Penn World Tables - Example

Datasets

François Geerolf

Contents

Introduction

```
rm(list = ls())
pklist <- c("tidyverse", "curl", "readstata13", "maps")
source("https://fgeerolf.github.io/code/load-packages.R")
options(tibble.print_max = 100)
load("pwt.9.0.long.RData")
```

Information

User Guide on the PWT 9.0: https://www.rug.nl/ggdc/docs/user_guide_to_pwt90_data_files.pdf

Variable names

```
pwt.9.0.long %>%
  select(variable, variable.desc1) %>%
  unique
```

```
## # A tibble: 43 x 2
##    variable  variable.desc1
##    <fct>     <fct>
##  1 rgdpe     Expenditure-side real GDP at chained PPPs (in mil. 2011US$)
##  2 rgdpo     Output-side real GDP at chained PPPs (in mil. 2011US$)
##  3 pop       Population (in millions)
```

```
##  4 emp       Number of persons engaged (in millions)
##  5 avh       Average annual hours worked by persons engaged (source: The ~
##  6 hc        Human capital index, see note hc
##  7 ccon      Real consumption of households and government, at current PP~
##  8 cda       Real domestic absorption, see note cda
##  9 cgdpe     Expenditure-side real GDP at current PPPs (in mil. 2011US$)
## 10 cgdpo     Output-side real GDP at current PPPs (in mil. 2011US$)
## 11 ck        Capital stock at current PPPs (in mil. 2011US$)
## 12 ctfp      TFP level at current PPPs (USA=1)
## 13 cwtfp     Welfare-relevant TFP levels at current PPPs (USA=1)
## 14 rgdpna    Real GDP at constant 2011 national prices (in mil. 2011US$)
## 15 rconna    Real consumption at constant 2011 national prices (in mil. 2~
## 16 rdana     Real domestic absorption at constant 2011 national prices (i~
## 17 rkna      Capital stock at constant 2011 national prices (in mil. 2011~
## 18 rtfpna    TFP at constant national prices (2011=1)
## 19 rwtfpna   Welfare-relevant TFP at constant national prices (2011=1)
## 20 labsh     Share of labour compensation in GDP at current national pric~
## 21 delta     Average depreciation rate of the capital stock
## 22 xr        Exchange rate, national currency/USD (market+estimated)
## 23 pl_con    Price level of CCON (PPP/XR), price level of USA GDPo in 201~
## 24 pl_da     Price level of CDA (PPP/XR), price level of USA GDPo in 2011~
## 25 pl_gdpo   Price level of CGDPo (PPP/XR),  price level of USA GDPo in 2~
## 26 i_cig     0/1/2, see note i_cig
## 27 i_xm      0/1/2, see note i_xm
## 28 i_xr      0/1: the exchange rate is market-based (0) or estimated (1)
## 29 i_outlier 0/1, see note i_outlier
## 30 cor_exp   Correlation between expenditure shares, see note cor_exp
## 31 statcap   Statistical capacity indicator (source: World Bank, developi~
## 32 csh_c     Share of household consumption at current PPPs
## 33 csh_i     Share of gross capital formation at current PPPs
## 34 csh_g     Share of government consumption at current PPPs
## 35 csh_x     Share of merchandise exports at current PPPs
## 36 csh_m     Share of merchandise imports at current PPPs
## 37 csh_r     Share of residual trade and GDP statistical discrepancy at c~
## 38 pl_c      Price level of household consumption,  price level of USA GD~
## 39 pl_i      Price level of capital formation, price level of USA GDPo in~
## 40 pl_g      Price level of government consumption, price level of USA GD~
## 41 pl_x      Price level of exports, price level of USA GDPo in 2011=1
## 42 pl_m      Price level of imports, price level of USA GDPo in 2011=1
## 43 pl_k      Price level of the capital stock, price level of USA 2011=1
```

Glance in France

```
pwt.9.0.long %>%
  filter(year == "2014",
         countrycode == "FRA",
         variable %in% c("cgdpo", "ck", "delta", "pop", "emp")) %>%
  select(variable, variable.desc1, value)
```

```
## # A tibble: 5 x 3
##   variable variable.desc1                                        value
##   <fct>    <fct>                                                 <dbl>
## 1 pop      Population (in millions)                             6.61e+1
```

```
## 2 emp      Number of persons engaged (in millions)                2.73e+1
## 3 cgdpo    Output-side real GDP at current PPPs (in mil. 201~     2.52e+6
## 4 ck       Capital stock at current PPPs (in mil. 2011US$)        1.21e+7
## 5 delta    Average depreciation rate of the capital stock         3.78e-2
```

Glance in United States

```
pwt.9.0.long %>%
  filter(year == "2014",
         countrycode == "USA",
         variable %in% c("cgdpo", "ck", "delta", "pop", "emp")) %>%
  select(variable, variable.desc1, value)
```

```
## # A tibble: 5 x 3
##   variable variable.desc1                                          value
##   <fct>    <fct>                                                   <dbl>
## 1 pop      Population (in millions)                              3.19e+2
## 2 emp      Number of persons engaged (in millions)               1.48e+2
## 3 cgdpo    Output-side real GDP at current PPPs (in mil. 201~    1.65e+7
## 4 ck       Capital stock at current PPPs (in mil. 2011US$)       5.28e+7
## 5 delta    Average depreciation rate of the capital stock        4.70e-2
```

Gross and net returns

```
pwt.9.0.long %>%
  filter(year == "2014",
         variable %in% c("cgdpo", "ck", "delta", "pop", "emp", "labsh")) %>%
  select(countrycode, country, variable, value) %>%
  spread(variable, value) %>%
  mutate(ck_gdp = ck / cgdpo,
         cgdpo_pop = cgdpo/pop,
         cgdpo_emp = cgdpo/emp,
         ck_emp = ck/emp,
         ret_gross = 100*(1-labsh)/ck_gdp,
         delta = 100*delta,
         ret_net = ret_gross - delta) %>%
  select(country, pop, cgdpo_pop, ck_gdp, ret_gross, delta, ret_net) %>%
  # Advanced Economies
  filter(cgdpo_pop >= 30000,
         pop >= 1) %>%
  mutate_at(vars(-country), funs(round(., digits = 2)))
```

```
## # A tibble: 31 x 7
##    country                pop cgdpo_pop ck_gdp ret_gross delta ret_net
##    <fct>                <dbl>     <dbl>  <dbl>     <dbl> <dbl>   <dbl>
##  1 United Arab Emirates  9.09    68021.   3.93     NA     5.84   NA
##  2 Australia            23.6     43590.   3.83     11.2   3.65    7.6
##  3 Austria               8.52    45158.   4.92      8.37  4.39    3.98
##  4 Belgium              11.2     38894.   5.27      7.03  4.47    2.56
##  5 Bahrain               1.36    38531.   4.31     16.6   3.34   13.2
##  6 Canada               35.6     42794.   4         9.51  3.63    5.88
```

```
##  7 Switzerland            8.21    61570.   3.77      9.17  5.44    3.73
##  8 Germany               80.6     46190.   4.07      9.28  3.8     5.48
##  9 Denmark                5.65    43733.   4.37      8.23  4.3     3.93
## 10 Spain                 46.3     32454.   5.62      7.48  3.86    3.62
## 11 Finland                5.48    37818.   4.84      8.05  3.97    4.08
## 12 France                66.1     38169.   4.79      7.73  3.78    3.95
## 13 United Kingdom        64.3     38324.   4.79      8.09  3.68    4.41
## 14 China, Hong Kong SAR   7.23    45399.   5.01      9.56  3.89    5.67
## 15 Ireland                4.68    51224.   4.14     12.4   5.1     7.33
## 16 Israel                 7.94    31242.   3.21     14.5   4.45   10.0
## 17 Italy                 59.8     34920.   6.24      7.38  3.67    3.71
## 18 Japan                127.      35271.   4.08      9.74  4.62    5.12
## 19 Republic of Korea     50.1     34585.   3.95     12.2   5.01    7.16
## 20 Kuwait                 3.75    67432.   1.95     38.7   5.9    32.8
## 21 Netherlands           16.9     47392.   4.39      9.22  4.16    5.05
## 22 Norway                 5.15    75920.   3.31     14.1   3.9    10.2
## 23 New Zealand            4.5     33713.   3.03     14.3   3.84   10.4
## 24 Oman                   4.24    36933.   3.47     20.1   6.24   13.9
## 25 Qatar                  2.17   146037.   2.96     27.3  10.4    16.8
## 26 Saudi Arabia          30.9     46772.   3.69     19.5   5.47   14.0
## 27 Singapore              5.51    66050.   4.42     12.7   5.04    7.64
## 28 Sweden                 9.7     42117.   4.34     10.0   4.51    5.5
## 29 Trinidad and Tobago    1.35    30985.   4.65     14.3   4.47    9.82
## 30 Taiwan                23.4     41514.   3.51     14.8   6.14    8.72
## 31 United States        319.      51623.   3.2      12.4   4.7     7.66
```

```
pwt.9.0.long %>%
  filter(year == "2014",
         variable %in% c("cgdpo", "ck", "delta", "pop", "emp", "labsh")) %>%
  select(countrycode, country, variable, value) %>%
  spread(variable, value) %>%
  mutate(ck_gdp = ck / cgdpo,
         cgdpo_pop = cgdpo/pop,
         cgdpo_emp = cgdpo/emp,
         ck_emp = ck/emp,
         capsh = 100*(1-labsh),
         ret_gross = capsh/ck_gdp,
         delta = 100*delta,
         ret_net = ret_gross - delta) %>%
  select(country, pop, cgdpo_pop, capsh, ck_gdp, ret_gross, delta, ret_net) %>%
  # Advanced Economies
  filter(cgdpo_pop >= 30000,
         pop >= 1) %>%
  mutate_at(vars(-country), funs(round(., digits = 2)))
```

```
## # A tibble: 31 x 8
##    country              pop cgdpo_pop capsh ck_gdp ret_gross delta ret_net
##    <fct>              <dbl>     <dbl> <dbl>  <dbl>     <dbl> <dbl>   <dbl>
##  1 United Arab Emir~   9.09    68021.  NA     3.93     NA     5.84   NA
##  2 Australia          23.6     43590.  43.1   3.83     11.2   3.65    7.6
##  3 Austria             8.52    45158.  41.1   4.92      8.37  4.39    3.98
##  4 Belgium            11.2     38894.  37.0   5.27      7.03  4.47    2.56
##  5 Bahrain             1.36    38531.  71.4   4.31     16.6   3.34   13.2
##  6 Canada             35.6     42794.  38.1   4         9.51  3.63    5.88
##  7 Switzerland         8.21    61570.  34.6   3.77      9.17  5.44    3.73
```

```
##  8 Germany            80.6     46190.  37.7   4.07      9.28  3.8     5.48
##  9 Denmark             5.65    43733.  36.0   4.37      8.23  4.3     3.93
## 10 Spain              46.3     32454.  42.0   5.62      7.48  3.86    3.62
## 11 Finland             5.48    37818.  39.0   4.84      8.05  3.97    4.08
## 12 France             66.1     38169.  37     4.79      7.73  3.78    3.95
## 13 United Kingdom     64.3     38324.  38.7   4.79      8.09  3.68    4.41
## 14 China, Hong Kong~   7.23    45399.  47.9   5.01      9.56  3.89    5.67
## 15 Ireland             4.68    51224.  51.5   4.14     12.4   5.1     7.33
## 16 Israel              7.94    31242.  46.4   3.21     14.5   4.45   10.0
## 17 Italy              59.8     34920.  46.0   6.24      7.38  3.67    3.71
## 18 Japan             127.      35271.  39.7   4.08      9.74  4.62    5.12
## 19 Republic of Korea  50.1     34585.  48.1   3.95     12.2   5.01    7.16
## 20 Kuwait              3.75    67432.  75.5   1.95     38.7   5.9    32.8
## 21 Netherlands        16.9     47392.  40.4   4.39      9.22  4.16    5.05
## 22 Norway              5.15    75920.  46.6   3.31     14.1   3.9    10.2
## 23 New Zealand         4.5     33713.  43.3   3.03     14.3   3.84   10.4
## 24 Oman                4.24    36933.  69.7   3.47     20.1   6.24   13.9
## 25 Qatar               2.17   146037.  80.7   2.96     27.3  10.4    16.8
## 26 Saudi Arabia       30.9     46772.  72.0   3.69     19.5   5.47   14.0
## 27 Singapore           5.51    66050.  56.0   4.42     12.7   5.04    7.64
## 28 Sweden              9.7     42117.  43.4   4.34     10.0   4.51    5.5
## 29 Trinidad and Tob~   1.35    30985.  66.5   4.65     14.3   4.47    9.82
## 30 Taiwan             23.4     41514.  52.1   3.51     14.8   6.14    8.72
## 31 United States     319.      51623.  39.6   3.2      12.4   4.7     7.66
```

Depreciation Rates for Different Assets

At the level of 9 assets, the capital stock deflator equals the investment deflator and the depreciation rate are chosen exogenously and remain constant over time. The depreciation rates are as follows:
- residential structures 1.1%
- non-residential structures 3.1%,
- computers 31.5%,
- communication equipment 11.5%,
- other machinery 12.6%,
- transport equipment 18.9%,
- software 31.5%,
- other intellectual property products 15%
- cultivated assets 12.6%.

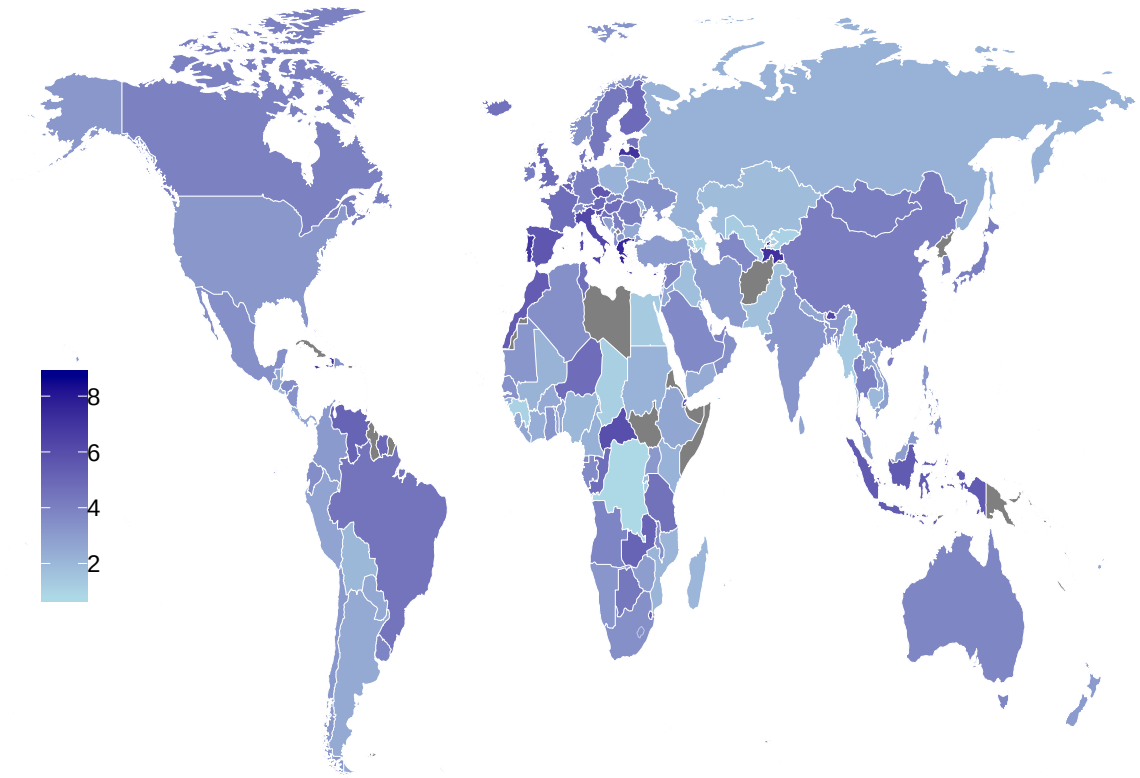
By country maps

Capital / output ratios by country

```
map_data("world") %>%
  filter(region != "Greenland", region != "Antarctica") %>%
  left_join(iso3166 %>%
              select(region = mapname, countrycode = a3) %>%
              mutate(region = ifelse(region == "China(?!:Hong Kong|:Macao)", "China", region),
                     region = ifelse(region == "Finland(?!:Aland)", "Finland", region),
                     region = ifelse(region == "UK(?!r)", "UK", region),
```

```
                    region = ifelse(region == "Norway(?!:Bouvet|:Svalbard|:Jan Mayen)", "Norway", regi
          by = "region") %>%
  left_join(pwt.9.0.long %>%
              filter(year == "2014",
                     variable %in% c("cgdpo", "ck", "delta", "pop", "emp", "labsh")) %>%
              select(countrycode, country, variable, value) %>%
              spread(variable, value) %>%
              mutate(ck_gdp = ck / cgdpo,
                     cgdpo_pop = cgdpo/pop,
                     cgdpo_emp = cgdpo/emp,
                     ck_emp = ck/emp,
                     capsh = 100*(1-labsh),
                     ret_gross = capsh/ck_gdp,
                     delta = 100*delta,
                     ret_net = ret_gross - delta) %>%
              filter(ck_gdp <= 10) %>%
              select(country, countrycode, pop, cgdpo_pop, capsh, ck_gdp, ret_gross, delta, ret_net),
            by = "countrycode") %>%
  ggplot(aes(long, lat, group = group)) +
  geom_polygon(aes(fill = ck_gdp),
               colour = alpha("white", 1/2),
               size = 0.1)  +
  scale_fill_continuous(low="lightblue", high="darkblue", guide="colorbar") +
  theme_void() +
  theme(legend.position = c(0.1, 0.4),
        legend.title = element_blank())
```

```
## Warning: Column `countrycode` joining character vector and factor, coercing
## into character vector
```



Computing Environment

```
Sys.time()
```

```
## [1] "2018-10-12 17:27:00 PDT"
```

```
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2    maps_3.3.0        readstata13_0.9.2
##  [4] curl_3.2          forcats_0.3.0     stringr_1.3.1
##  [7] dplyr_0.7.6       purrr_0.2.5       readr_1.1.1
## [10] tidyr_0.8.1       tibble_1.4.2      ggplot2_3.0.0
```

```
## [13] tidyverse_1.2.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.4 haven_1.1.2      lattice_0.20-35  colorspace_1.3-2
##  [5] htmltools_0.3.6  yaml_2.2.0       utf8_1.1.4       rlang_0.2.2
##  [9] pillar_1.3.0     glue_1.3.0       withr_2.1.2      modelr_0.1.2
## [13] readxl_1.1.0     bindr_0.1.1      plyr_1.8.4       munsell_0.5.0
## [17] gtable_0.2.0     cellranger_1.1.0 rvest_0.3.2      evaluate_0.11
## [21] labeling_0.3     knitr_1.20       fansi_0.3.0      broom_0.5.0
## [25] Rcpp_0.12.18     scales_1.0.0     backports_1.1.2  jsonlite_1.5
## [29] hms_0.4.2        digest_0.6.15    stringi_1.2.4    grid_3.5.1
## [33] rprojroot_1.3-2  cli_1.0.0        tools_3.5.1      magrittr_1.5
## [37] lazyeval_0.2.1   crayon_1.3.4     pkgconfig_2.0.2  xml2_1.2.0
## [41] lubridate_1.7.4  assertthat_0.2.0 rmarkdown_1.10   httr_1.3.1
## [45] rstudioapi_0.7   R6_2.2.2         nlme_3.1-137     compiler_3.5.1
```