R skills assessment

GPID Team, The World Bank

Joseph Zahar

Basic Stats

1. Summary statistics of GDP per capita by region

```
weighted_sd <- function(x, pop) {
  sqrt(weighted.mean((x - weighted.mean(x, pop, na.rm = TRUE))^2,
                     pop, na.rm = TRUE))
}
gdp_sum <- wdi[, .(
  N = sum(!is.na(gdp)),
  Mean = weighted_sd(gdp, pop),
  SD = sqrt(weighted.mean((gdp - weighted.mean(gdp, pop, na.rm = TRUE))^2,
                          pop, na.rm = TRUE)),
  Min = min(gdp, na.rm = TRUE),
  Max = max(gdp, na.rm = TRUE)
), by = .(region, date)]

setorder(gdp_sum, region, date)
setnames(gdp_sum, "date", "year")
correct_q1 <- readr::read_rds(paste0(data_url, "wdi_summ_out.Rds"))
# waldo::compare(correct_q1, gdp_sum)
# datatable(gdp_sum, options = list(pageLength = 10))
gdp_sum
```

```
                region year  N     Mean       SD      Min      Max
1: East Asia & Pacific 1990 22 8496.536 8496.536 581.6133 32846.39
2: East Asia & Pacific 1991 22 8690.758 8690.758 579.3788 33870.37
3: East Asia & Pacific 1992 22 8666.693 8666.693 597.2022 34048.78
4: East Asia & Pacific 1993 22 8576.397 8576.397 635.1072 33782.74
```

```
  5: East Asia & Pacific 1994 22 8619.321 8619.321 669.3751 34053.52
 ---
214:  Sub-Saharan Africa 2015 44 3212.804 3212.804 781.5793 25961.03
215:  Sub-Saharan Africa 2016 44 3151.939 3151.939 764.3366 26923.73
216:  Sub-Saharan Africa 2017 44 3128.585 3128.585 750.7876 27336.61
217:  Sub-Saharan Africa 2018 44 3104.802 3104.802 740.4482 28081.38
218:  Sub-Saharan Africa 2019 44 3058.614 3058.614 729.6585 29190.55
```

2. Aggregate Stats

```r
agg_stats <- wdi[, .(
  mean_lifeex = weighted.mean(lifeex, pop, na.rm = TRUE),
  sd_lifeex = weighted_sd(lifeex, pop),
  min_lifeex = min(lifeex, na.rm = TRUE),
  max_lifeex = max(lifeex, na.rm = TRUE),
  median_lifeex = weighted.median(lifeex, pop, na.rm = TRUE),
  mean_gdp = weighted.mean(gdp, pop, na.rm = TRUE),
  sd_gdp = weighted_sd(gdp, pop),
  min_gdp = min(gdp, na.rm = TRUE),
  max_gdp = max(gdp, na.rm = TRUE),
  median_gdp = weighted.median(gdp, pop, na.rm = TRUE),
  mean_pov_intl = weighted.mean(pov_intl, pop, na.rm = TRUE),
  sd_pov_intl = weighted_sd(pov_intl),
  min_pov_intl = min(pov_intl, na.rm = TRUE),
  max_pov_intl = max(pov_intl, na.rm = TRUE),
  median_pov_intl = weighted.median(pov_intl, pop, na.rm = TRUE),
  pop = sum(pop, na.rm = TRUE)
),
by = .(region, date)
]

agg_stats <- melt(agg_stats,
  id.vars = c("region", "date", "pop"),
  measure.vars = list(
    c("mean_lifeex", "sd_lifeex", "min_lifeex", "max_lifeex", "median_lifeex"),
    c("mean_gdp", "sd_gdp", "min_gdp", "max_gdp", "median_gdp"),
    c("mean_pov_intl", "sd_pov_intl", "min_pov_intl", "max_pov_intl",
      "median_pov_intl")
  ),
  variable.name = "estimate", value.name = c("lifeex", "gdp", "pov_intl")
)
```

```r
agg_stats[, estimate := factor(estimate, labels = c("mean", "sd", "min", "max",
                                                    "median"))]
setorder(agg_stats, estimate, region, date)
agg_stats <- agg_stats[, c(4, 1, 2, 3, 5, 6, 7)]

correct_q2 <- readr::read_rds(paste0(data_url, "wdi_agg_out.Rds"))
# waldo::compare(correct_q2, agg_stats)

# datatable(agg_stats, options = list(pageLength = 10))
agg_stats
```

```
     estimate              region date        pop   lifeex      gdp  pov_intl
   1:     mean East Asia & Pacific 1990 1754166013 68.19770 4913.103 0.5897045
   2:     mean East Asia & Pacific 1991 1779284317 68.41732 5105.010 0.5731783
   3:     mean East Asia & Pacific 1992 1802946756 68.89536 5290.810 0.5495899
   4:     mean East Asia & Pacific 1993 1825777375 69.34064 5482.790 0.5234072
   5:     mean East Asia & Pacific 1994 1848480100 69.62833 5740.088 0.4830632
  ---
1086:   median  Sub-Saharan Africa 2015  990247914 60.43600 2737.729 0.3045670
1087:   median  Sub-Saharan Africa 2016 1017098928 60.94250 2534.193 0.3208582
1088:   median  Sub-Saharan Africa 2017 1044173426 61.61650 2420.599 0.3140970
1089:   median  Sub-Saharan Africa 2018 1072045414 61.91600 2506.419 0.3120162
1090:   median  Sub-Saharan Africa 2019 1100515900 62.39950 2561.465 0.3090027
```

3. Find outliers

```r
is_outlier_cols <- function(dt, col) {
  new_col_ll <- paste0("ll_", col)
  new_col_hl <- paste0("hl_", col)
  mean_col <- paste0("mean_", col)
  sd_col <- paste0("sd_", col)
  dt[, (new_col_ll) := get(col) < get(mean_col) - 2.5 * get(sd_col),
     by = 1:nrow(dt)]
  dt[, (new_col_hl) := get(col) > get(mean_col) + 2.5 * get(sd_col),
     by = 1:nrow(dt)]

  return(dt)
}

temp_dt <- wdi[, .(
```

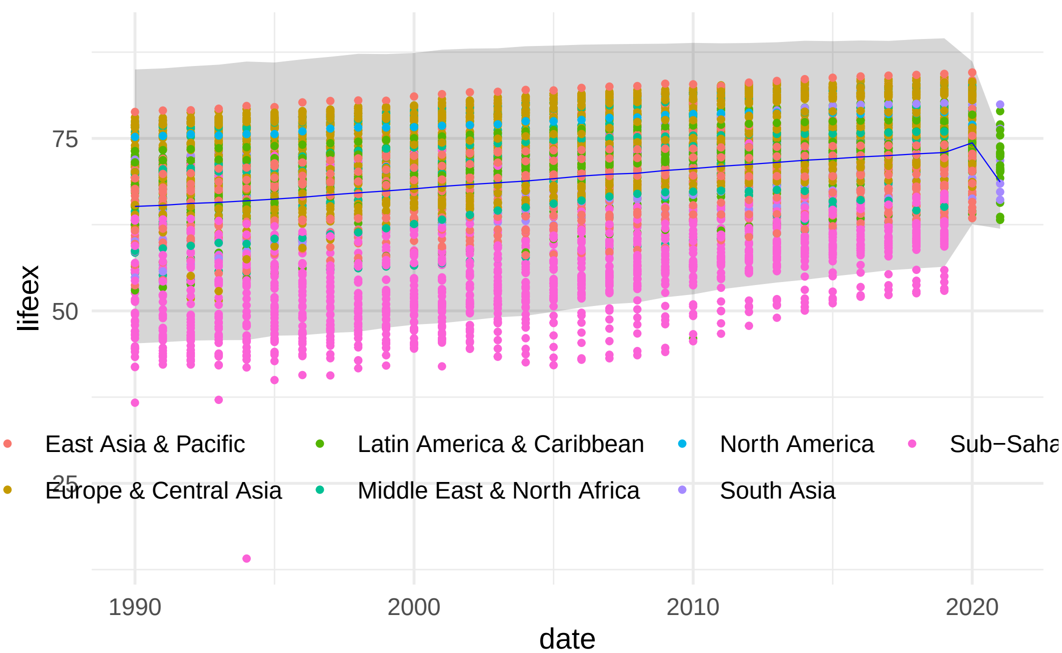
```
    mean_lifeex = weighted.mean(lifeex, pop, na.rm = TRUE),
    sd_lifeex = weighted_sd(lifeex, pop),
    mean_gdp = weighted.mean(gdp, pop, na.rm = TRUE),
    sd_gdp = weighted_sd(gdp, pop),
    mean_gini = weighted.mean(gini, pop, na.rm = TRUE),
    sd_gini = weighted_sd(gini, pop)
  ), by = .(date)]

outliers_dt <- merge(y = temp_dt, x = wdi, by = c("date"), all.x = TRUE)
setorder(outliers_dt, iso3c, date, -region)

for (col in c("lifeex", "gdp", "gini")) {
  outliers_dt <- is_outlier_cols(outliers_dt, col)
}

correct_q3 <- readr::read_rds(paste0(data_url, "wdi_outliers_out.Rds"))
outliers_dt <- outliers_dt[, colnames(correct_q3), with = FALSE]
# waldo::compare(correct_q3, outliers_dt)

# datatable(outliers_dt, options = list(pageLength = 10))
outliers_dt
```

```
                  region iso3c date  country pov_ofcl      gdp gini lifeex
   1: Sub-Saharan Africa   AGO 1990   Angola       NA 5793.085   NA 41.893
   2: Sub-Saharan Africa   AGO 1991   Angola       NA 5659.119   NA 43.813
   3: Sub-Saharan Africa   AGO 1992   Angola       NA 5158.384   NA 42.209
   4: Sub-Saharan Africa   AGO 1993   Angola       NA 3799.195   NA 42.101
   5: Sub-Saharan Africa   AGO 1994   Angola       NA 3728.886   NA 43.422
  ---
5025: Sub-Saharan Africa   ZWE 2015 Zimbabwe       NA 2313.879   NA 59.591
5026: Sub-Saharan Africa   ZWE 2016 Zimbabwe       NA 2286.624   NA 60.306
5027: Sub-Saharan Africa   ZWE 2017 Zimbabwe       NA 2331.781 44.3 60.709
5028: Sub-Saharan Africa   ZWE 2018 Zimbabwe       NA 2399.622   NA 61.414
5029: Sub-Saharan Africa   ZWE 2019 Zimbabwe       NA 2203.397 50.3 61.292
           pop  pov_intl  pov_lmic  pov_umic mean_lifeex sd_lifeex hl_lifeex
   1: 11828638 0.1652797 0.3093024 0.5843191    65.13871  7.941912     FALSE
   2: 12228691 0.1680163 0.3142586 0.5963407    65.30392  7.937230     FALSE
   3: 12632507 0.1919029 0.3537655 0.6382768    65.57109  7.953732     FALSE
   4: 13038270 0.2736178 0.4874785 0.7609357    65.72071  7.985901     FALSE
   5: 13462031 0.2789797 0.4950096 0.7659570    65.95488  8.075165     FALSE
  ---
5025: 14154937 0.2857660 0.5502272 0.8045123    72.04576  6.819779     FALSE
```

```
5026: 14452704 0.3221182 0.5903596 0.8265666    72.30952  6.753262     FALSE
5027: 14751101 0.3420605 0.6158357 0.8410902    72.50898  6.649733     FALSE
5028: 15052184 0.3396693 0.6042348 0.8316396    72.75578  6.643447     FALSE
5029: 15354608 0.3975453 0.6450986 0.8501632    72.95224  6.623025     FALSE
      ll_lifeex  mean_gdp   sd_gdp hl_gdp ll_gdp mean_gini  sd_gini hl_gini
   1:      TRUE  9566.977 12598.52  FALSE  FALSE  35.80927 7.621505      NA
   2:      TRUE  9510.650 12531.81  FALSE  FALSE  34.84978 6.331182      NA
   3:      TRUE  9492.389 12537.99  FALSE  FALSE  41.83976 8.899886      NA
   4:      TRUE  9489.380 12479.77  FALSE  FALSE  35.40868 7.161314      NA
   5:      TRUE  9599.831 12690.65  FALSE  FALSE  40.98136 6.867103      NA
  ---
5025:     FALSE 15220.731 15157.66  FALSE  FALSE  37.56216 4.923590      NA
5026:     FALSE 15550.686 15250.47  FALSE  FALSE  37.71794 5.119278      NA
5027:     FALSE 15965.417 15476.97  FALSE  FALSE  37.92075 4.860761   FALSE
5028:     FALSE 16379.375 15713.16  FALSE  FALSE  37.56201 5.213651      NA
5029:     FALSE 16689.970 15891.68  FALSE  FALSE  37.57519 5.062162    TRUE
      ll_gini
   1:      NA
   2:      NA
   3:      NA
   4:      NA
   5:      NA
  ---
5025:      NA
5026:      NA
5027:   FALSE
5028:      NA
5029:   FALSE
```

```
outlier_cols <- function(dt, col) {
  new_col_ll <- paste0("lo_ci_", col)
  new_col_hl <- paste0("hi_ci_", col)
  mean_col <- paste0("mean_", col)
  sd_col <- paste0("sd_", col)
  dt[, (new_col_ll) := get(mean_col) - 2.5 * get(sd_col), by = 1:nrow(dt)]
  dt[, (new_col_hl) := get(mean_col) + 2.5 * get(sd_col), by = 1:nrow(dt)]

  return(dt)
}
outliers_dt_2 <- unique(outlier_cols(outliers_dt, "lifeex"),
                        by = c("date", "lo_ci_lifeex", "hi_ci_lifeex"))
```

```r
ggplot(data = outliers_dt, aes(x = date, y = lifeex)) +
  geom_ribbon(data = outliers_dt_2, aes(x = date, ymin = lo_ci_lifeex,
                                        ymax = hi_ci_lifeex), alpha = 0.2) +
  geom_point(aes(color = region), size = 0.8) +
  geom_line(aes(x = date, y = mean_lifeex), color = "blue", linewidth = 0.2) +
  theme_minimal() +
  theme(
    legend.position = c(0.5, 0.1),
    legend.justification = c(0.5, 0),
    legend.direction = "horizontal",
    legend.title = element_blank(),
    legend.background = element_blank(),
    legend.box.background = element_blank()
  )
```



Simulated data

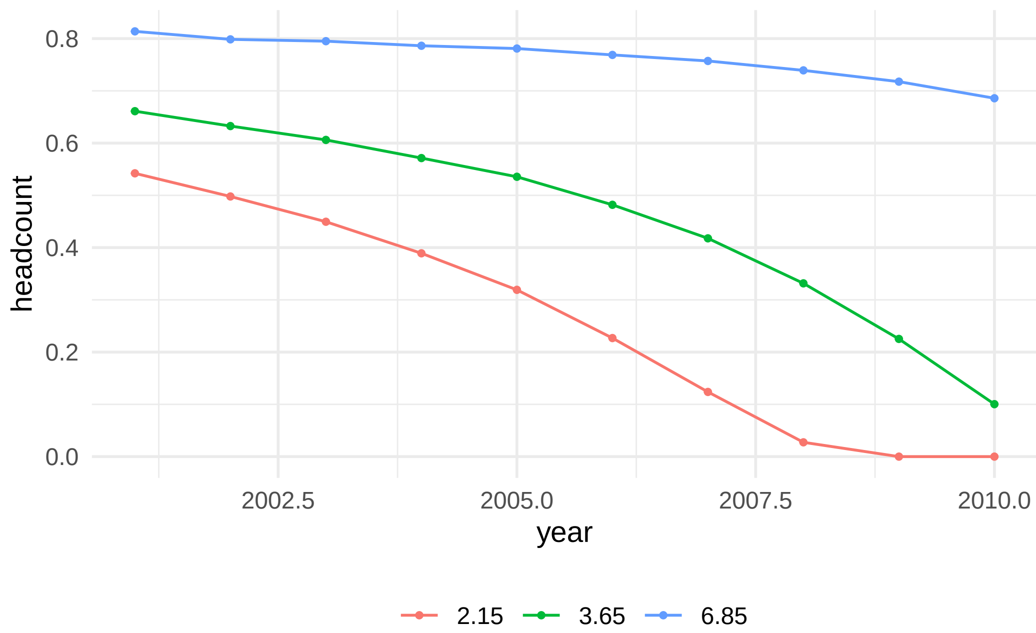
4. Poverty measures

```r
cols <- c("year", "pov_line", "headcount", "povgap", "povseverity")
pov_dt <- data.table(matrix(ncol = length(cols), nrow = 0))
setnames(pov_dt, cols)

FGT <- function(pov_line, year, dt) {
  N <- sum(dt$weight)
  dt[, `:=`(FGTi = (pov_line - income) / pov_line)]
  dt_subset <- dt[income <= pov_line]
  FGT0 <- sum(dt_subset$weight * dt_subset$FGTi^0) / N
  FGT1 <- sum(dt_subset$weight * dt_subset$FGTi^1) / N
  FGT2 <- sum(dt_subset$weight * dt_subset$FGTi^2) / N

  new_data <- data.table(year = year, pov_line = pov_line, headcount = FGT0,
                         povgap = FGT1, povseverity = FGT2)
  pov_dt <<- rbindlist(list(pov_dt, new_data), use.names = TRUE, fill = TRUE)
}

year <- 2001
for (dt in svy_sim) {
  FGT(2.15, year, dt)
  FGT(3.65, year, dt)
  FGT(6.85, year, dt)
  year <- year + 1
}

correct_q4 <- readr::read_rds(paste0(data_url, "dt_pov_out.Rds"))
# waldo::compare(correct_q4, pov_dt)
# datatable(pov_dt, options = list(pageLength = 10))
pov_dt
```

```
   year pov_line    headcount       povgap  povseverity
1: 2001     2.15 5.422254e-01 4.228365e-01 3.798612e-01
2: 2001     3.65 6.611328e-01 4.975328e-01 4.352643e-01
3: 2001     6.85 8.138747e-01 6.139778e-01 5.287430e-01
4: 2002     2.15 4.978546e-01 3.613057e-01 3.129150e-01
5: 2002     3.65 6.326504e-01 4.470285e-01 3.759358e-01
6: 2002     6.85 7.985686e-01 5.774394e-01 4.818298e-01
```

```
 7: 2003     2.15 4.495065e-01 2.949849e-01 2.407590e-01
 8: 2003     3.65 6.061122e-01 3.927487e-01 3.119177e-01
 9: 2003     6.85 7.951019e-01 5.422696e-01 4.328688e-01
10: 2004     2.15 3.891313e-01 2.162681e-01 1.575609e-01
11: 2004     3.65 5.713907e-01 3.271271e-01 2.363397e-01
12: 2004     6.85 7.863333e-01 4.981569e-01 3.735382e-01
13: 2005     2.15 3.191814e-01 1.342803e-01 7.730057e-02
14: 2005     3.65 5.355700e-01 2.577392e-01 1.591762e-01
15: 2005     6.85 7.810200e-01 4.536021e-01 3.124131e-01
16: 2006     2.15 2.269120e-01 6.553235e-02 2.660092e-02
17: 2006     3.65 4.819491e-01 1.870087e-01 9.473435e-02
18: 2006     6.85 7.688042e-01 4.029516e-01 2.514004e-01
19: 2007     2.15 1.237929e-01 2.140403e-02 5.398057e-03
20: 2007     3.65 4.177703e-01 1.256204e-01 5.030210e-02
21: 2007     6.85 7.572493e-01 3.545981e-01 1.990952e-01
22: 2008     2.15 2.737478e-02 1.988324e-03 2.208799e-04
23: 2008     3.65 3.316249e-01 7.139247e-02 2.100636e-02
24: 2008     6.85 7.392448e-01 3.050429e-01 1.516430e-01
25: 2009     2.15 3.282336e-07 7.073100e-10 2.139115e-12
26: 2009     3.65 2.252440e-01 3.132202e-02 6.112839e-03
27: 2009     6.85 7.176459e-01 2.579189e-01 1.122589e-01
28: 2010     2.15 0.000000e+00 0.000000e+00 0.000000e+00
29: 2010     3.65 1.003947e-01 7.446393e-03 8.115501e-04
30: 2010     6.85 6.858685e-01 2.091219e-01 7.828692e-02
    year pov_line    headcount       povgap  povseverity
```

```r
ggplot(data = pov_dt, aes(x = year, y = headcount, group = pov_line,
                          color = as.factor(pov_line))) +
  geom_line(linewidth = 0.5) +
  geom_point(size = 0.8) +
  theme_minimal() +
  theme(
    legend.position = "bottom",
    legend.direction = "horizontal",
    legend.title = element_blank(),
    legend.background = element_blank(),
    legend.box.background = element_blank()
  )
```



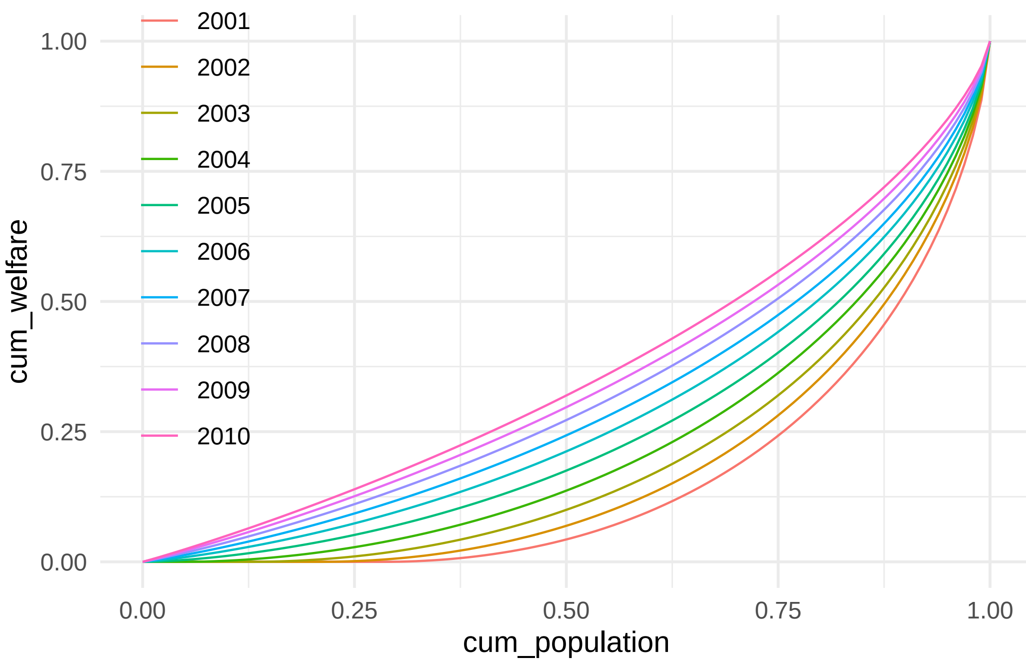
5. Lorenz curve

```
cols <- c("welfare", "cum_welfare", "cum_population", "year", "bin")
lorenz_dt <- data.table(matrix(ncol = length(cols), nrow = 0))
setnames(lorenz_dt, cols)

Lorenz <- function(dt, year) {
  dt <- dt[order(dt$income), ]
  dt$cum_pop <- cumsum(dt$weight) / sum(dt$weight)
  dt$cum_welfare <- cumsum(dt$weight * dt$income) / sum(dt$weight * dt$income)
  dt$welfare <- cumsum(dt$weight * dt$income)


  approx_points <- approx(dt$cum_pop, dt$cum_welfare, n = 100)
  income_val <- sapply(approx_points$x, function(x) {
    idx <- which.min(abs(dt$cum_pop - x))
    return(dt$income[idx])
  })

  new_data <- data.frame(welfare = income_val, cum_welfare = approx_points$y,
                         cum_population = approx_points$x, year = year, bin = 1:100)
```

```r
  lorenz_dt <<- rbindlist(list(lorenz_dt, new_data), use.names = TRUE,
                          fill = TRUE)
}

year <- 2001
for (dt in svy_sim) {
  Lorenz(dt, year)
  year <- year + 1
}

correct_q5 <- readr::read_rds(paste0(data_url, "dt_lorenz_out.Rds"))
# waldo::compare(correct_q4, pov_dt)
# datatable(lorenz_dt, options = list(pageLength = 10))
lorenz_dt
```

```
        welfare cum_welfare cum_population year bin
   1:   0.00000   0.0000000   2.989371e-08 2001   1
   2:   0.00000   0.0000000   1.010104e-02 2001   2
   3:   0.00000   0.0000000   2.020205e-02 2001   3
   4:   0.00000   0.0000000   3.030306e-02 2001   4
   5:   0.00000   0.0000000   4.040407e-02 2001   5
  ---
 996:  14.93882   0.8712323   9.595960e-01 2010  96
 997:  16.42707   0.8949060   9.696970e-01 2010  97
 998:  18.67449   0.9212143   9.797980e-01 2010  98
 999:  23.04845   0.9523613   9.898990e-01 2010  99
1000: 171.48122   1.0000000   1.000000e+00 2010 100
```

```r
ggplot(data = lorenz_dt, aes(x = cum_population, y = cum_welfare, group = year,
                             color = as.factor(year))) +
  geom_line(linewidth = 0.4) +
  theme_minimal() +
  theme(
    legend.position = c(0.1, 0.2),
    legend.justification = c(0.5, 0),
    legend.direction = "vertical",
    legend.title = element_blank(),
    legend.background = element_blank(),
    legend.box.background = element_blank()
  )
```



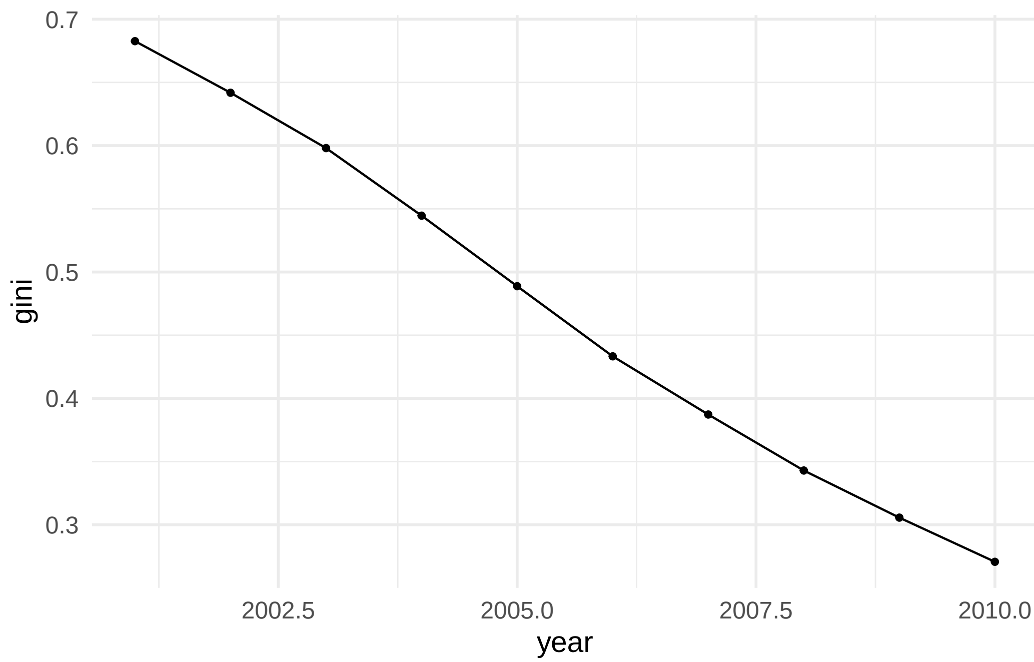
6. Gini coefficient

```
cols <- c("year", "gini")
gini_dt <- data.table(matrix(ncol = length(cols), nrow = 0))
setnames(gini_dt, cols)

Gini <- function(dt, years) {
  dt <- dt[year == years]
  setorder(dt, bin)
  A <- 0
  for (i in 2:length(dt$cum_pop)) {
    width <- dt$cum_pop[i] - dt$cum_pop[i - 1]
    height_avg <- (dt$cum_welfare[i] + dt$cum_welfare[i - 1]) / 2
    A <- A + (width * height_avg)
  }


  gini_index <- 1 - 2 * A
  new_data <- data.frame(year = years, gini = gini_index)
  gini_dt <<- rbindlist(list(gini_dt, new_data), use.names = TRUE, fill = TRUE)
}
```

```r
year <- 2001
for (i in 1:10) {
  Gini(lorenz_dt, year)
  year <- year + 1
}

correct_q6 <- readr::read_rds(paste0(data_url, "dt_gini_out.Rds"))
# waldo::compare(correct_q6, gini_dt)
# datatable(gini_dt, options = list(pageLength = 10))
gini_dt
```

```
     year      gini
 1: 2001 0.6826469
 2: 2002 0.6418104
 3: 2003 0.5980288
 4: 2004 0.5445630
 5: 2005 0.4887706
 6: 2006 0.4332867
 7: 2007 0.3872429
 8: 2008 0.3429453
 9: 2009 0.3056660
10: 2010 0.2707043
```

```r
ggplot(data = gini_dt, aes(x = year, y = gini)) +
  geom_line(linewidth = 0.4) +
  geom_point(size = 0.8) +
  theme_minimal()
```


0.7
0.6
0.5
0.4
0.3
gini
2002.5
2005.0
2007.5
2010.0
year