# Exploring gapminder

**Practice notebook for DSC 205 Spring 2022**

# 1 Data transformations

## 1.1 Value transformation (before plotting)

- Transformations can help provide more informative summaries and plots.
- `gapminder` contains a `gdp` column.

```
str(gapminder)
```

```
'data.frame':   10545 obs. of  9 variables:
 $ country         : Factor w/ 185 levels "Albania","Algeria",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ year            : int  1960 1960 1960 1960 1960 1960 1960 1960 1960 1960 ...
 $ infant_mortality: num  115.4 148.2 208 NA 59.9 ...
 $ life_expectancy : num  62.9 47.5 36 63 65.4 ...
 $ fertility       : num  6.19 7.65 7.32 4.43 3.11 4.55 4.82 3.45 2.7 5.57 ...
 $ population      : num  1636054 11124892 5270844 54681 20619075 ...
 $ gdp             : num  NA 1.38e+10 NA NA 1.08e+11 ...
 $ continent       : Factor w/ 5 levels "Africa","Americas",..: 4 1 1 2 2 3 2 5 4 3 ...
 $ region          : Factor w/ 22 levels "Australia and New Zealand",..: 19 11 10 2 15 21
```

- We add a column `dollars_per_day` by dividing the GDP by `population` (that gives us GDP per person) and then by 365. The `dplyr::mutate` function adds the new column to the data frame.

```
gm_dplyr <- gapminder %>%
  mutate(dollars_per_day = gdp/population/365)
str(gm_dplyr)
```

```
'data.frame':   10545 obs. of  10 variables:
 $ country         : Factor w/ 185 levels "Albania","Algeria",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ year            : int  1960 1960 1960 1960 1960 1960 1960 1960 1960 1960 ...
 $ infant_mortality: num  115.4 148.2 208 NA 59.9 ...
 $ life_expectancy : num  62.9 47.5 36 63 65.4 ...
 $ fertility       : num  6.19 7.65 7.32 4.43 3.11 4.55 4.82 3.45 2.7 5.57 ...
 $ population      : num  1636054 11124892 5270844 54681 20619075 ...
 $ gdp             : num  NA 1.38e+10 NA NA 1.08e+11 ...
 $ continent       : Factor w/ 5 levels "Africa","Americas",..: 4 1 1 2 2 3 2 5 4 3 ...
 $ region          : Factor w/ 22 levels "Australia and New Zealand",..: 19 11 10 2 15 21
 $ dollars_per_day : num  NA 3.41 NA NA 14.39 ...
```

- In `Base-R`, it works like this:

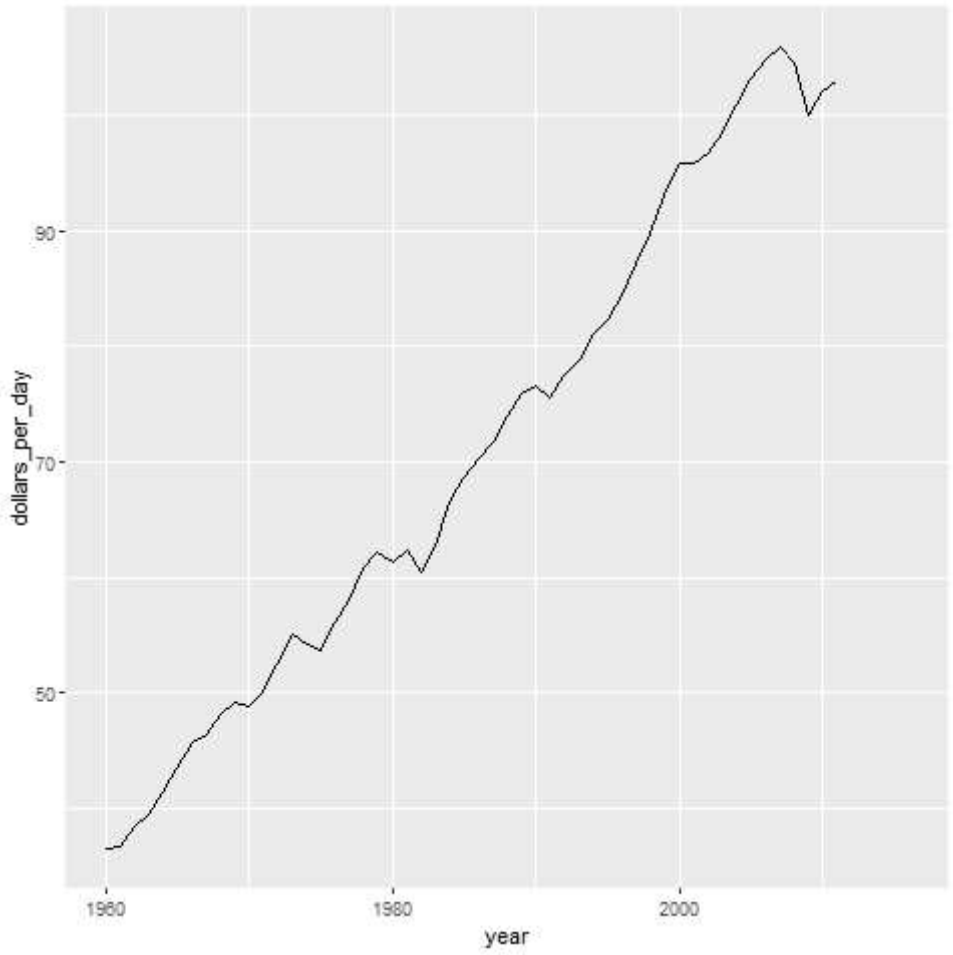
```
dollars_per_day <- gapminder$gdp / gapminder$population / 365
gm <- cbind(gapminder,dollars_per_day)
str(gm)
```

```
'data.frame':   10545 obs. of  10 variables:
 $ country         : Factor w/ 185 levels "Albania","Algeria",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ year            : int  1960 1960 1960 1960 1960 1960 1960 1960 1960 1960 ...
 $ infant_mortality: num  115.4 148.2 208 NA 59.9 ...
 $ life_expectancy : num  62.9 47.5 36 63 65.4 ...
 $ fertility       : num  6.19 7.65 7.32 4.43 3.11 4.55 4.82 3.45 2.7 5.57 ...
 $ population      : num  1636054 11124892 5270844 54681 20619075 ...
 $ gdp             : num  NA 1.38e+10 NA NA 1.08e+11 ...
 $ continent       : Factor w/ 5 levels "Africa","Americas",..: 4 1 1 2 2 3 2 5 4 3 ...
 $ region          : Factor w/ 22 levels "Australia and New Zealand",..: 19 11 10 2 15 21
 $ dollars_per_day : num  NA 3.41 NA NA 14.39 ...
```

- [ ]

  Plot `dollars_per_day` for the data set in 1960 and 2012 using `ggplot2`.

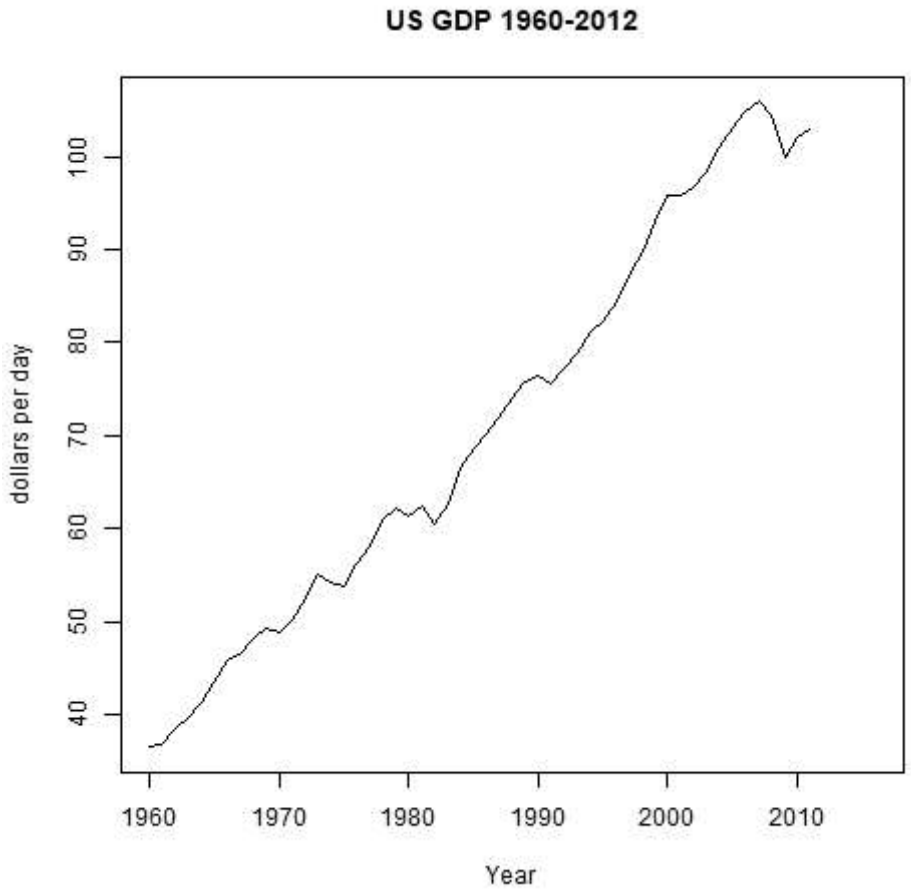
```
gm_dplyr %>%
  filter(country == "United States") %>%
  ggplot( aes ( x = year, y = dollars_per_day ) ) +
  geom_line()
```




- [ ]

  Plot `dollars_per_day` for the data set in 1960 and 2012 using the `Base_R` function `plot`.

```
plot(y = gm$dollars_per_day[gm$country=="United States"],
     x = gm$year[gm$country=="United States"],
     type="l", xlab="Year", ylab="dollars per day", main="US GDP 1960-2012",
     na.rm=TRUE)
```

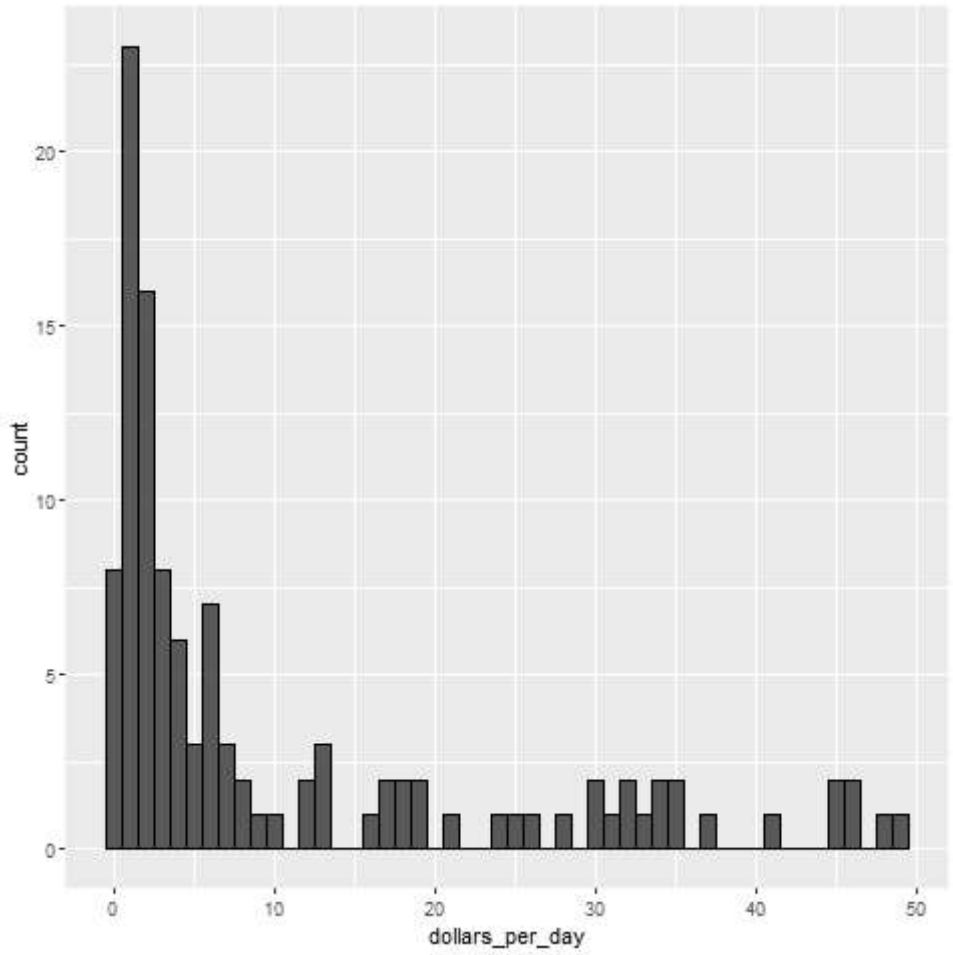



## 1.2 Scale transformations (scale axes)

- [ ]

  Make a histogram using `ggplot2` for the `year` 1970. For the histogram, use the arguments `binwidth = 1` and `color = "black"`.

```
gm_dplyr %>%
  filter( year == 1970 & !is.na(gdp) ) %>%
  ggplot( aes (dollars_per_day ) ) +
  geom_histogram(binwidth = 1, color = "black")
```




- [ ]

  Do it with `hist` in `Base-R`.

```
```

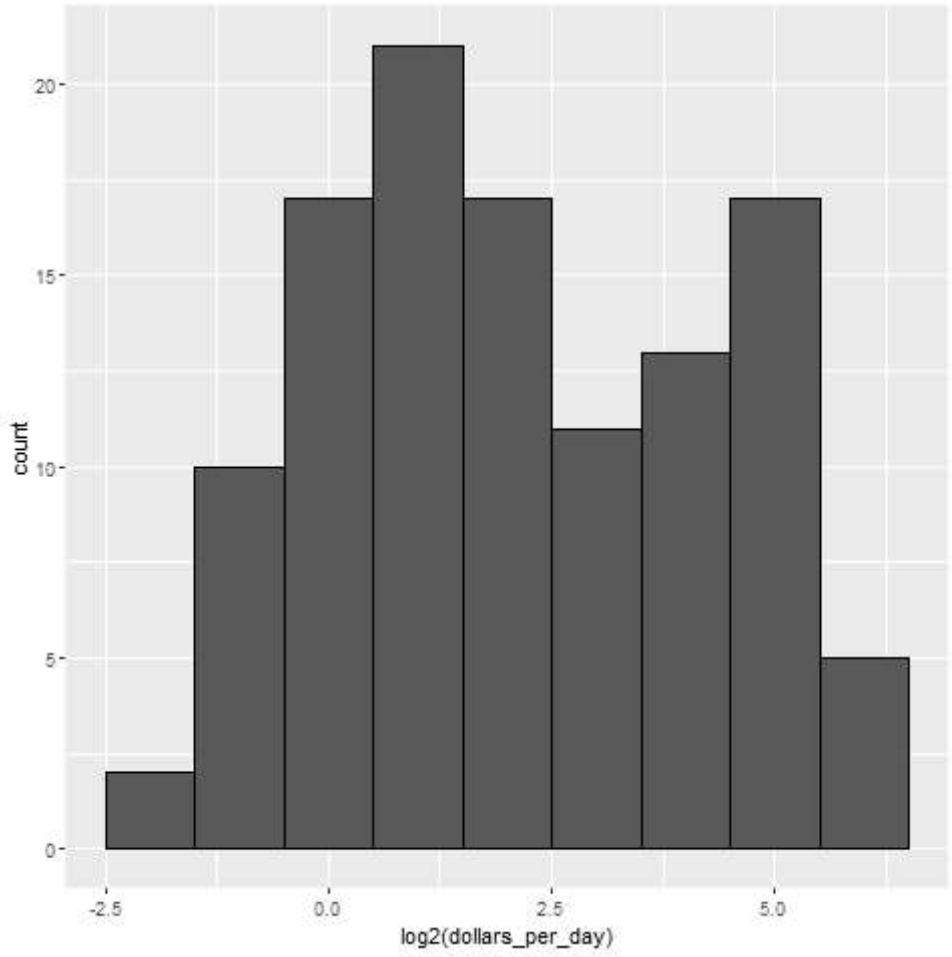
- It might be more informative to apply a logarithm (base 2) transform to see how many countries have average daily incomes that are multiples of 2:

| INCOME | POVERY |
|---|---|
| $1 | extremely poor |
| $2 | very poor |
| $4 | poor |
| $8 | middle |
| $16 | well off |
| $32 | rich |
| $64 | very rich |

- [ ]

Change the variable in the previous code block from `dollars_per_day` to `log2(dollars_per_day)`.

```
gm_dplyr %>%
  filter( year == 1970 & !is.na(gdp) ) %>%
  ggplot( aes (log2(dollars_per_day) ) ) +
  geom_histogram(binwidth = 1, color = "black")
```




## 1.3 Which base should you use?

- Common choices are `log2`, `log10`, and the natural `log` (base e).
- For data exploration, do not use the natural log (hard to imagine)
- Example: `population` sizes.
- `[ ]`

  What is the range of `population` sizes in `gapminder`?

  Do it in `dplyr` and then in `Base-R`.

  dplyr:

```
filter(gapminder, year == 1970) %>%
  summarize(
    min = min(population),
    max = max(population))
```

```
    min       max
1 46075 808510713
```

Base-R:

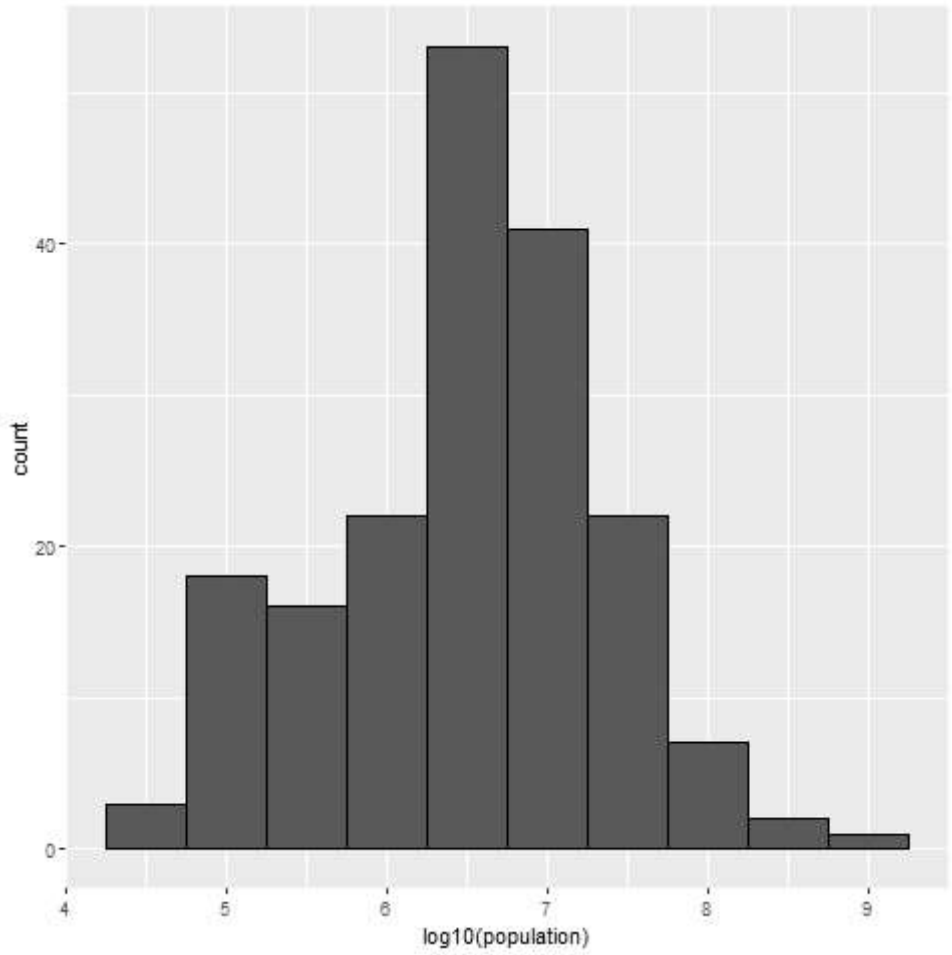
```
pop <- gapminder$population
yr  <- gapminder$year

summary(pop[yr==1970])
```

```
 Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
46075    826447   3875719  19490870  10232758 808510713
```

- [ ]

Draw a histogram of the transformed values of `population` using the argument `x = log10(population)`.

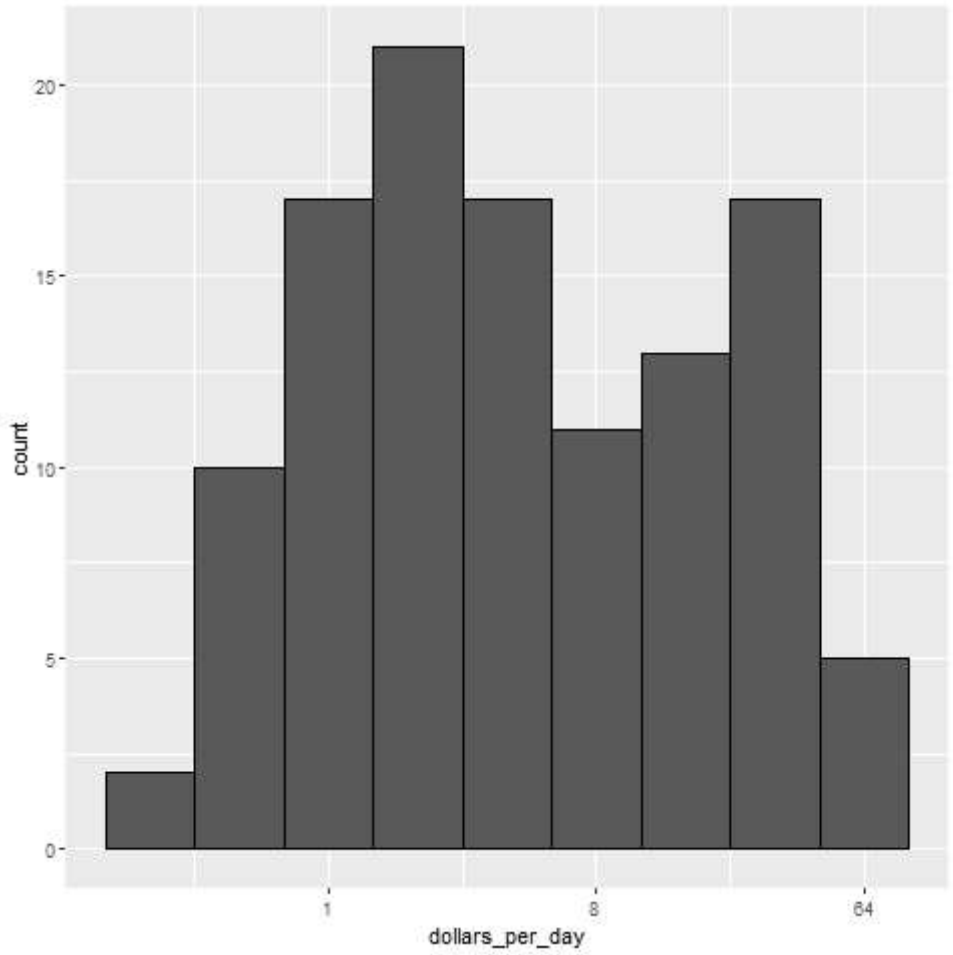
```
gapminder %>%
  filter( year == 1970 ) %>%
  ggplot( aes(log10(population) )) +
  geom_histogram(binwidth = 0.5, color = "black")
```




- [ ]

  To transform the axis with logs, you can use `scale_x_continuous` in `ggplot2`:

```
gm_dplyr %>%
  filter( year == 1970 & !is.na(gdp)) %>%
  ggplot( aes(dollars_per_day) ) +
  geom_histogram(binwidth = 1, color = "black") +
  scale_x_continuous(trans = "log2")
```




Author: Marcus Birkenkrahe
Created: 2022-04-11 Mon 14:43