Cleaning Instructions (Manual data clean in Excel)

**Country Characteristic Data Frames**

1. **Polity\_IV**
   1. I sorted all data by year and deleted any entries prior to 1946 as there is no data in USAID to match.
   2. I created a concatenated unique id [=CONCATENATE(country,year)] for all rows. This is the first column
   3. I found two duplicate entries by comparing replications in the “year” and country” columns and deleted them (Yugoslavia 1991 and Eritrea 1993).
   4. I deleted all unnecessary variable columns – every column except the following:
      1. unique\_id
      2. scode
      3. country
      4. year
      5. democ
      6. autoc
      7. polity
   5. Saved to a new unique file with RKEDITS in name
2. **GDP per Capita (World Bank)**
   1. World bank data was queried via their website query function:
      1. Indicator= GDP per capita(constant 2010 US$)
      2. Countries = Sub-Saharan Africa; Choose ALL
      3. Time = 2000-2015
      4. Print output to Excel
   2. In excel I created a concatenated unique id [=CONCATENATE(country,year)] for all rows. This is the first column
   3. I renamed the columns as follows for ease of use:
      1. Country Name: name
      2. Country Code: code
      3. Time: year
      4. GDP per capita(constant 2010 US$)[NY.GDP.PCAP.KD]: gdp\_per\_capita
   4. Saved to a new unique file with RKEDITS in name
3. **Human Freedom Index**
   1. I created a concatenated unique id [=CONCATENATE(Countries,Year)] for all rows. This is the first column
   2. I filtered the data frame by Region==’Sub-Saharan Africa’. I deleted rows that did not fit this filter.
   3. I deleted all unnecessary variable columns – every column except the following:
      1. unique\_id
      2. Year
      3. ISO Code
      4. Countries
      5. Region
      6. human\_freedom\_score
      7. human\_freedom\_rank
      8. personal\_freedom\_score
      9. personal\_freedom\_rank
      10. economic\_freedom\_score
      11. economic\_freedom\_rank
   4. Saved to a new unique file with RKEDITS in name

**Aid Obligation Data Frames**

1. **OECD**
   1. OECD.stat Query Function via their website:
      1. Search Data by theme🡪Development🡪Flows by Provider and Recipient🡪GeoBook: ODA by sector
      2. Donor: USA
      3. Recipient: Customize selection – check Countries under South of Sahara
   2. Unmerge all and delete Row 1-4 as it’s unnecessary heading. Delete all empty columns post merge
   3. As the ‘Year’ column was exported in merged format (ie. for all countries in 2005 there is only one entry for 2005 in the ‘Year’ column) I drug the year to fill the column in. Repeat this process for 2005-2017
   4. I created a concatenated unique id [=CONCATENATE(Recipient,Year)] for all rows. This is the first column
   5. Renamed columns for ease of use in python:
      1. Recipient: Country
      2. Bilateral ODA: bilat\_oda
      3. Social Infrastructure & Services: social\_infrastructure
      4. Education: education
      5. Water supply and sanitation: water\_supply\_sanitation
      6. Economic Infrastructure and Services: economic\_infrastructure
      7. Energy: energy
      8. Transport and Communications: transport
      9. Production Sectors: production
      10. Agriculture, forestry and fishing: ag\_forestry\_fishing
      11. Industry, mining and construction: industry\_mining\_construction
      12. Trade and tourism: trade\_tourism
      13. Multisector: multi
      14. Programme Assistance: program\_assistance
      15. Food Aid: food\_aid
      16. Action Relating to Debt: debt
      17. Humanitarian Aid: humanitarian\_aid
      18. Unallocated/unspecified: unspecified
   6. As OECD export has a ‘..’ value for any sector that received $0, I performed a find & replace all [find: .., replace all: 0].
      1. In the future this should be automated in Python instead of prior cleaning.
2. **AidData – Chinese Global Finance**
   1. Filters:
      1. ‘umbrella’ column==0. Delete all umbrella==1
         1. REASON: per double counting explanation in AidData README file (see ‘Data Dictionary and other Docs for README)
      2. recipient\_region ==Africa; delete all others
      3. recommended\_for\_research==TRUE; delete all others
         1. REASON: per explanation in AidData README file (see ‘Data Dictionary and other Docs for README)
      4. usd\_defl\_2014==delete if blank
         1. REASON: redundancies and additional detail unnecessary in this analysis (ie. multiple lines for one project that spans multiple regions such as a rail line). Value is captured in one row only
      5. flow\_class==ODA-like
         1. REASON: OOF-like and Vague (Official Finance) are unnecessary in this analysis comparing US ODA to Chinese ODA
   2. I created a concatenated unique id [=CONCATENATE(Country,Year)] for all rows. This is the first column
   3. Create Pivot Table to create a data set version that is similar in structure to the OECD for comparability
      1. Columns==’unique\_id’, ‘Country’, ‘Year’, ‘crs\_sector\_name’
      2. Values== ‘usd\_defl\_2014’
   4. Add a ‘Total’ column that sums across each row for total ODA-like flow by country and year
   5. Delete all unnecessary variable columns except:
      1. unique\_id
      2. Country
      3. Year
      4. Action Relating to Debt
      5. Agriculture, Forestry and Fishing
      6. Business and Other Services
      7. Communication
      8. Developmental Food Aid/Food Security Assistance
      9. Education
      10. Emergency Response
      11. Energy Generation and Support
      12. General Budget Support
      13. General Environmental Protection
      14. Government and Civil Society
      15. Health
      16. Industry, Mining, Construction
      17. Non-food commodity assistance
      18. Other Multisector
      19. Other Social Infrastructure and services
      20. Population Policies/Programmes and Reproductive Health
      21. Support to Non-governmental Organizations (NGOs) and Government Organizations
      22. Trade and Tourism
      23. Transport and storage
      24. Unallocated/Unspecified
      25. Water Supply and Sanitation
      26. Women in Development
      27. Total