Lab 3- Data Wrangling - Questions

Suyog Chandramouli

2025-09-24

Lab 3 - Gapminder Data Wrangling Lab Assignment

Using the Gapminder dataset, complete the following tasks. Use tidyverse and dplyr functions and pipes where appropriate. Remember to load the necessary libraries and the Gapminder dataset before starting.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
#install.packages('gapminder')
library(gapminder)
#tinytex::install_tinytex()
```

1) Find all countries in Asia with a life expectancy greater than 75 years in 2007.

```
gapminder
```

```
## # A tibble: 1,704 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.
##  2 Afghanistan Asia       1957    30.3  9240934      821.
##  3 Afghanistan Asia       1962    32.0 10267083      853.
##  4 Afghanistan Asia       1967    34.0 11537966      836.
##  5 Afghanistan Asia       1972    36.1 13079460      740.
##  6 Afghanistan Asia       1977    38.4 14880372      786.
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # i 1,694 more rows
```

```
gapminder %>%
  filter(year == '2007', lifeExp > 75, continent == 'Asia')
```

```
## # A tibble: 9 x 6
##   country          continent  year lifeExp       pop gdpPercap
##   <fct>            <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Bahrain          Asia       2007    75.6    708573    29796.
## 2 Hong Kong, China Asia       2007    82.2   6980412    39725.
## 3 Israel           Asia       2007    80.7   6426679    25523.
## 4 Japan            Asia       2007    82.6 127467972    31656.
## 5 Korea, Rep.      Asia       2007    78.6  49044790    23348.
## 6 Kuwait           Asia       2007    77.6   2505559    47307.
## 7 Oman             Asia       2007    75.6   3204897    22316.
## 8 Singapore        Asia       2007    80.0   4553009    47143.
## 9 Taiwan           Asia       2007    78.4  23174294    28718.
```

2) List the top 5 countries with the highest GDP per capita in 2007, in descending order.

[Hint: head() is a verb that takes in a parameter of number of rows to show]

```
gapminder %>%
  filter(year == '2007') %>%
  arrange(desc(gdpPercap)) %>%
  head(5)
```

```
## # A tibble: 5 x 6
##   country       continent  year lifeExp       pop gdpPercap
##   <fct>         <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Norway        Europe     2007    80.2   4627926    49357.
## 2 Kuwait        Asia       2007    77.6   2505559    47307.
## 3 Singapore     Asia       2007    80.0   4553009    47143.
## 4 United States Americas   2007    78.2 301139947    42952.
## 5 Ireland       Europe     2007    78.9   4109086    40676.
```

3) Create a new dataframe with only the country, continent, and GDP per capita for the year 2007.

```
gap <- gapminder %>%
  filter(year == '2007') %>%
  select(country, continent, gdpPercap)

gap
```

```
## # A tibble: 142 x 3
##    country     continent gdpPercap
##    <fct>       <fct>         <dbl>
##  1 Afghanistan Asia           975.
##  2 Albania     Europe        5937.
##  3 Algeria     Africa        6223.
##  4 Angola      Africa        4797.
##  5 Argentina   Americas     12779.
##  6 Australia   Oceania      34435.
```

```
##  7 Austria     Europe       36126.
##  8 Bahrain     Asia         29796.
##  9 Bangladesh  Asia          1391.
## 10 Belgium     Europe       33693.
## # i 132 more rows
```

4) Add a new column called "gdp" that calculates the total GDP (GDP per capita * population) for each country in 2007.

```
gapminder %>%
  filter(year == '2007') %>%
  mutate(gdp = pop * gdpPercap)
```

```
## # A tibble: 142 x 7
##    country     continent  year lifeExp       pop gdpPercap           gdp
##    <fct>       <fct>     <int>   <dbl>     <int>     <dbl>         <dbl>
##  1 Afghanistan Asia       2007    43.8  31889923      975.  31079291949.
##  2 Albania     Europe     2007    76.4   3600523     5937.  21376411360.
##  3 Algeria     Africa     2007    72.3  33333216     6223. 207444851958.
##  4 Angola      Africa     2007    42.7  12420476     4797.  59583895818.
##  5 Argentina   Americas   2007    75.3  40301927    12779. 515033625357.
##  6 Australia   Oceania    2007    81.2  20434176    34435. 703658358894.
##  7 Austria     Europe     2007    79.8   8199783    36126. 296229400691.
##  8 Bahrain     Asia       2007    75.6    708573    29796.  21112675360.
##  9 Bangladesh  Asia       2007    64.1 150448339     1391. 209311822134.
## 10 Belgium     Europe     2007    79.4  10392226    33693. 350141166520.
## # i 132 more rows
```

5) Create two separate dataframes: one for countries in Europe and one for countries in Africa, both for the year 2007. Then bind these dataframes together.

```
eur_2007 <- gapminder %>% filter(year == '2007', continent == 'Europe')
afr_2007 <- gapminder %>% filter(year == '2007', continent == 'Africa')

afr_eur_2007 <- bind_rows(eur_2007, afr_2007)
afr_eur_2007
```

```
## # A tibble: 82 x 6
##    country                continent  year lifeExp      pop gdpPercap
##    <fct>                  <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Albania                Europe     2007    76.4  3600523     5937.
##  2 Austria                Europe     2007    79.8  8199783    36126.
##  3 Belgium                Europe     2007    79.4 10392226    33693.
##  4 Bosnia and Herzegovina Europe     2007    74.9  4552198     7446.
##  5 Bulgaria               Europe     2007    73.0  7322858    10681.
##  6 Croatia                Europe     2007    75.7  4493312    14619.
##  7 Czech Republic         Europe     2007    76.5 10228744    22833.
##  8 Denmark                Europe     2007    78.3  5468120    35278.
##  9 Finland                Europe     2007    79.3  5238460    33207.
## 10 France                 Europe     2007    80.7 61083916    30470.
## # i 72 more rows
```

6) Calculate the average life expectancy and total population for each continent in 2007.

```
gapminder %>%
  filter(year == '2007') %>%
  group_by(continent) %>%
  summarize(
    avg_lifeExp = mean(lifeExp),
    total_pop = sum(pop)
  )
```

```
## # A tibble: 5 x 3
##   continent avg_lifeExp  total_pop
##   <fct>           <dbl>      <dbl>
## 1 Africa           54.8  929539692
## 2 Americas         73.6  898871184
## 3 Asia             70.7 3811953827
## 4 Europe           77.6  586098529
## 5 Oceania          80.7   24549947
```

7) Create a wide format dataframe that shows as columns the population for each country for each year E.g. the column names would be year, population_India, population_Canada, ... and so on.

[Hint: you will need to select only the relevant variables for generating this final output; names_]

```
gapminder_wide <- gapminder %>%
  select(country, year, pop) %>%
  pivot_wider(names_from = country,
              values_from = pop)

gapminder_wide
```

```
## # A tibble: 12 x 143
##     year Afghanistan Albania  Algeria Angola Argentina Australia Austria Bahrain
##    <int>       <int>   <int>    <int>  <int>     <int>     <int>   <int>   <int>
##  1  1952     8425333 1282697  9279525 4.23e6  17876956   8691212 6927772  120447
##  2  1957     9240934 1476505 10270856 4.56e6  19610538   9712569 6965860  138655
##  3  1962    10267083 1728137 11000948 4.83e6  21283783  10794968 7129864  171863
##  4  1967    11537966 1984060 12760499 5.25e6  22934225  11872264 7376998  202182
##  5  1972    13079460 2263554 14760787 5.89e6  24779799  13177000 7544201  230800
##  6  1977    14880372 2509048 17152804 6.16e6  26983828  14074100 7568430  297410
##  7  1982    12881816 2780097 20033753 7.02e6  29341374  15184200 7574613  377967
##  8  1987    13867957 3075321 23254956 7.87e6  31620918  16257249 7578903  454612
##  9  1992    16317921 3326498 26298373 8.74e6  33958947  17481977 7914969  529491
## 10  1997    22227415 3428038 29072015 9.88e6  36203463  18565243 8069876  598561
## 11  2002    25268405 3508512 31287142 1.09e7  38331121  19546792 8148312  656397
## 12  2007    31889923 3600523 33333216 1.24e7  40301927  20434176 8199783  708573
## # i 134 more variables: Bangladesh <int>, Belgium <int>, Benin <int>,
## #   Bolivia <int>, 'Bosnia and Herzegovina' <int>, Botswana <int>,
## #   Brazil <int>, Bulgaria <int>, 'Burkina Faso' <int>, Burundi <int>,
## #   Cambodia <int>, Cameroon <int>, Canada <int>,
## #   'Central African Republic' <int>, Chad <int>, Chile <int>, China <int>,
## #   Colombia <int>, Comoros <int>, 'Congo, Dem. Rep.' <int>,
## #   'Congo, Rep.' <int>, 'Costa Rica' <int>, 'Cote d'Ivoire' <int>, ...
```

```
# I guess they just put the original value as column names when there's only one variable to be shown??
# e.g., only the population data here so it doesn't need to be 'pop_Albania'
```