline(xintercept = 2*h_bar, lty = 2, col = "gray55") +
  geom_vline(xintercept = 3*h_bar, lty = 2, col = "gray55") +
  geom_hline(yintercept = 2, lty = 2, col = "gray55") +
  geom_hline(yintercept = -2, lty = 2, col = "gray55") +
  geom_hline(yintercept = 3, lty = 2, col = "gray55") +
  geom_hline(yintercept = -3, lty = 2, col = "gray55") +
  scale_color_manual(values = c("turquoise2", "tan1", "olivedrab2", "indianred1")) +
  scale_size(range = c(4, 80))  +
  labs(x = "Hat-values", y = "Externally studentized residuals",
     col = "Income level", size = "Cook's D") +
  guides(size = FALSE, fill = FALSE,
```

```
        color = guide_legend(override.aes = list(size=5))) +
  xlim(0,1) +
  scale_y_continuous(breaks=c( -3 ,  -2 ,  -1 ,  0 ,  1 ,  2 ,  3 )) +
  theme_light() +
  theme(axis.text = element_text(size = 25),
        axis.title = element_text(size = 25),
        legend.text = element_text(size = 25),
        legend.title = element_text(size = 25),
        legend.position = "top",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  geom_text_repel(aes(label=ifelse(hv > 3*h_bar, as.character(rownames(dt_new)), "")),
                size = 8, col ="black", direction="y", nudge_x = 0.0, nudge_y = 0) +
  geom_text_repel(aes(label=ifelse(rst > 2, as.character(rownames(dt_new)), "")),
                size = 8, col ="black", direction="y", nudge_x = 0.0, nudge_y = 0) +
  geom_text_repel(aes(label=ifelse(rst < -2, as.character(rownames(dt_new)), "")),
                size = 8, col ="black", direction="y", nudge_x = 0.0, nudge_y = 0)
dev.off()

#-------
# DFFITS
#-------

DFFITs <- abs(dffits(mod2))
( DFFITs_crit <- 2*sqrt(15/(183-15)) )

names(which(DFFITs > DFFITs_crit))

ols_plot_dffits(mod2)

dffits = dffits(mod2)

png("17.DFFITS.png", width=1800, height = 800)

par( mar = c(5,5,4,2))
{plot(x = seq(1, 183, by = 1),
      y = dffits,
      type = "h", lwd = 1.5,
      xlab = "Country",
      ylab = "DFFITS",
      col = "turquoise3",
      cex.lab = 2,
      cex.axis = 2)

abline(h = -DFFITs_crit, col="indianred3")
abline(h = 0, col="grey")
abline(h = DFFITs_crit, col="indianred3")

text(x = c(95, 109, 137, 146, 159, 160, 174),
     y = dffits[c(95, 109, 137, 146, 159, 160, 174)],
     labels = names(which(dffits > DFFITs_crit)),
     cex = 1.7, col = "indianred3",
     pos = 3,
     offset = 0.15)

text(x = c(10, 52, 86, 87, 96, 101, 103, 176),
     y = dffits[c(10, 52, 86, 87, 96, 101, 103, 176)],
     labels = names(which(dffits < -DFFITs_crit)),
     cex = 1.7, col = "indianred3",
     pos = 1,
     offset = 0.15)

legend(-6.1,6.7, legend = paste("|",round(DFFITs_crit, 4),"|"),
       lty = 1, col = "indianred3",
       cex = 2)}
dev.off()
```

```
DFFITs_mat <- as.matrix(sort(DFFITs[DFFITs > DFFITs_crit], decreasing=T))
colnames(DFFITs_mat) <- "| DFFITs |"
DFFITs_mat

write.csv(round(DFFITs_mat, 4), file = "DFFITS.csv")

#-------
# Summarize influence measures
#-------

sum_hv <- hv[c('Monaco', 'Switzerland', 'Syrian Arab Republic',
          'Malta', 'Equatorial Guinea', 'Luxembourg',
          'Kiribati', 'USA', 'Ukraine',
          'Maldives', 'Saint Lucia', 'Singapore',
          'Lithuania', 'Kuwait', 'Azerbaijan'), 1]

sum_rst <- rst[c('Monaco', 'Switzerland', 'Syrian Arab Republic',
           'Malta', 'Equatorial Guinea', 'Luxembourg',
           'Kiribati', 'USA', 'Ukraine',
           'Maldives', 'Saint Lucia', 'Singapore',
           'Lithuania', 'Kuwait', 'Azerbaijan')]

sum_cookD <- cookD[c('Monaco', 'Switzerland', 'Syrian Arab Republic',
                'Malta', 'Equatorial Guinea', 'Luxembourg',
                'Kiribati', 'USA', 'Ukraine',
                'Maldives', 'Saint Lucia', 'Singapore',
                'Lithuania', 'Kuwait', 'Azerbaijan')]

sum_DFFITs <- DFFITs[c('Monaco', 'Switzerland', 'Syrian Arab Republic',
                 'Malta', 'Equatorial Guinea', 'Luxembourg',
                 'Kiribati', 'USA', 'Ukraine',
                 'Maldives', 'Saint Lucia', 'Singapore',
                 'Lithuania', 'Kuwait', 'Azerbaijan')]

summary <- cbind(sum_hv, sum_rst, sum_cookD, sum_DFFITs)
colnames(summary) <- c("Leverage", "Ext. stud. resid", "Cook's D", "| DFFITs |")
summary

write.csv(round(summary,4), file = "Outliers summary.csv")

#-------
# Delete obs. one at a time and follow change in the coefficients
#-------
mod2_woEqGuinea <- update(mod2,
                  subset = rownames(dt_tr) != "Equatorial Guinea")

mod2_woLithuania <- update(mod2,
                  subset = rownames(dt_tr) != "Lithuania")

mod2_woLuxembourg <- update(mod2,
                    subset = rownames(dt_tr) != "Luxembourg")

mod2_woMalta <- update(mod2,
                    subset = rownames(dt_tr) != "Malta")

mod2_woMonaco <- update(mod2,
                  subset = rownames(dt_tr) != "Monaco")

mod2_woSwitzerland <- update(mod2,
                  subset = rownames(dt_tr) != "Switzerland")

mod2_woSyria <- update(mod2,
                 subset = rownames(dt_tr) != "Syrian Arab Republic")

mod2_woUkraine <- update(mod2,
                 subset = rownames(dt_tr) != "Ukraine")
```

```
mod2_woUSA <- update(mod2,
                     subset = rownames(dt_tr) != "USA")

compcoef <- compareCoefs(mod2,
                 mod2_woEqGuinea, mod2_woLithuania, mod2_woLuxembourg,
                 mod2_woMalta, mod2_woMonaco, mod2_woSwitzerland,
                 mod2_woSyria, mod2_woUkraine, mod2_woUSA,

            se = FALSE)

colnames(compcoef) <- c("All", "-GNQ", "-LTU", "-LUX", "-MLT", "-MCO", "-CHE", "-SYR", "-UKR",
                    "-USA")
round(compcoef,4)

################################
# 10. Variable Selection Procedures  #
################################

# Full model
# full <- lm(data=dt_new,
#                     life_exp ~ inc_level +
#                       adult_m +
#                       HepB +
#                       BMI_v +
#                       alcohol_v +
#                       logpm25  +
#                       GDP_v +
#                       hlth_exp_v,
#            weights = diag(W_inverse))

#######
### !!! With Monaco and Switzerland
#######

# full <- lm(data=dt_new,
#           life_exp ~ inc_level +
#             adult_m +
#             HepB +
#             poly(BMI, 2, raw = F) +
#             poly(alcohol, 2, raw = F) +
#             log(pm2.5)  +
#             poly(GDP, 2, raw = F) +
#             poly(hlth_exp, 2, raw = F),
#           weights = diag(W_inverse))

full <- lm(data=dt_new,
             life_exp ~ inc_level +
               adult_m +
               HepB +
               poly(BMI, 2, raw = F) +
               poly(alcohol, 2, raw = F) +
               log(pm2.5)  +
               poly(GDP, 2, raw = F) +
               poly(hlth_exp, 2, raw = F),
             weights = diag(W_inverse))

summary(full)

extractAIC(full)[2]
extractAIC(full, k = log(n))[2]

# Null model (without intercept, but forced to include inc_level)
null <- lm(data = dt_new,
             life_exp ~ inc_level,
          weights = diag(W_inverse))
summary(null)
```

```
extractAIC(null)[2]
extractAIC(null, k = log(n))[2]

#-------
# Backward Elimination
#-------

# AIC
step(full,
     direction = "backward",
     scope = list(lower = null,
                  upper = full),
     trace = 1)

# BIC
step(full,
     direction = "backward",
     scope = list(lower = null,
                  upper = full),
     trace = 1, k = log(n))

# Alpha
ols_step_backward_p(full, prem = 0.001, details = T)
ols_step_backward_p(full, prem = 0.01)
ols_step_backward_p(full, prem = 0.05)
ols_step_backward_p(full, prem = 0.1)

#-------
# Forward Selection
#-------

# AIC
step(null,
     direction = "forward",
     scope = list(lower = null,
                  upper = full),
     trace = 1)

# BIC
step(null,
     direction = "forward",
     scope = list(lower = null,
                  upper = full),
     trace = 1, k = log(n))

# Alpha
ols_step_forward_p(full, penter = 0.001)
ols_step_forward_p(full, penter = 0.01)
ols_step_forward_p(full, penter = 0.05)
ols_step_forward_p(full, penter = 0.1)

#-------
# Stepwise Regression
#-------

# AIC
step(null,
     direction = "both",
     scope = list(lower = null,
                  upper = full),
     trace = 1)

# BIC
step(null,
     direction = "both",
     scope = list(lower = null,
                  upper = full),
```

```
    trace = 1, k = log(n))

# Alpha
ols_step_both_p(full, prem = 0.001, penter = 0.001)
ols_step_both_p(full, prem = 0.01, penter = 0.05)
ols_step_both_p(full, prem = 0.01, penter = 0.01)
ols_step_both_p(full, prem = 0.1, penter = 0.1)

################################
# 11. Cross-Validation & Bootrstrap #
################################

lm1 <- lm(data=dt_tr, life_exp ~ 0 + .)
summary(lm1)

lm2 <- update(lm1, . ~ . - BMI_v1 - BMI_v2)
summary(lm2)

lm3 <- update(lm2, . ~ . - HepB - logpm25)
summary(lm3)

#-------
# Validation set approach (with Monaco and Switzerland)
#-------

# Split the data into training and test set
set.seed(713)

training.samples <- dt_tr$life_exp %>%
                  createDataPartition(p = .8, list = FALSE)

train.data  <- dt_tr[training.samples, ]
test.data <- dt_tr[-training.samples, ]

# Build the model on training data
lm1_SET <- update(lm1, data = train.data)
lm2_SET <- update(lm2, data = train.data)
lm3_SET <- update(lm3, data = train.data)

# Make predictions and compute the R2, RMSE and MAE
predictions1 <- lm1_SET %>% predict(test.data)
predictions2 <- lm2_SET %>% predict(test.data)
predictions3 <- lm3_SET %>% predict(test.data)

data.frame( R2 = R2(predictions1, test.data$life_exp),
          RMSE = RMSE(predictions1, test.data$life_exp),
          MAE = MAE(predictions1, test.data$life_exp))

data.frame( R2 = R2(predictions2, test.data$life_exp),
          RMSE = RMSE(predictions2, test.data$life_exp),
          MAE = MAE(predictions2, test.data$life_exp))

data.frame( R2 = R2(predictions3, test.data$life_exp),
          RMSE = RMSE(predictions3, test.data$life_exp),
          MAE = MAE(predictions3, test.data$life_exp))

RMSE(predictions1, test.data$life_exp)/mean(test.data$life_exp)
RMSE(predictions2, test.data$life_exp)/mean(test.data$life_exp)
RMSE(predictions3, test.data$life_exp)/mean(test.data$life_exp)

par(mfrow=c(1,3))
plot(predictions1, test.data$life_exp)
abline(0,1)
plot(predictions2, test.data$life_exp)
abline(0,1)
plot(predictions3, test.data$life_exp)
abline(0,1)
```

```
# #-------
# # Validation set approach (without Monaco and Switzerland)
# #-------

# dt_woMS <- dt_tr[-c(109,159),]
#
# lm1 <- lm(data=dt_woMS, life_exp ~ 0 + .)
# summary(lm1)
#
# lm2 <- update(lm1, . ~ . - BMI_v1 - BMI_v2)
# summary(lm2)
#
# lm3 <- update(lm2, . ~ . - HepB - logpm25)
# summary(lm3)
#
# # Split the data into training and test set
# set.seed(713)
#
# training.samples <- dt_woMS$life_exp %>%
#   createDataPartition(p = .8, list = FALSE)
#
# train.data  <- dt_woMS[training.samples, ]
# test.data <- dt_woMS[-training.samples, ]
#
# # Build the model on training data
# lm1_SET <- update(lm1, data = train.data)
# lm2_SET <- update(lm2, data = train.data)
# lm3_SET <- update(lm3, data = train.data)
#
# # Make predictions and compute the R2, RMSE and MAE
# predictions1 <- lm1_SET %>% predict(test.data)
# predictions2 <- lm2_SET %>% predict(test.data)
# predictions3 <- lm3_SET %>% predict(test.data)
#
# data.frame( R2 = R2(predictions1, test.data$life_exp),
#             RMSE = RMSE(predictions1, test.data$life_exp),
#             MAE = MAE(predictions1, test.data$life_exp))
#
# data.frame( R2 = R2(predictions2, test.data$life_exp),
#             RMSE = RMSE(predictions2, test.data$life_exp),
#             MAE = MAE(predictions2, test.data$life_exp))
#
# data.frame( R2 = R2(predictions3, test.data$life_exp),
#             RMSE = RMSE(predictions3, test.data$life_exp),
#             MAE = MAE(predictions3, test.data$life_exp))

#-------
# Leave one out cross validation - LOOCV (with Monaco and Switzerland)
#-------

# Define training control
train.control_LOO <- trainControl(method = "LOOCV")

# Train the model
lm1_LOO <- train(life_exp ~ 0 + .,
             data=dt_tr,
             method = "lm",
             trControl = train.control_LOO)

lm2_LOO <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
               data=dt_tr,
               method = "lm",
               trControl = train.control_LOO)

lm3_LOO <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
               data=dt_tr,
               method = "lm",
```

```
                trControl = train.control_LOO)

# Summarize the results
print(lm1_LOO)
print(lm2_LOO)
print(lm3_LOO)

# #-------
# # Leave one out cross validation - LOOCV (without Monaco and Switzerland)
# #-------
#
# # Define training control
# train.control_LOO <- trainControl(method = "LOOCV")
#
# # Train the model
# lm1_LOO <- train(life_exp ~ 0 + .,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_LOO)
#
# lm2_LOO <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_LOO)
#
# lm3_LOO <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_LOO)
#
# # Summarize the results
# print(lm1_LOO)
# print(lm2_LOO)
# print(lm3_LOO)

#-------
# K-fold cross-validation (with Monaco and Switzerland)
#-------

# Define training control
set.seed(713)

train.control_KF <- trainControl(method = "cv", number = 10)

# Train the model
lm1_KF <- train(life_exp ~ 0 + .,
                data=dt_tr,
                method = "lm",
                trControl = train.control_KF)

lm2_KF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
                data=dt_tr,
                method = "lm",
                trControl = train.control_KF)

lm3_KF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
                data=dt_tr,
                method = "lm",
                trControl = train.control_KF)

# Summarize the results
print(lm1_KF)
print(lm2_KF)
print(lm3_KF)
```

```
# #-------
# # K-fold cross-validation (without Monaco and Switzerland)
# #-------
#
# # Define training control
# set.seed(713)
#
# train.control_KF <- trainControl(method = "cv", number = 10)
#
# # Train the model
# lm1_KF <- train(life_exp ~ 0 + .,
#                 data=dt_woMS,
#                 method = "lm",
#                 trControl = train.control_KF)
#
# lm2_KF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
#                 data=dt_woMS,
#                 method = "lm",
#                 trControl = train.control_KF)
#
# lm3_KF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
#                 data=dt_woMS,
#                 method = "lm",
#                 trControl = train.control_KF)
#
# # Summarize the results
# print(lm1_KF)
# print(lm2_KF)
# print(lm3_KF)

#-------
# Repeated K-fold cross-validation (with Monaco and Switzerland)
#-------

# Define training control
set.seed(713)

train.control_RKF <- trainControl(method = "repeatedcv",
                        number = 10, repeats = 3)

# Train the model
lm1_RKF <- train(life_exp ~ 0 + .,
            data=dt_tr,
            method = "lm",
            trControl = train.control_RKF)

lm2_RKF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
            data=dt_tr,
            method = "lm",
            trControl = train.control_RKF)

lm3_RKF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
            data=dt_tr,
            method = "lm",
            trControl = train.control_RKF)

# Summarize the results
print(lm1_RKF)
print(lm2_RKF)
print(lm3_RKF)

# #-------
# # Repeated K-fold cross-validation (without Monaco and Switzerland)
# #-------
#
# # Define training control
# set.seed(713)
```

```
# train.control_RKF <- trainControl(method = "repeatedcv",
#                                   number = 10, repeats = 3)
#
# # Train the model
# lm1_RKF <- train(life_exp ~ 0 + .,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_RKF)
#
# lm2_RKF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_RKF)
#
# lm3_RKF <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
#                  data=dt_woMS,
#                  method = "lm",
#                  trControl = train.control_RKF)
#
# # Summarize the results
# print(lm1_RKF)
# print(lm2_RKF)
# print(lm3_RKF)

#-------
# Bootstrap (with Monaco and Switzerland)
#-------

# Define training control
set.seed(713)
train.control_BOOT <- trainControl(method = "boot", number = 100)

# Train the model
lm1_BOOT <- train(life_exp ~ 0 + .,
                  data=dt_tr,
                  method = "lm",
                  trControl = train.control_BOOT)

lm2_BOOT <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
                  data=dt_tr,
                  method = "lm",
                  trControl = train.control_BOOT)

lm3_BOOT <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
                  data=dt_tr,
                  method = "lm",
                  trControl = train.control_BOOT)

# Summarize the results
print(lm1_BOOT)
print(lm2_BOOT)
print(lm3_BOOT)

# Create function that returns regression coefficients
model_coef <- function(data, index){
  coef(lm(life_exp ~ 0 + ., data = data, subset = index))}

model_coef(dt_tr, 1:183)                # lm1
model_coef(dt_tr[, -c(8,9)], 1:183)     # lm2
model_coef(dt_tr[, -c(7,8,9,12)], 1:183)  # lm3
# Compute standard errors of 500 bootstrap estimates
set.seed(713)
boootlm1 <- boot(dt_tr, model_coef, 500, sim = "parametric")          # lm1
cbind(round(summary(lm1)$coefficients[,c(1,2)],3))          # lm1 (OLS standard errors)
plot(boootlm1, index = 4)

set.seed(713)
```

```
boootlm2 <- boot(dt_tr[, -c(8,9)], model_coef, 500)        # lm2
cbind(round(summary(lm2)$coefficients[,2],3))               # lm2 (OLS standard errors)
plot(boootlm2, index = 4)

set.seed(713)
boootlm3 <- boot(dt_tr[, -c(7,8,9,12)], model_coef, 500)   # lm3
cbind(round(summary(lm3)$coefficients[,2],3))               # lm3 (OLS standard errors)
plot(boootlm3, index = 4)

write.csv(cbind(round(summary(lm1)$coefficients[,c(1,2)],3)), file = "Bootcoefs1.csv")
write.csv(cbind(round(summary(lm2)$coefficients[,2],3)), file = "Bootcoefs2.csv")
write.csv(cbind(round(summary(lm3)$coefficients[,2],3)), file = "Bootcoefs3.csv")

#-------
# Bootstrap (without Monaco and Switzerland)
#-------

# Define training control
set.seed(713)
train.control_BOOT <- trainControl(method = "boot", number = 500)

# Train the model
lm1_BOOT <- train(life_exp ~ 0 + .,
                  data=dt_woMS,
                  method = "lm",
                  trControl = train.control_BOOT)

lm2_BOOT <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2,
                  data=dt_woMS,
                  method = "lm",
                  trControl = train.control_BOOT)

lm3_BOOT <- train(life_exp ~ 0 + . - BMI_v1 - BMI_v2 - HepB - logpm25,
                  data=dt_woMS,
                  method = "lm",
                  trControl = train.control_BOOT)

# Summarize the results
print(lm1_BOOT)
print(lm2_BOOT)
print(lm3_BOOT)

# Create function that returns regression coefficients
model_coef <- function(data, index){
  coef(lm(life_exp ~ 0 + ., data = data, subset = index))}

model_coef(dt_woMS, 1:181)                # lm1
model_coef(dt_woMS[, -c(8,9)], 1:181)        # lm2
model_coef(dt_woMS[, -c(7,8,9,12)], 1:181)   # lm3

# Compute standard errors of 500 bootstrap estimates
set.seed(713)
boot(dt_woMS, model_coef, 500)                # lm1
cbind(summary(lm1)$coefficients[,2])               # lm1 (OLS standard errors)

set.seed(713)
boot(dt_woMS[, -c(8,9)], model_coef, 500)        # lm2
cbind(summary(lm2)$coefficients[,2])               # lm2 (OLS standard errors)

set.seed(713)
boot(dt_woMS[, -c(7,8,9,12)], model_coef, 500)   # lm3
cbind(summary(lm3)$coefficients[,2])               # lm3 (OLS standard errors)
```