Oppgave-5–Innlevering_1-.R

falkakselfalkum

2022-09-06

```
##Oppgave 5

#Download the Excel file "GCIPrawdatatest.xlsx".
#I have taken away data from Norway 1980-1990 as it was faulty
#Save it in an easily accessible location, such as a folder on your Desktop or in your personal folder.
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

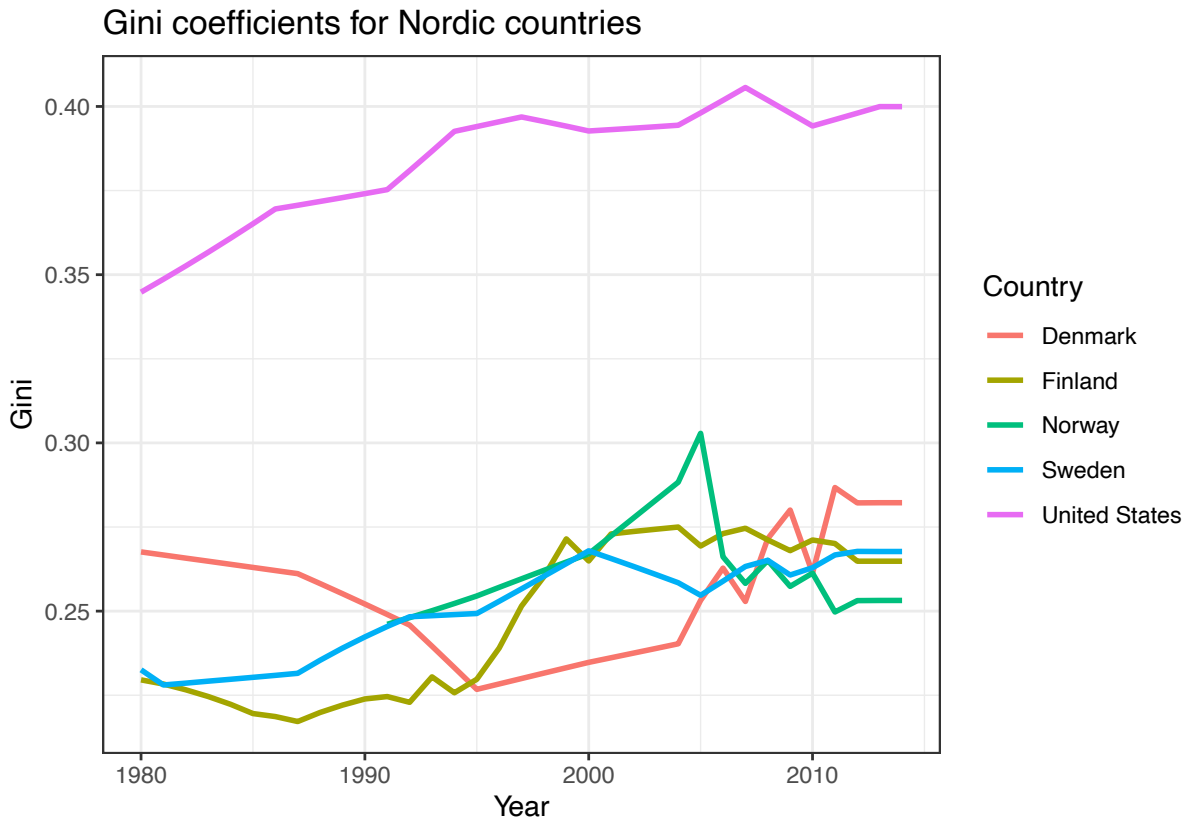
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(ineq)
# Set your working directory to the correct folder.
# Insert your file path for 'YOURFILEPATH'.
#setwd("YOURFILEPATH")


decile_data <- read_excel("GCIPrawdatatest.xlsx", skip = 2)
#The data is now in a 'tibble' (like a spreadsheet for R). Let's use the head function to look at the f
head(decile_data)
```

```
## # A tibble: 6 x 14
##   Country      Year `Decile 1 Income` `Decile 2 Income` `Decile 3 Income`
##   <chr>       <dbl>             <dbl>             <dbl>             <dbl>
## 1 Afghanistan  1980               206               350               455
## 2 Afghanistan  1981               212               361               469
## 3 Afghanistan  1982               221               377               490
## 4 Afghanistan  1983               238               405               527
## 5 Afghanistan  1984               249               424               551
## 6 Afghanistan  1985               256               435               566
## # ... with 9 more variables: `Decile 4 Income` <dbl>, `Decile 5 Income` <dbl>,
## #   `Decile 6 Income` <dbl>, `Decile 7 Income` <dbl>, `Decile 8 Income` <dbl>,
## #   `Decile 9 Income` <dbl>, `Decile 10 Income` <dbl>, `Mean Income` <dbl>,
## #   Population <dbl>
```

```r
#Now we use loops to complete our task. We begin by creating a new variable in our dataset, gini, which
decile_data$gini <- 0
#Now we use a loop to run through all the rows in our dataset (country-year combinations). For each row
#The function that calculates Gini coefficients from a vector of numbers is called Gini,and we apply it
# Give us the number of rows in decile_data
noc <- nrow(decile_data)

for (i in seq(1, noc)){
  # Go to Row I to get the decile data
  decs_i <- unlist(decile_data[i, 3:12])
  decile_data$gini[i] <- Gini(decs_i)
}
#With this code, we calculated 4,799 Gini coefficients without having to manually run the same command
#First we use the subset function to select Nordic countries and save their data as temp_data. As an ex
temp_data <- subset(
  decile_data, Country %in% c("United States","Sweden","Finland","Norway",
                              "Denmark"))
#Now we plot the data using ggplot.

ggplot(temp_data,
       aes(x = Year, y = gini, color = Country)) +
  geom_line(size = 1) +
  theme_bw() +
  geom_abline(linetype = "dotted", color = "grey50", size = 1) +
  ylab("Gini") +
  ggtitle("Gini coefficients for Nordic countries")
```

```
## Warning: Removed 11 row(s) containing missing values (geom_path).
```




```
#This example is based on great webpages of CORE: https://www.core-econ.org/doing-econo


#Oppgave 6
library(ggplot2)
library(gglorenz)
library(tidyverse)
library(PxWebApiData)
library(readxl)
library(ineq)
library(ggpubr)
#Hvilke variabler som finnes i tabellen
variables <- ApiData("https://data.ssb.no/api/v0/en/table/12558/",
                     returnMetaFrames = TRUE)
names(variables)
```

```
## [1] "Region"       "InntektSkatt" "Desiler"      "ContentsCode" "Tid"
```

```
#hvilke verdier har ulike variablene
values <- ApiData("https://data.ssb.no/api/v0/en/table/12558/",
                  returnMetaData = TRUE)
#Kommunekoder
values[[1]]$values
```

```
##   [1] "0"     "30"    "01"    "3001"  "3002"  "3003"  "3004"  "3005"  "3006"
```

```
##  [10] "3007"  "3011"  "3012"  "3013"  "3014"  "3015"  "3016"  "3017"  "3018"
##  [19] "3019"  "3020"  "3021"  "3022"  "3023"  "3024"  "3025"  "3026"  "3027"
##  [28] "3028"  "3029"  "3030"  "3031"  "3032"  "3033"  "3034"  "3035"  "3036"
##  [37] "3037"  "3038"  "3039"  "3040"  "3041"  "3042"  "3043"  "3044"  "3045"
##  [46] "3046"  "3047"  "3048"  "3049"  "3050"  "3051"  "3052"  "3053"  "3054"
##  [55] "0101"  "0102"  "0103"  "0104"  "0105"  "0106"  "0111"  "0113"  "0114"
##  [64] "0115"  "0116"  "0117"  "0118"  "0119"  "0121"  "0122"  "0123"  "0124"
##  [73] "0125"  "0127"  "0128"  "0130"  "0131"  "0133"  "0134"  "0135"  "0136"
##  [82] "0137"  "0138"  "0199"  "02"    "0211"  "0213"  "0214"  "0215"  "0216"
##  [91] "0217"  "0219"  "0220"  "0221"  "0226"  "0227"  "0228"  "0229"  "0230"
## [100] "0231"  "0233"  "0234"  "0235"  "0236"  "0237"  "0238"  "0239"  "0299"
## [109] "03"    "0301"  "0399"  "34"    "04"    "3401"  "3403"  "3405"  "3407"
## [118] "3411"  "3412"  "3413"  "3414"  "3415"  "3416"  "3417"  "3418"  "3419"
## [127] "3420"  "3421"  "3422"  "3423"  "3424"  "3425"  "3426"  "3427"  "3428"
## [136] "3429"  "3430"  "3431"  "3432"  "3433"  "3434"  "3435"  "3436"  "3437"
## [145] "3438"  "3439"  "3440"  "3441"  "3442"  "3443"  "3446"  "3447"  "3448"
## [154] "3449"  "3450"  "3451"  "3452"  "3453"  "3454"  "0401"  "0402"  "0403"
## [163] "0412"  "0414"  "0415"  "0417"  "0418"  "0419"  "0420"  "0423"  "0425"
## [172] "0426"  "0427"  "0428"  "0429"  "0430"  "0432"  "0434"  "0435"  "0436"
## [181] "0437"  "0438"  "0439"  "0441"  "0499"  "05"    "0501"  "0502"  "0511"
## [190] "0512"  "0513"  "0514"  "0515"  "0516"  "0517"  "0518"  "0519"  "0520"
## [199] "0521"  "0522"  "0528"  "0529"  "0532"  "0533"  "0534"  "0536"  "0538"
## [208] "0540"  "0541"  "0542"  "0543"  "0544"  "0545"  "0599"  "06"    "3801"
## [217] "3802"  "3803"  "3804"  "3805"  "3806"  "3807"  "3808"  "3811"  "3812"
## [226] "3813"  "3814"  "3815"  "3816"  "3817"  "3818"  "3819"  "3820"  "3821"
## [235] "3822"  "3823"  "3824"  "3825"  "0601"  "0602"  "0604"  "0605"  "0612"
## [244] "0615"  "0616"  "0617"  "0618"  "0619"  "0620"  "0621"  "0622"  "0623"
## [253] "0624"  "0625"  "0626"  "0627"  "0628"  "0631"  "0632"  "0633"  "0699"
## [262] "38"    "07"    "0701"  "0702"  "0703"  "0704"  "0705"  "0706"  "0707"
## [271] "0708"  "0709"  "0710"  "0711"  "0712"  "0713"  "0714"  "0715"  "0716"
## [280] "0716u" "0717"  "0718"  "0719"  "0720"  "0721"  "0722"  "0723"  "0724"
## [289] "0725"  "0726"  "0727"  "0728"  "0729"  "0799"  "08"    "0805"  "0806"
## [298] "0807"  "0811"  "0814"  "0815"  "0817"  "0819"  "0821"  "0822"  "0826"
## [307] "0827"  "0828"  "0829"  "0830"  "0831"  "0833"  "0834"  "0899"  "42"
## [316] "09"    "4201"  "4202"  "4203"  "4204"  "4205"  "4206"  "4207"  "4211"
## [325] "4212"  "4213"  "4214"  "4215"  "4216"  "4217"  "4218"  "4219"  "4220"
## [334] "4221"  "4222"  "4223"  "4224"  "4225"  "4226"  "4227"  "4228"  "0901"
## [343] "0903"  "0904"  "0906"  "0911"  "0912"  "0914"  "0918"  "0919"  "0920"
## [352] "0921"  "0922"  "0923"  "0924"  "0926"  "0928"  "0929"  "0932"  "0933"
## [361] "0935"  "0937"  "0938"  "0940"  "0941"  "0999"  "10"    "1001"  "1002"
## [370] "1003"  "1004"  "1014"  "1017"  "1018"  "1021"  "1026"  "1027"  "1029"
## [379] "1032"  "1034"  "1037"  "1046"  "1099"  "11"    "1101"  "1102"  "1103"
## [388] "1106"  "1108"  "1111"  "1112"  "1114"  "1119"  "1120"  "1121"  "1122"
## [397] "1124"  "1127"  "1129"  "1130"  "1133"  "1134"  "1135"  "1141"  "1142"
## [406] "1144"  "1145"  "1146"  "1149"  "1151"  "1154"  "1159"  "1160"  "1199"
## [415] "46"    "12"    "4601"  "4602"  "4611"  "4612"  "4613"  "4614"  "4615"
## [424] "4616"  "4617"  "4618"  "4619"  "4620"  "4621"  "4622"  "4623"  "4624"
## [433] "4625"  "4626"  "4627"  "4628"  "4629"  "4630"  "4631"  "4632"  "4633"
## [442] "4634"  "4635"  "4636"  "4637"  "4638"  "4639"  "4640"  "4641"  "4642"
## [451] "4643"  "4644"  "4645"  "4646"  "4647"  "4648"  "4649"  "4650"  "4651"
## [460] "1201"  "1211"  "1214"  "1216"  "1219"  "1221"  "1222"  "1223"  "1224"
## [469] "1227"  "1228"  "1230"  "1231"  "1232"  "1233"  "1234"  "1235"  "1238"
## [478] "1241"  "1242"  "1243"  "1244"  "1245"  "1246"  "1247"  "1248"  "1249"
## [487] "1250"  "1251"  "1252"  "1253"  "1255"  "1256"  "1259"  "1260"  "1263"
```

```
## [496] "1264"  "1265"  "1266"  "1299"  "13"    "1301"  "14"    "1401"  "1411"
## [505] "1412"  "1413"  "1416"  "1417"  "1418"  "1419"  "1420"  "1421"  "1422"
## [514] "1424"  "1426"  "1428"  "1429"  "1430"  "1431"  "1432"  "1433"  "1438"
## [523] "1439"  "1441"  "1443"  "1444"  "1445"  "1448"  "1449"  "1499"  "15"
## [532] "1501"  "1502"  "1503"  "1504"  "1505"  "1506"  "1507"  "1511"  "1514"
## [541] "1515"  "1516"  "1517"  "1519"  "1520"  "1523"  "1524"  "1525"  "1526"
## [550] "1527"  "1528"  "1529"  "1531"  "1532"  "1534"  "1535"  "1539"  "1543"
## [559] "1545"  "1546"  "1547"  "1548"  "1551"  "1554"  "1556"  "1557"  "1560"
## [568] "1563"  "1566"  "1567"  "1569"  "1571"  "1572"  "1573"  "1576"  "1577"
## [577] "1578"  "1579"  "1599"  "50"    "16"    "5001"  "5004"  "5005"  "5006"
## [586] "5007"  "5011"  "5012"  "5013"  "5014"  "5015"  "5016"  "5017"  "5018"
## [595] "5019"  "5020"  "5021"  "5022"  "5023"  "5024"  "5025"  "5026"  "5027"
## [604] "5028"  "5029"  "5030"  "5031"  "5032"  "5033"  "5034"  "5035"  "5036"
## [613] "5037"  "5038"  "5039"  "5040"  "5041"  "5042"  "5043"  "5044"  "5045"
## [622] "5046"  "5047"  "5048"  "5049"  "5050"  "5051"  "5052"  "5053"  "5054"
## [631] "5055"  "5056"  "5057"  "5058"  "5059"  "5060"  "5061"  "1601"  "1612"
## [640] "1613"  "1617"  "1620"  "1621"  "1622"  "1624"  "1627"  "1630"  "1632"
## [649] "1633"  "1634"  "1635"  "1636"  "1638"  "1640"  "1644"  "1645"  "1648"
## [658] "1653"  "1657"  "1662"  "1663"  "1664"  "1665"  "1699"  "17"    "1702"
## [667] "1703"  "1711"  "1714"  "1717"  "1718"  "1719"  "1721"  "1723"  "1724"
## [676] "1725"  "1729"  "1736"  "1738"  "1739"  "1740"  "1742"  "1743"  "1744"
## [685] "1748"  "1749"  "1750"  "1751"  "1755"  "1756"  "1799"  "18"    "1804"
## [694] "1805"  "1806"  "1811"  "1812"  "1813"  "1814"  "1815"  "1816"  "1818"
## [703] "1820"  "1822"  "1824"  "1825"  "1826"  "1827"  "1828"  "1832"  "1833"
## [712] "1834"  "1835"  "1836"  "1837"  "1838"  "1839"  "1840"  "1841"  "1842"
## [721] "1843"  "1845"  "1848"  "1849"  "1850"  "1851"  "1852"  "1853"  "1854"
## [730] "1855"  "1856"  "1857"  "1858"  "1859"  "1860"  "1865"  "1866"  "1867"
## [739] "1868"  "1870"  "1871"  "1874"  "1875"  "1899"  "54"    "19"    "5401"
## [748] "5402"  "5403"  "5404"  "5405"  "5406"  "5411"  "5412"  "5413"  "5414"
## [757] "5415"  "5416"  "5417"  "5418"  "5419"  "5420"  "5421"  "5422"  "5423"
## [766] "5424"  "5425"  "5426"  "5427"  "5428"  "5429"  "5430"  "5432"  "5433"
## [775] "5434"  "5435"  "5436"  "5437"  "5438"  "5439"  "5440"  "5441"  "5442"
## [784] "5443"  "5444"  "1901"  "1902"  "1903"  "1911"  "1913"  "1915"  "1917"
## [793] "1919"  "1920"  "1921"  "1922"  "1923"  "1924"  "1925"  "1926"  "1927"
## [802] "1928"  "1929"  "1931"  "1933"  "1936"  "1938"  "1939"  "1940"  "1941"
## [811] "1942"  "1943"  "1999"  "20"    "2001"  "2002"  "2003"  "2004"  "2011"
## [820] "2012"  "2014"  "2015"  "2016"  "2017"  "2018"  "2019"  "2020"  "2021"
## [829] "2022"  "2023"  "2024"  "2025"  "2027"  "2028"  "2030"  "2099"  "21"
## [838] "2111"  "2112"  "2115"  "2121"  "2131"  "2199"  "22"    "2211"  "2299"
## [847] "23"    "2300"  "2311"  "2321"  "2399"  "25"    "2599"  "26"    "88"
## [856] "99"    "9999"
```

```
#Inntekt før/etter skatt
values[[2]]$values # 00 = Samlet inntekt, 00S=Inntekt etter skatt
```

```
## [1] "00"  "00S"
```

```
#Desiler
values[[3]]$values
```

```
##  [1] "01" "02" "03" "04" "05" "06" "07" "08" "09" "10"
```

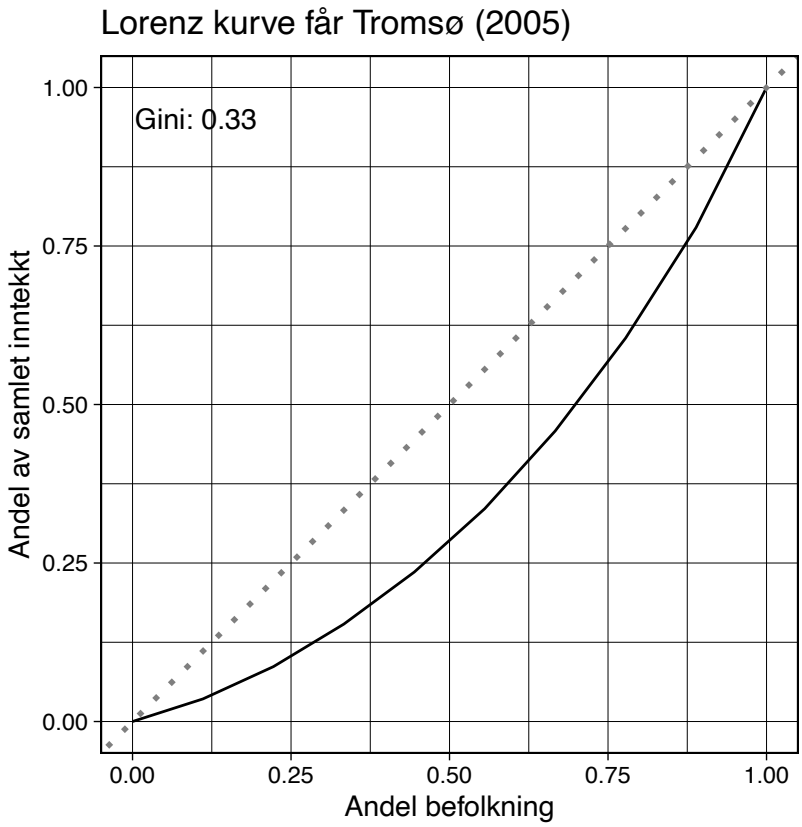
```
#Statistikkvariabel
values[[4]]$values
```

```
## [1] "AndelHush"  "VerdiDesil" "AntHush"
```

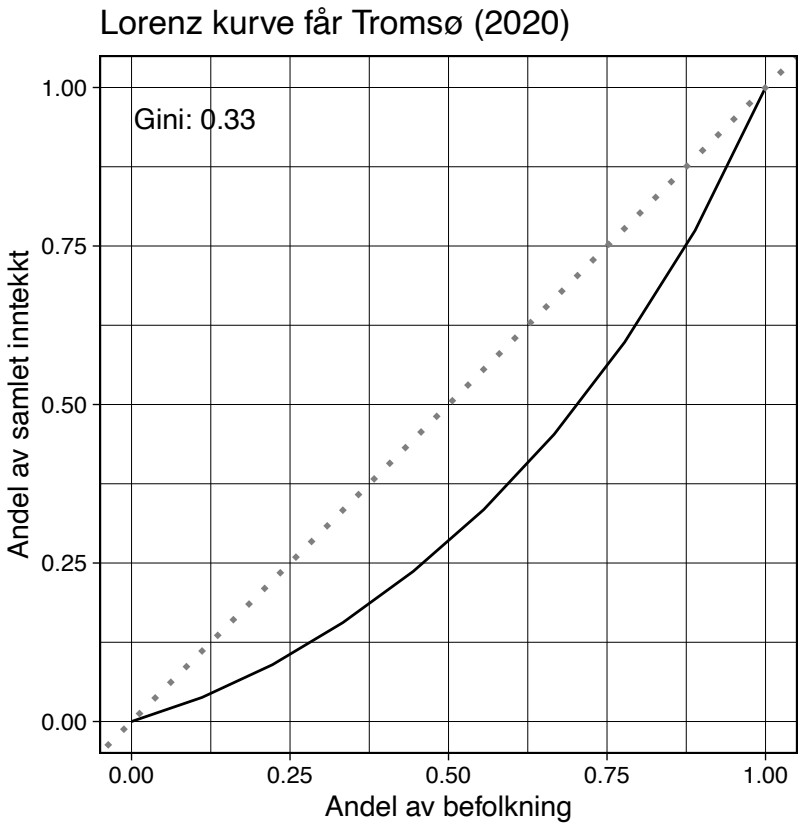
```
#År
values[[5]]$values
```

```
##  [1] "2005" "2006" "2007" "2008" "2009" "2010" "2011" "2012" "2013" "2014"
## [11] "2015" "2016" "2017" "2018" "2019" "2020"
```

```
data <- ApiData("https://data.ssb.no/api/v0/en/table/12558/",
                Tid =c("2005","2020"), # Velg årene 2005 og 2020
                Desiler=c("01", "02", "03" ,"04", "05", "06" ,"07", "08" ,"09", "10"), #Vi velger alle
                InntektSkatt="00", #Vi velger samlet inntekt
                ContentsCode="VerdiDesil", #Velger den høyeste verdien i desilen
                Region=c("5401","1902")) #Tromsø endret kommunenummer i 2020

# Henter frem tabelen og kaller den Lorenz1
Lorenz1<- data[[2]]



DF<- Lorenz1%>%
  filter(Tid=='2005')%>%
  filter(value!="NA")

DF %>%
  ggplot(aes(value), color = Country) +
  stat_lorenz(desc = FALSE) +
  coord_fixed() +
  geom_abline(linetype = "dotted", color = "grey50", size = 1) +
  theme(plot.title = element_text(face = "bold", size = 12),
        legend.background = element_rect(fill = "white", size = 4, colour = "white"),
        legend.justification = c(0, 1),
        legend.position = c(0, 1),
        axis.ticks = element_line(colour = "grey70", size = 0.2),
        panel.grid.major = element_line(colour = "grey70", size = 0.2),
        panel.grid.minor = element_blank()) +
  labs(x = "Andel befolkning",
       y = "Andel av samlet inntekkt",) +
  theme_linedraw() + ggtitle("Lorenz kurve får Tromsø (2005)") +
  scale_color_brewer(type = "seq", palette = "Spectral") +
  annotate_ineq(Lorenz1$value)
```




```
view(DF)

DF_2<- Lorenz1%>%
  filter(Tid=='2020')%>%
  filter(value!="NA")


DF_2 %>%
  ggplot(aes(value), color = Country) +
  stat_lorenz(desc = FALSE) +
  coord_fixed() +
  geom_abline(linetype = "dotted", color = "grey50", size = 1) +
  theme(plot.title = element_text(face = "bold", size = 12),
        legend.background = element_rect(fill = "white", size = 4, colour = "white"),
        legend.justification = c(0, 1),
        legend.position = c(0, 1),
        axis.ticks = element_line(colour = "grey70", size = 0.2),
        panel.grid.major = element_line(colour = "grey70", size = 0.2),
        panel.grid.minor = element_blank()) +
  labs(x = "Andel av befolkning",
       y = "Andel av samlet inntekkt",) +
  theme_linedraw() + ggtitle("Lorenz kurve får Tromsø (2020)") +
  scale_color_brewer(type = "seq", palette = "Spectral") +
  annotate_ineq(Lorenz1$value)
```


Lorenz kurve får Tromsø (2020)
Gini: 0.33
Andel av samlet inntekkt
Andel av befolkning
0.00
0.25
0.50
0.75
1.00

```
view(DF_2)
```