Data exploration and descriptives

Zoltan Kekecs

November 04, 2020

Data exploration

This exercise will teach specific guidelines for exploring the distribution of data and the relationship of different types of variables.

Load packages

We will need the following packages in this exercise:

```
if (!require("gridExtra")) install.packages("gridExtra")
library(gridExtra) # for grid.arrange
if (!require("psych")) install.packages("psych")
library(psych) # for describe
if (!require("tidyverse")) install.packages("tidyverse")
library(tidyverse) # for dplyr and ggplot2
```

Load data

We will work with the latest COVID dataset published by the WHO as of today. We use the **read_csv()** to read the data. That function is the tidyverse counterpart of the previously encountered read.csv (which is a base R function). They work the same way, both read files with a .csv format. The only difference is the read_csv() creates a **tibble** format instead of the data.frame format. tibbles are data frames but they are slightly more processed, containing a bit more structural information about the dataset than the regular data.frame

```
COVID_data <- read_csv("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-co
```

Data overview

We should always start by getting an overall look at the dataset (we can print the data object, look at the dataset with View, or get some more structural information using the functions learned in the previous exercises.)

Printing a **tibble object** shows more information than printing a simple data.frame. It basically shows the same information as if str() and head() was also run.

```
COVID_data
```

```
## # A tibble: 54,607 x 49
##    iso_code continent location date       total_cases new_cases new_cases_smoot~
##    <chr>    <chr>     <chr>    <date>           <dbl>     <dbl>            <dbl>
##  1 AFG      Asia      Afghani~ 2019-12-31          NA         0               NA
##  2 AFG      Asia      Afghani~ 2020-01-01          NA         0               NA
##  3 AFG      Asia      Afghani~ 2020-01-02          NA         0               NA
##  4 AFG      Asia      Afghani~ 2020-01-03          NA         0               NA
##  5 AFG      Asia      Afghani~ 2020-01-04          NA         0               NA
##  6 AFG      Asia      Afghani~ 2020-01-05          NA         0               NA
##  7 AFG      Asia      Afghani~ 2020-01-06          NA         0                0
##  8 AFG      Asia      Afghani~ 2020-01-07          NA         0                0
##  9 AFG      Asia      Afghani~ 2020-01-08          NA         0                 0
## 10 AFG      Asia      Afghani~ 2020-01-09          NA         0                 0
## # ... with 54,597 more rows, and 42 more variables: total_deaths <dbl>,
## #   new_deaths <dbl>, new_deaths_smoothed <dbl>, total_cases_per_million <dbl>,
## #   new_cases_per_million <dbl>, new_cases_smoothed_per_million <dbl>,
## #   total_deaths_per_million <dbl>, new_deaths_per_million <dbl>,
## #   new_deaths_smoothed_per_million <dbl>, icu_patients <lgl>,
## #   icu_patients_per_million <lgl>, hosp_patients <lgl>,
## #   hosp_patients_per_million <lgl>, weekly_icu_admissions <lgl>,
## #   weekly_icu_admissions_per_million <lgl>, weekly_hosp_admissions <lgl>,
## #   weekly_hosp_admissions_per_million <lgl>, total_tests <lgl>,
## #   new_tests <lgl>, total_tests_per_thousand <lgl>,
## #   new_tests_per_thousand <lgl>, new_tests_smoothed <lgl>,
## #   new_tests_smoothed_per_thousand <lgl>, tests_per_case <lgl>,
## #   positive_rate <lgl>, tests_units <lgl>, stringency_index <dbl>,
## #   population <dbl>, population_density <dbl>, median_age <dbl>,
## #   aged_65_older <dbl>, aged_70_older <dbl>, gdp_per_capita <dbl>,
## #   extreme_poverty <dbl>, cardiovasc_death_rate <dbl>,
## #   diabetes_prevalence <dbl>, female_smokers <dbl>, male_smokers <dbl>,
## #   handwashing_facilities <dbl>, hospital_beds_per_thousand <dbl>,
## #   life_expectancy <dbl>, human_development_index <dbl>
```

```
View(COVID_data)
```

Descriptive statistics

There are a number of functions dedicated to get descriptive statistics from datasets. One of the most commonly used ones is the **summary()** function, giving us information about the minimum and maximum values, the mean, the median, the kvartiles and missing data all at once.

For example we can get descriptives from the variable "total cases" by using the code below: (note that we used the **select()** function to subset the dataset to only show descriptives for this one variable)

```
COVID_data %>%
  select(total_cases) %>%
  summary()
```

```
##   total_cases
##  Min.   :       1
##  1st Qu.:     157
##  Median :    1900
##  Mean   :  150602
##  3rd Qu.:   19343
##  Max.   :47594234
##  NA's   :3634
```

Or we can get the same data for all of the variables, if we run the summary function on the whole dataset.

```
COVID_data %>%
  summary()
```

One thing you might notice during this initial exploration is that there are some extremely large numbers. If we look at the dataset, it is because the last rows contain aggregated "World" and "International" data. Let's exclude these rows from our dataset. **Note that we used "!" to indicate "NOT"**.

```
COVID_data <- COVID_data %>%
  filter(location != "World", location != "International")
```

______________*Practice*_________________

- How many registered cases were there in total in Sweden on 2020.Nov.04 (*total_cases*)?
- What was the highest value of registered new daily cases in Sweden during this pandemic (*new_cases*)?

More descriptives!

We can use the **describe()** function within the **Psych** package to get even more descriptive statistics such as skew and kurtosis which can be used to check the assumptions of some statistical tests about normality of distribution.

One limitation of the describe function is that it primarily works for numerical, continuous data, and it returns warning messages when encountering categorical/nominal variables. So we exclude these categorical variables using the select() function, and only run the function below on the numerical data.

```
COVID_data %>%
  select(-date, -iso_code, -continent, -location, -contains("tests"), -positive_rate) %>%
  describe()
```

```
##                                     vars     n        mean           sd
## total_cases                            1 50389    76147.45    458646.69
## new_cases                              2 53311      892.67      4998.46
## new_cases_smoothed                     3 52516      878.18      4859.23
## total_deaths                           4 41305     3280.96     14921.32
## new_deaths                             5 53311       22.80       124.15
## new_deaths_smoothed                    6 52516       22.76       114.38
## total_cases_per_million                7 50389     2792.58      5493.60
## new_cases_per_million                  8 53311       34.50       122.66
## new_cases_smoothed_per_million         9 52516       33.43        87.70
## total_deaths_per_million              10 41305       90.08       174.97
## new_deaths_per_million                11 53311        0.62         2.96
## new_deaths_smoothed_per_million       12 52516        0.62         1.89
## icu_patients                          13     0         NaN           NA
## icu_patients_per_million              14     0         NaN           NA
## hosp_patients                         15     0         NaN           NA
## hosp_patients_per_million             16     0         NaN           NA
## weekly_icu_admissions                 17     0         NaN           NA
## weekly_icu_admissions_per_million     18     0         NaN           NA
## weekly_hosp_admissions                19     0         NaN           NA
## weekly_hosp_admissions_per_million    20     0         NaN           NA
## stringency_index                      21 45313       56.89        26.31
## population                            22 53987 41986805.10 155334360.94
## population_density                    23 51443      363.20      1649.81
## median_age                            24 48288       31.23         9.07
## aged_65_older                         25 47558        9.20         6.32
## aged_70_older                         26 48035        5.82         4.31
## gdp_per_capita                        27 47647    20731.45     20404.02
## extreme_poverty                       28 31658       12.38        19.47
## cardiovasc_death_rate                 29 48261      252.40       117.80
## diabetes_prevalence                   30 50031        8.06         4.18
## female_smokers                        31 37528       10.80        10.51
## male_smokers                          32 37041       32.63        13.50
## handwashing_facilities                33 22637       52.07        31.84
## hospital_beds_per_thousand            34 43464        3.10         2.53
## life_expectancy                      35 53291       73.97         7.41
## human_development_index              36 46889        0.72         0.15
##                                        median     trimmed          mad      min
## total_cases                           1903.00    11832.74      2802.11     1.00
## new_cases                               13.00      120.90        19.27 -8261.00
## new_cases_smoothed                      17.86      125.02        26.47  -552.00
## total_deaths                            79.00      458.83       114.16     1.00
## new_deaths                               0.00        2.06         0.00 -1918.00
## new_deaths_smoothed                      0.29        2.24         0.42  -232.14
## total_cases_per_million                532.73     1464.90       778.93     0.00
## new_cases_per_million                    2.00       12.43         2.97 -2212.55
## new_cases_smoothed_per_million           3.64       14.15         5.39  -269.98
## total_deaths_per_million                19.00       45.62        26.71     0.00
## new_deaths_per_million                   0.00        0.16         0.00   -67.90
## new_deaths_smoothed_per_million          0.03        0.21         0.04    -9.68
## icu_patients                               NA         NaN           NA      Inf
## icu_patients_per_million                   NA         NaN           NA      Inf
## hosp_patients                              NA         NaN           NA      Inf
## hosp_patients_per_million                  NA         NaN           NA      Inf
## weekly_icu_admissions                      NA         NaN           NA      Inf
## weekly_icu_admissions_per_million          NA         NaN           NA      Inf
## weekly_hosp_admissions                     NA         NaN           NA      Inf
## weekly_hosp_admissions_per_million         NA         NaN           NA      Inf
## stringency_index                        61.11       58.90        27.46     0.00
## population                         7976985.00 14887158.18 11554429.61   809.00
## population_density                      90.67      125.18        99.01     0.14
## median_age                              31.10       31.24        12.16    15.10
## aged_65_older                            6.93        8.62         5.87     1.14
## aged_70_older                            4.32        5.33         3.82     0.53
## gdp_per_capita                       13913.84    17521.62     15805.32   661.24
## extreme_poverty                          2.00        7.93         2.67     0.10
## cardiovasc_death_rate                  240.21      241.37       122.28    79.37
## diabetes_prevalence                      7.11        7.64         3.68     0.99
## female_smokers                           6.30        9.46         7.86     0.10
## male_smokers                            31.40       31.97        14.53     7.70
## handwashing_facilities                  50.54       52.53        46.35     1.19
## hospital_beds_per_thousand               2.50        2.71         1.93     0.10
## life_expectancy                         75.40       74.65         7.12    53.28
## human_development_index                  0.75        0.73         0.16     0.35
##                                          max        range  skew kurtosis
## total_cases                        9.383979e+06 9.383978e+06 12.55   180.09
## new_cases                          1.012730e+05 1.095340e+05 10.98   143.21
## new_cases_smoothed                 9.319857e+04 9.375057e+04 10.98   143.57
## total_deaths                       2.326270e+05 2.326260e+05  8.83    94.61
## new_deaths                         4.928000e+03 6.846000e+03 11.58   217.56
## new_deaths_smoothed                2.715140e+03 2.947290e+03  9.06   109.18
## total_cases_per_million            6.354753e+04 6.354753e+04  3.77    18.74
## new_cases_per_million              8.652660e+03 1.086520e+04 22.28  1128.33
## new_cases_smoothed_per_million     2.472190e+03 2.742170e+03  7.55   101.39
## total_deaths_per_million           1.237550e+03 1.237550e+03  3.26    12.69
## new_deaths_per_million             2.153800e+02 2.832800e+02 28.62  1505.57
## new_deaths_smoothed_per_million    6.314000e+01 7.282000e+01  8.98   138.51
## icu_patients                               -Inf         -Inf    NA       NA
## icu_patients_per_million                   -Inf         -Inf    NA       NA
## hosp_patients                              -Inf          -Inf    NA        NA
## hosp_patients_per_million                  -Inf          -Inf    NA        NA
## weekly_icu_admissions                      -Inf          -Inf    NA        NA
## weekly_icu_admissions_per_million          -Inf          -Inf    NA        NA
## weekly_hosp_admissions                     -Inf          -Inf    NA        NA
## weekly_hosp_admissions_per_million         -Inf          -Inf    NA        NA
## stringency_index                   1.000000e+02 1.000000e+02 -0.57     -0.57
## population                         1.439324e+09 1.439323e+09  7.93     66.34
## population_density                 1.934750e+04 1.934736e+04  9.88    105.70
## median_age                         4.820000e+01 3.310000e+01 -0.01     -1.24
## aged_65_older                      2.705000e+01 2.591000e+01  0.66     -0.87
## aged_70_older                      1.849000e+01 1.797000e+01  0.80     -0.54
## gdp_per_capita                     1.169356e+05 1.162744e+05  1.65      3.45
## extreme_poverty                    7.760000e+01 7.750000e+01  1.76      2.13
## cardiovasc_death_rate              7.244200e+02 6.450500e+02  0.90      0.83
## diabetes_prevalence                3.053000e+01 2.954000e+01  1.10      1.46
## female_smokers                     4.400000e+01 4.390000e+01  0.89     -0.31
## male_smokers                       7.810000e+01 7.040000e+01  0.55      0.30
## handwashing_facilities             9.900000e+01 9.781000e+01 -0.11     -1.48
## hospital_beds_per_thousand         1.380000e+01 1.370000e+01  1.76      3.91
## life_expectancy                    8.675000e+01 3.347000e+01 -0.74     -0.15
## human_development_index            9.500000e-01 6.000000e-01 -0.48     -0.76
##                                           se
## total_cases                          2043.20
## new_cases                              21.65
## new_cases_smoothed                     21.20
## total_deaths                           73.42
## new_deaths                              0.54
## new_deaths_smoothed                     0.50
## total_cases_per_million                24.47
## new_cases_per_million                   0.53
## new_cases_smoothed_per_million          0.38
## total_deaths_per_million                0.86
## new_deaths_per_million                  0.01
## new_deaths_smoothed_per_million         0.01
## icu_patients                              NA
## icu_patients_per_million                  NA
## hosp_patients                             NA
## hosp_patients_per_million                 NA
## weekly_icu_admissions                     NA
## weekly_icu_admissions_per_million         NA
## weekly_hosp_admissions                    NA
## weekly_hosp_admissions_per_million        NA
## stringency_index                        0.12
## population                         668533.14
## population_density                      7.27
## median_age                              0.04
## aged_65_older                           0.03
## aged_70_older                           0.02
## gdp_per_capita                         93.48
## extreme_poverty                         0.11
## cardiovasc_death_rate                   0.54
## diabetes_prevalence                     0.02
## female_smokers                          0.05
## male_smokers                               0.07
## handwashing_facilities                     0.21
## hospital_beds_per_thousand                 0.01
## life_expectancy                            0.03
## human_development_index                    0.00
```

______________*Practice*________________

- Check the skewness of the *new_cases_per_million* variable. (skewness values between 1 and -1 indicate a roughly normal distribution)
- Check the number of valid cases in the dataset for the variable *gdp_per_capita*

____

Factors

Lets meet a new vector class: **factor**. Factors are just like character vectors. They contain character strings as data. The main difference is that character vectors can contain any values as long as they are character values, while factors can only contain **specified values**. This means that we can set what are the valid values for a given factor, and this information can be used by certain functions.

For example the variable **continent** can be set as a factor variable, since there are only certain values that are meaningful as "continents" in this dataset: (North America, Asia, Africa, Europe, South America, Oceania).

We can use the **factor()** function to coerce a vector or variable to be a factor. R will automatically set the accepted values based on the unique values in the vector.

```
COVID_data <- COVID_data %>%
              mutate(continent = factor(continent),
                     location = factor(location))
```

The **levels()** function displays the levels (valid values) of the factor.

When using the **table()** function we cen get a table of the number of cases within each level.

When we simply print a factor, we will also get information about its levels.

```
levels(COVID_data$continent)

table(COVID_data$continent)

COVID_data$continent
```

Now we create a daset that only contains the data published for the latest day (we use the max(COVID_data$date) code segment to determine the latest date in the dataset).

```
COVID_data_latest = COVID_data %>%
  filter(date == max(COVID_data$date))
```

After designating some varialbes as factors, the summary() function will display more information about it: It will display the frequency of cases within each factor level.

```
COVID_data_latest %>%
  select(continent) %>%
  summary()
```

```
##          continent
##  Africa       :55
##  Asia         :46
##  Europe       :48
##  North America:36
##  Oceania      :11
##  South America:13
```

Lets **exclude** oceania from our dataset using the **filter()** function to see how R reacts to losing all observations from one of the valid factor levels.

One thing to realize in this case is that R remembers all valid factor levels, even if all of the cases within that level are gone.

```
COVID_data_latest %>%
  filter(continent != "Oceania") %>%
  select(continent) %>%
  summary()
```

```
##          continent
##  Africa       :55
##  Asia         :46
##  Europe       :48
##  North America:36
##  Oceania      : 0
##  South America:13
```

Sometimes, we don't want to keep remembering these factor levels. We can force R to remove factor levels that are no longer "in use" by any of the cases by using the **droplevels()**function.

```
COVID_data_latest_noOceania = COVID_data_latest %>%
  filter(continent != "Oceania") %>%
  mutate(continent = droplevels(continent))

COVID_data_latest_noOceania %>%
  select(continent) %>%
  summary()
```

```
##          continent
##  Africa       :55
##  Asia         :46
##  Europe       :48
##  North America:36
##  South America:13
```

Relation of factor levels

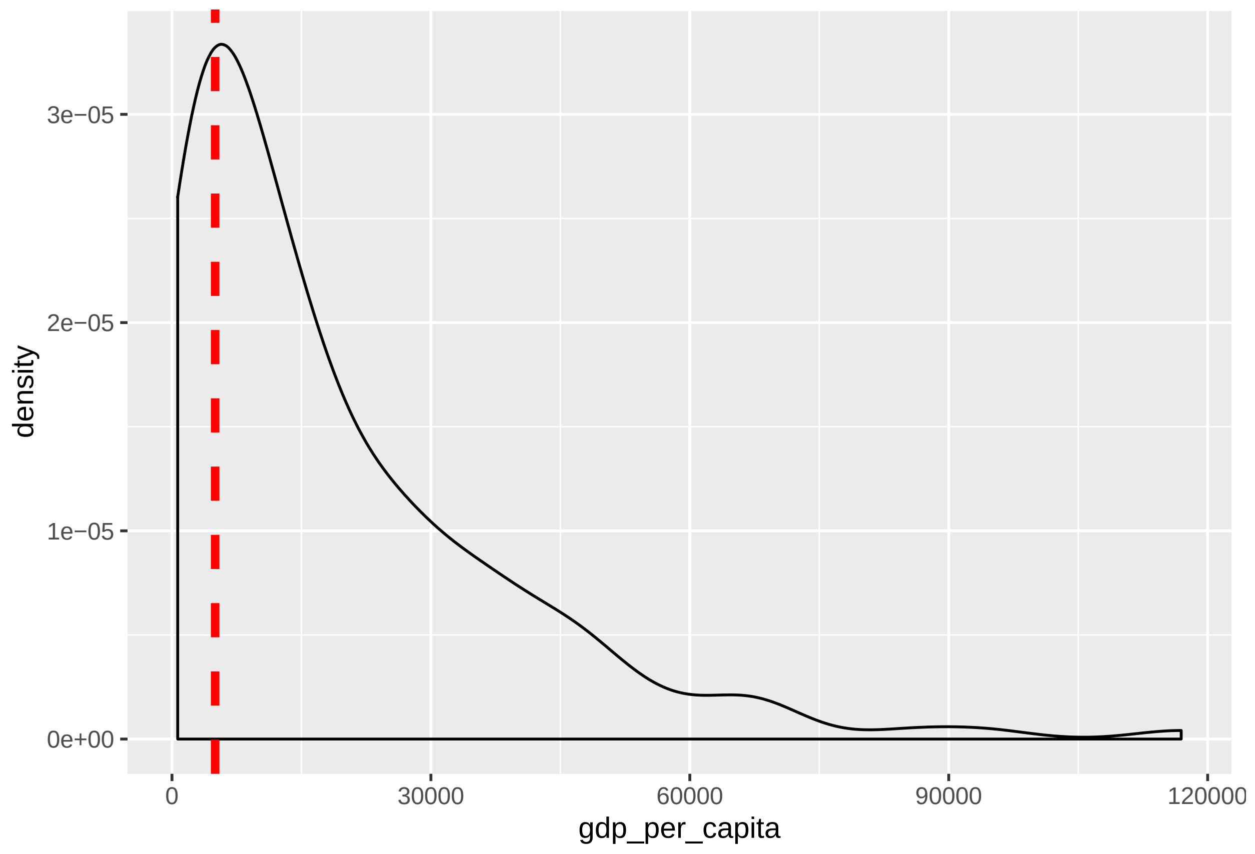
So far we have worked with nominal variables, variables in which the factor levels are not ordered. However during data analysis we might encounter ordered categories, for example education level (no education < primary school < middleschool < highschool < higher education . . .). This dataset does not contain such categorical variables, so let's create some:

As we have seen earlier, we can use **case_when()** to designate categories based on values on a numerical variable. For example if we want to compare COVId statistics between countries with gdp_per_capita below 5000 with those with medium or high gdps. If we look at the distribution of gdp_per_capita we can see that the mode of the distribution is somewhere around 5000, but this is a very scewed distribution.

```
COVID_data_latest %>%
  select(gdp_per_capita, continent) %>%
  drop_na() %>%
  group_by(continent) %>%
  summarize(mean_gdp = mean(gdp_per_capita))
```

```
## # A tibble: 6 x 2
##   continent     mean_gdp
##   <fct>            <dbl>
## 1 Africa           5444.
## 2 Asia            22185.
## 3 Europe          32750.
## 4 North America   21655.
## 5 Oceania         16548.
## 6 South America   13841.
```

```
COVID_data_latest %>%
  select(gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
  aes(x = gdp_per_capita) +
  geom_density() +
  geom_vline(xintercept = 5000, linetype="dashed",
              color = "red", size=1.5)
```



So lets create a categorical variable based on this:

Note that by using the factor() function we already designate this variable as a factor while it is being created.

```
COVID_data = COVID_data %>%
  mutate(gdp_per_capita_kat = factor(
                                    case_when(gdp_per_capita < 5000 ~ "small",
                                              gdp_per_capita >= 5000 & gdp_per_capita < 10000 ~ "medi
                                              gdp_per_capita > 10000 ~ "large")))
```

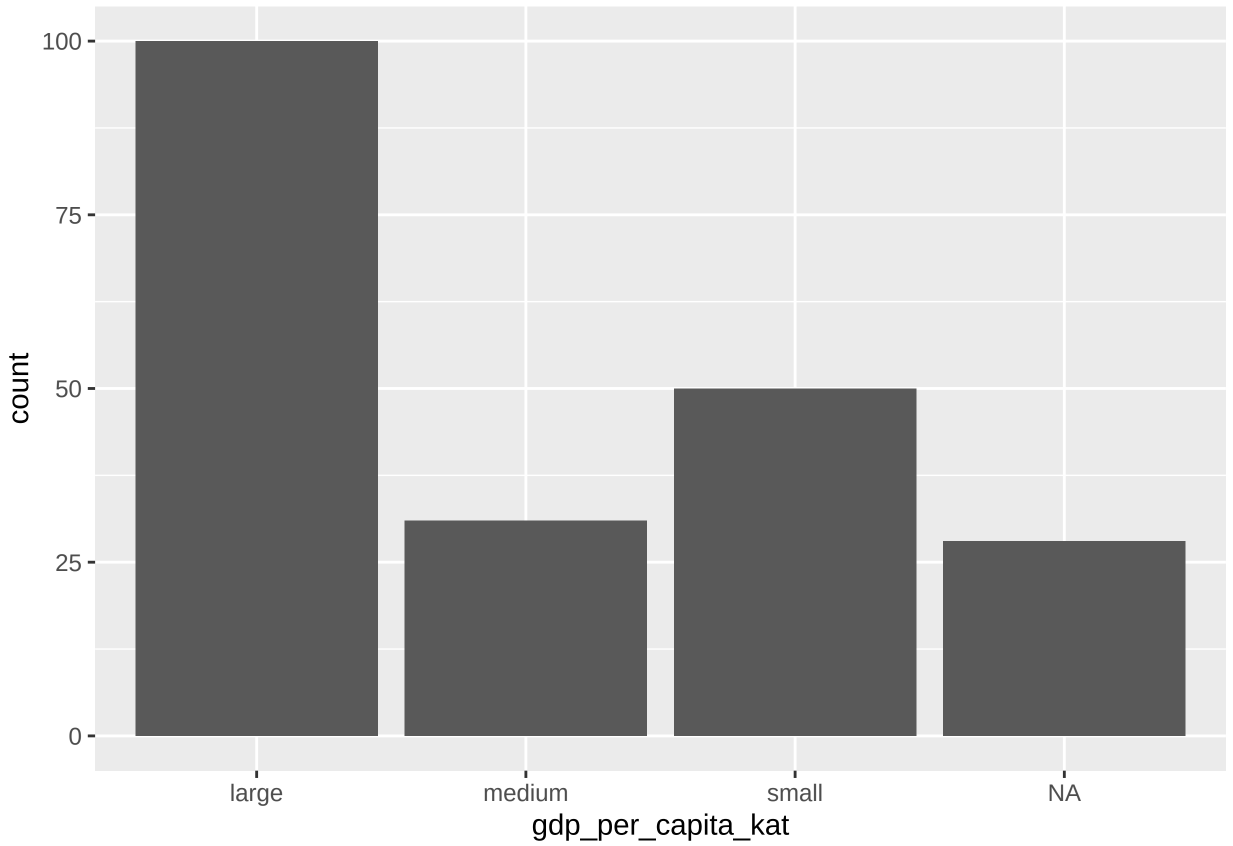
```
levels(COVID_data$gdp_per_capita_kat)
```

```
## [1] "large"  "medium" "small"
```

```
# ugyanez a COVID_data_latest -al

COVID_data_latest = COVID_data_latest %>%
  mutate(gdp_per_capita_kat = factor(
                                    case_when(gdp_per_capita < 5000 ~ "small",
                                              gdp_per_capita >= 5000 & gdp_per_capita < 10000 ~ "medi
                                              gdp_per_capita > 10000 ~ "large")))
```

When we plot the data using ggplot (using the geom_bar for a barchart), we can see that the factor levels are displayed as "large", "medium", and "small", from left to right.

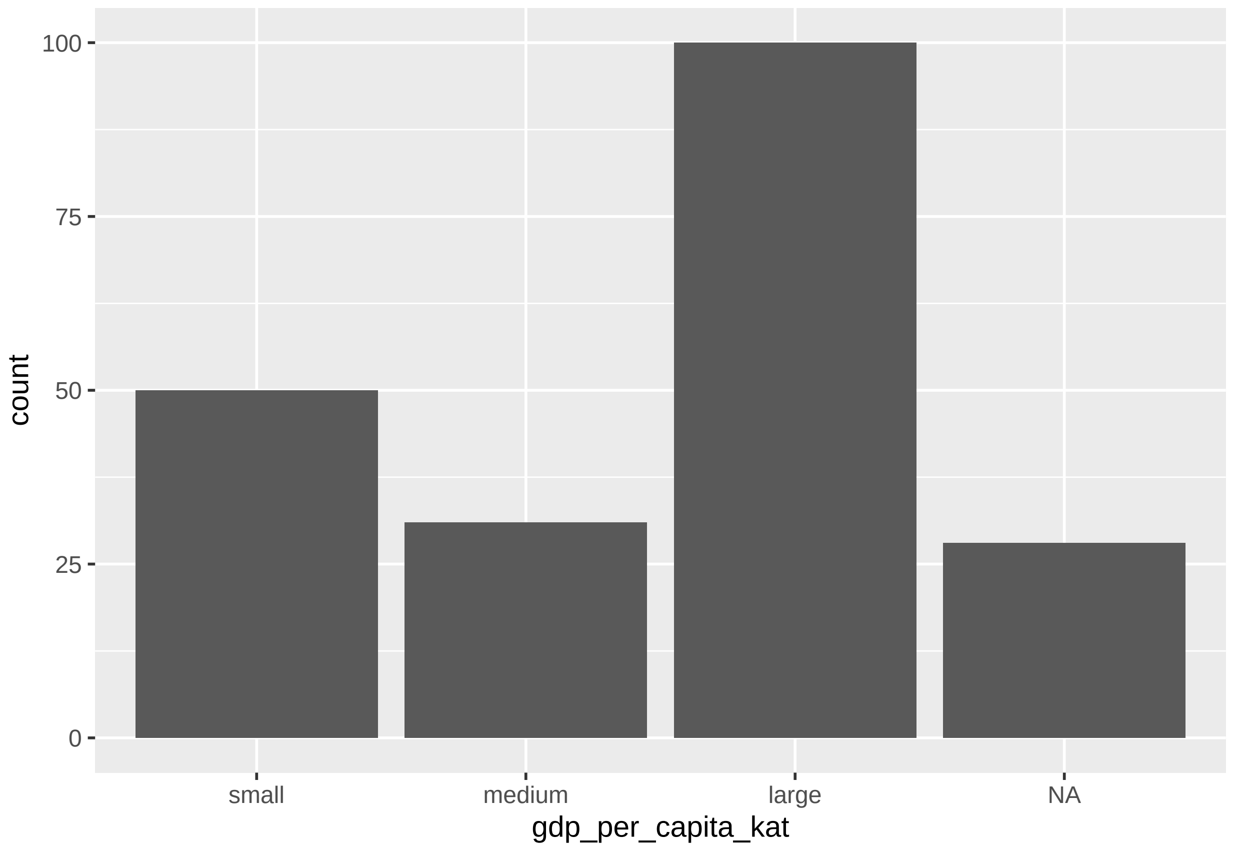
```
COVID_data_latest %>%
  ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar()
```



This might seem a little off at first sight, since we usually display things in an ascending order from left to right, so "small" might be expected to be displayed at the left side of the graph. The reason for R using this display sequence is that R orders the factor levels by default based on **alphabetical order**.

We can change the order in which factor levels are listed and displayed by R by forcing this order. Within the factor() we can list the levels in the desired order using the **levels = c()** parameter.

```
COVID_data_latest = COVID_data_latest %>%
mutate(gdp_per_capita_kat = factor(gdp_per_capita_kat, levels = c(
                                          "small",
                                           "medium",
                                           "large")))

COVID_data_latest %>%
  ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar()
```



We can also specify to R that this factor is an ordinal (ordered) factor, in which case, R will recodnise the hierarchy/order of the factor levels, and can use this information in certain statistical tests. This can be done by using the , **ordered = T** paramater.

If we do this, notice that when listing this vector R will put relation signs between the factor levels.

```
COVID_data_latest = COVID_data_latest %>%
mutate(gdp_per_capita_kat = factor(gdp_per_capita_kat, ordered = T, levels = c(
                                          "small",
                                           "medium",
                                           "large")))
COVID_data_latest$gdp_per_capita_kat
```

```
##   [1] small  large  large  <NA>   medium <NA>   large  large  medium large
##  [11] large  large  large  large  large  small  large  large  medium small
##  [21] large  medium medium <NA>   large  large  large  <NA>   large  large
##  [31] small  small  small  small  large  medium large  small  small  large
##  [41] large  large  small  small  large  small  large  <NA>   <NA>   large
##  [51] large  small  large  small  medium large  large  large  medium large
##  [61] small  large  small  <NA>   <NA>   medium large  large  <NA>   large
##  [71] small  medium large  small  <NA>   large  <NA>   large  <NA>   medium
##  [81] <NA>   small  small  medium small  small  large  large  medium large
##  [91] large  large  large  <NA>   large  large  medium large  <NA>   medium
## [101] large  small  medium large  small  medium large  large  small  small
## [111] large  <NA>   large  large  large  small  small  large  large  small
## [121] large  small  small  large  large  medium <NA>   large  large  <NA>
## [131] medium small  medium medium small  large  <NA>   large  medium small
## [141] medium <NA>   large  large  medium small  large  small  medium large
## [151] medium large  large  large  large  large  large  small  large  large
## [161] large  large  small  large  small  large  large  small  large  large
## [171] large  large  small  <NA>   large  large  small  large  small  large
## [181] medium large  <NA>   <NA>   small  small  large  medium small  large
## [191] large  large  <NA>   small  medium large  large  large  <NA>   large
## [201] medium <NA>   large  medium <NA>   <NA>   small  small  small
## Levels: small < medium < large
```

Practice

- Filter the dataset so that we only work with data from 2020-09-28 and save it as a new object.
- Create a new categorical variable (called *new_cases_per_million_kat*) with countries with *new_cases_per_million* under 20 coded as "small", and countries with *new_cases_per_million* 20 or above coded as "large".
- Pay attention to designate this new variable as factor.
- Save this new variable into the data object so that we can work with this variable later.
- Create a table showing the number of observations belonging to each factor level.
- Designate the correct oder of the factor levels: small, large (make sure that this is saved into the original data object as well)
- Check whether the factor levels are displayed in the right order.

Exploration through visualization

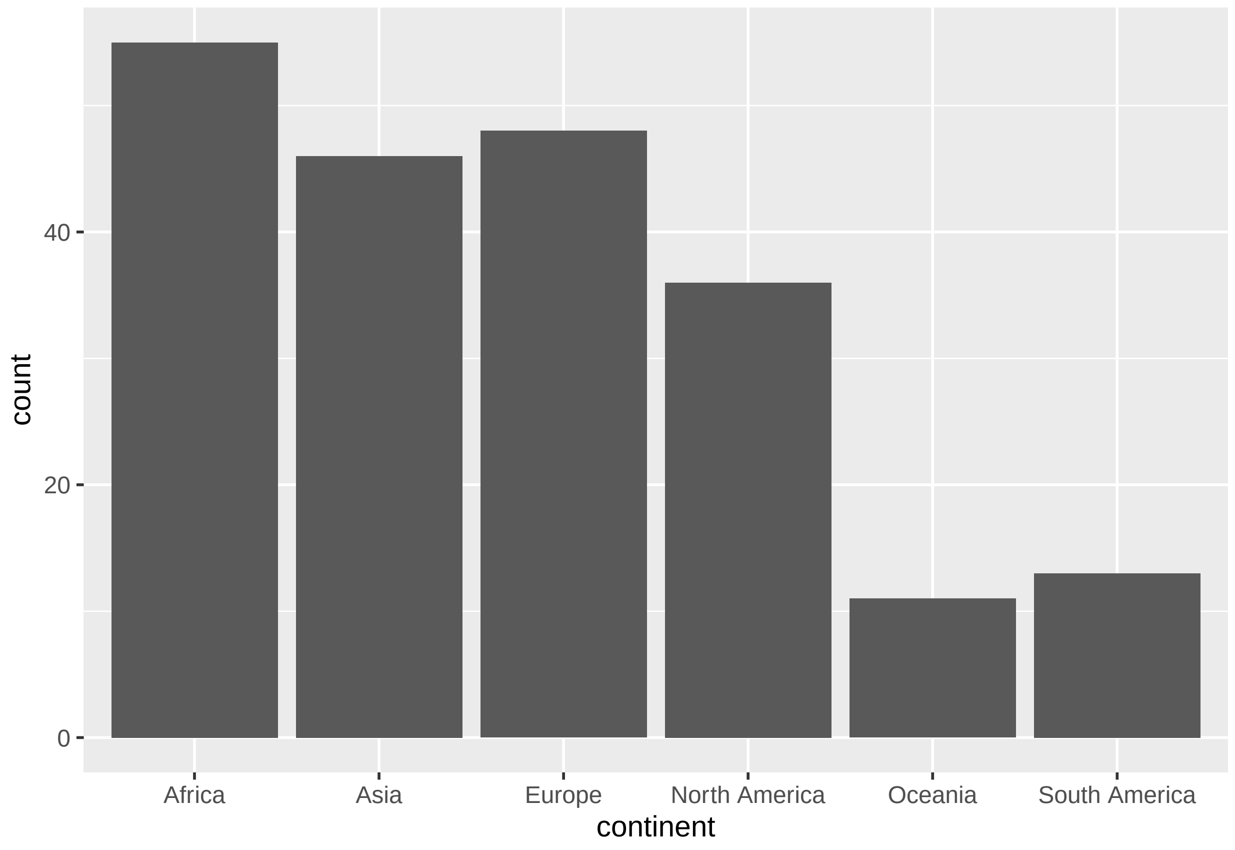
Before data analysis we should always do explorative analysis of the data to get a picture of the data we are working with, and to identify errors. One of the main tools for exploration is visualization.

Visualizing univariate distributions and frequencies

When visualizing variables, one of the main considerations when choosing the plot type to sue is the type of the data we are working with.

The distribution (frequencies) of categorical variables is most often visualized using barcharts (**geom_bar**).

```
COVID_data_latest %>%
ggplot() +
  aes(x = continent) +
  geom_bar()
```

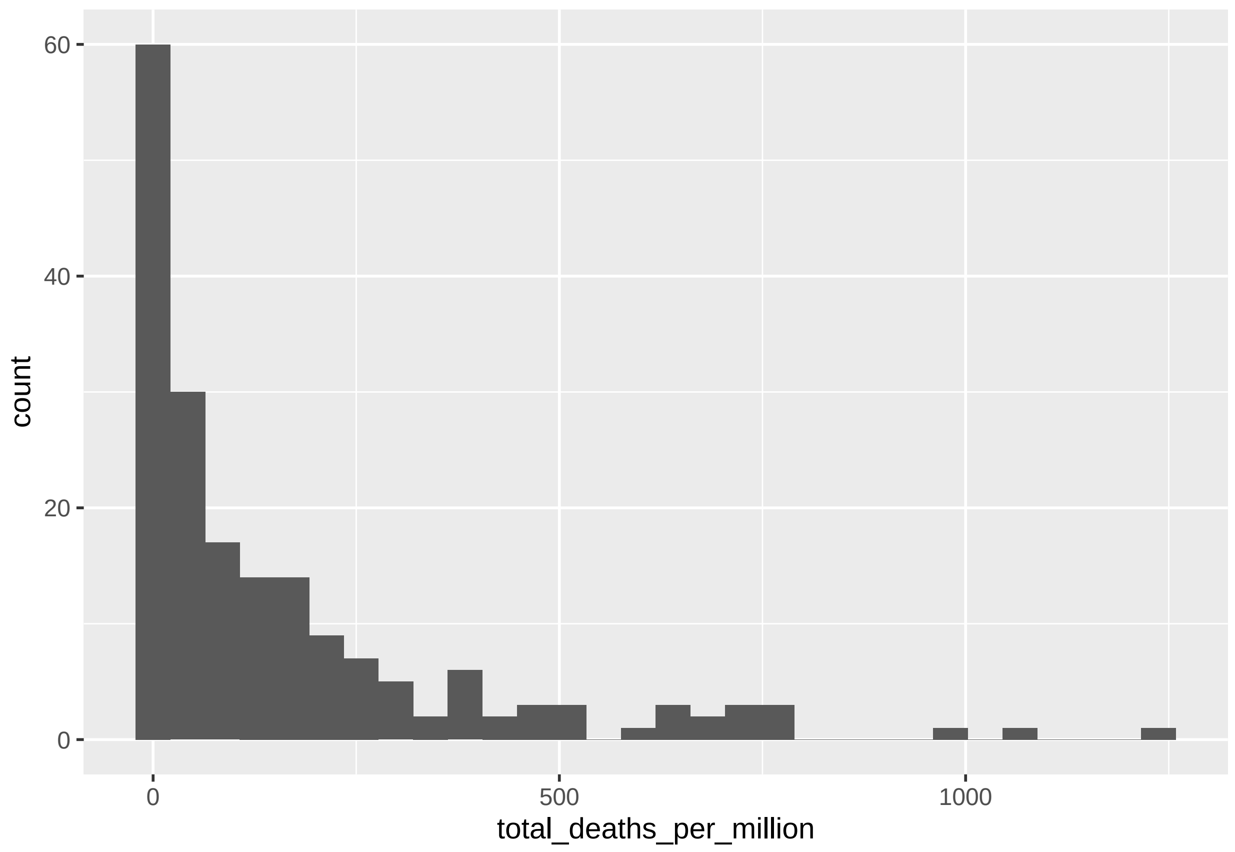


While the distribution of continuous variables are usually plotted using historgrams or density plots.

```
COVID_data_latest %>%
ggplot() +
  aes(x = total_deaths_per_million) +
  geom_histogram()
```

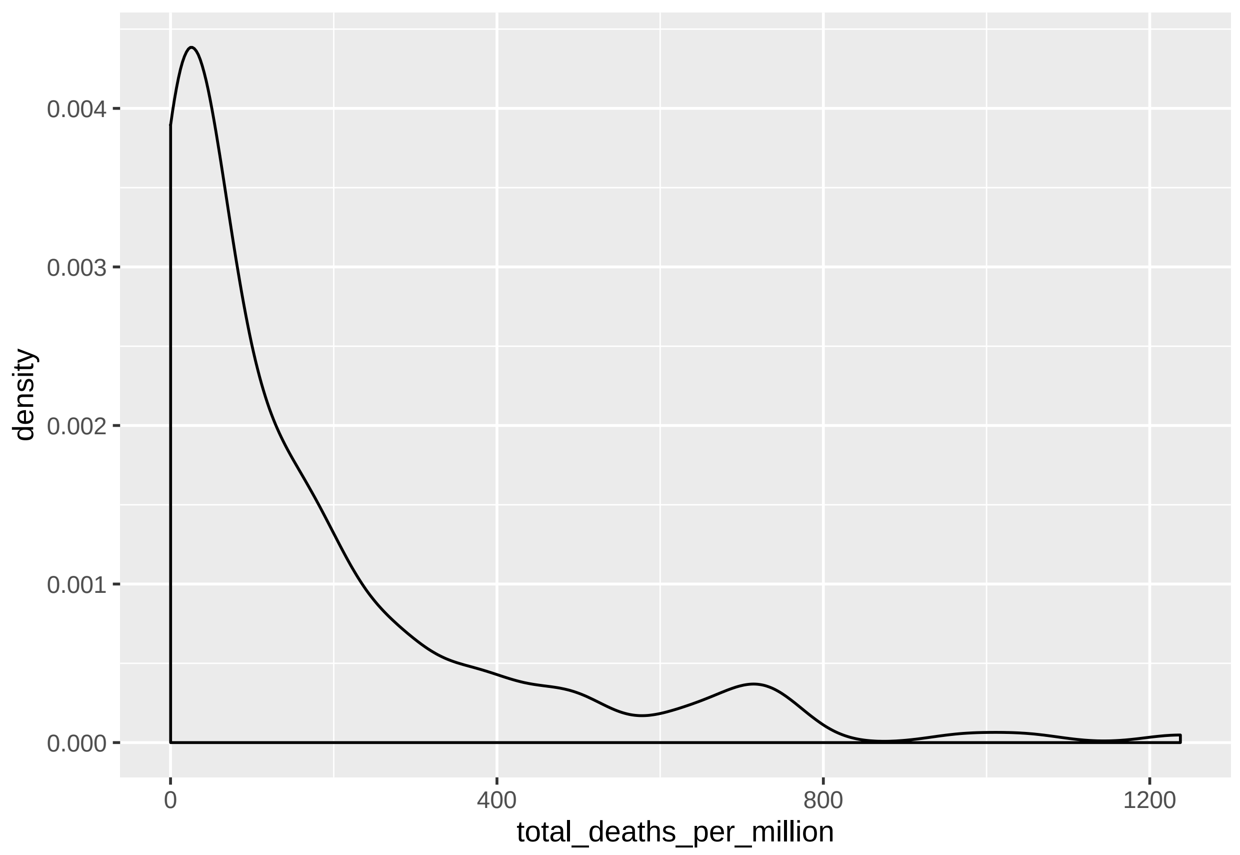
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 22 rows containing non-finite values (stat_bin).
```



```
COVID_data_latest %>%
ggplot() +
  aes(x = total_deaths_per_million) +
  geom_density()
```

```
## Warning: Removed 22 rows containing non-finite values (stat_density).
```



____________*Practice*________________

- filter the COVID dataset so that we only work with data from 2020-09-07.
- Using the previously learned methods to explore the univariate distributions and identify potential errors or unexplected values in the dataset.
- Aside from visualization you can also use the View(), summary(), and descibe() functions to get data about the minimum and maximum values and number of valid cases.
- For numerical data it is a good starting point to look at the minimal and maximal values for error detection.
- For categorcial variables it is a good idea to look at the number of observations belonging to each factor level.

____

Correcting errors

We can use the **mutate()** and **replace()** functions to correct incorrectly entered values.

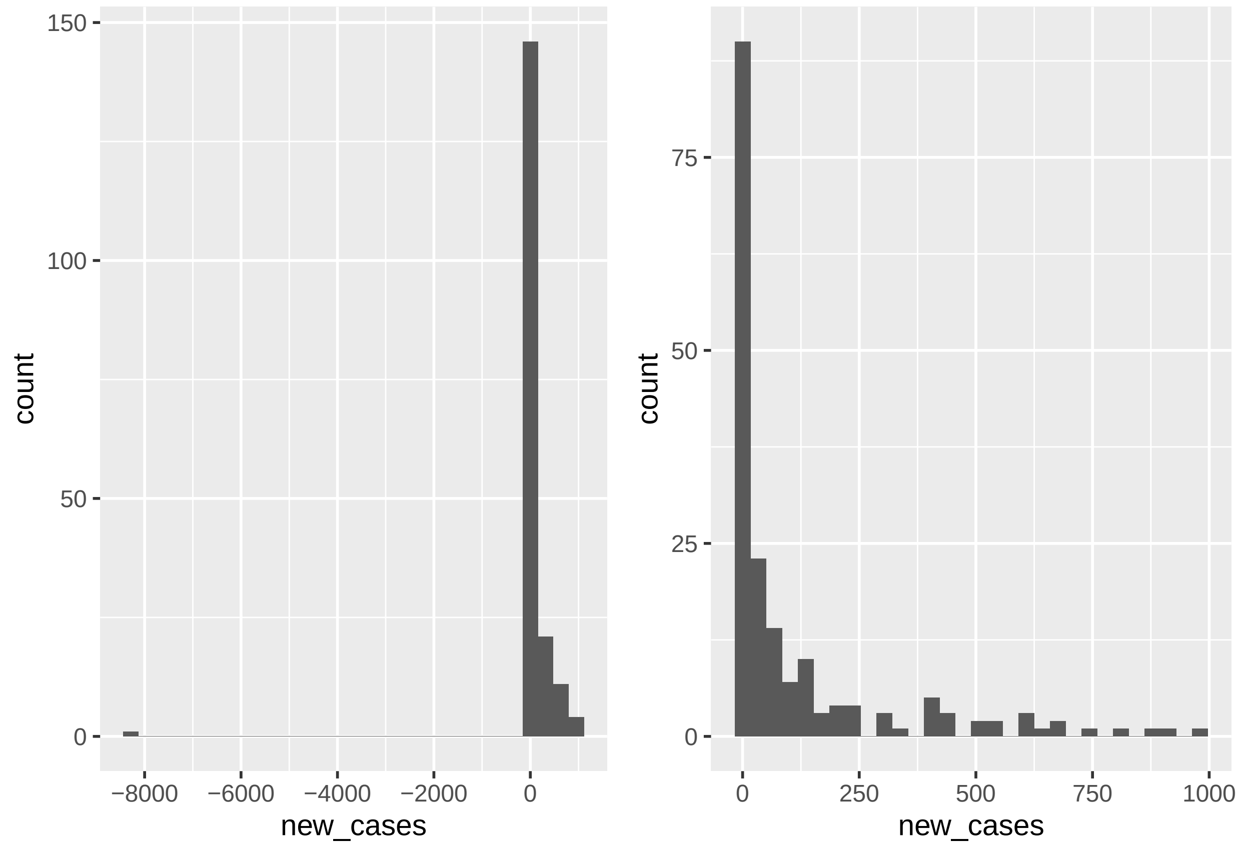
It is always a good idea to save the corrected data into a **new data object**, so the raw data is kept in an unmanipulated way. This can come in handy when you want to report the original (replaced) values in a paper or research report, or if you change your mind and want to use a different approach to handle the error.

```r
COVID_data_corrected <- COVID_data %>%
  mutate(new_cases = replace(new_cases,  new_cases=="-8261", NA))
```

It is important when replacing or recoding values to **double check** that the replacement/recode worked as intended. Below we check this using visualization. The raw original data is displayed on the left hand side panel, while the corrected data is shown on the right.

Notice how we use the **grid.arrange()** function from the gridExtra package to put these two plots side-by-side.

```
# hasznalhatnak meg az alabbiakat is arra,
# hogy megbizonyosodjunk abban, hogy sikeres volt a csere
# View(COVID_data_corrected)
# describe(COVID_data_corrected)
# summary(COVID_data_corrected$szocmedia_3)
# COVID_data_corrected$szocmedia_3

old_plot <-
  COVID_data %>%
  filter(date == "2020-09-07", new_cases < 1000) %>%
  ggplot()+
    aes(x = new_cases)+
    geom_histogram()

new_plot <-
  COVID_data_corrected %>%
  filter(date == "2020-09-07", new_cases < 1000) %>%
  ggplot()+
    aes(x = new_cases)+
    geom_histogram()


grid.arrange(old_plot, new_plot, ncol=2)
```



Exploring the relationship of multiple variables

We can use tables and plots to explore the relationship of two variables before running statistical test. Visualization and running these exploratory tables is always recommended before running the actual statistical tests.

Exploring the relationship of two categorcial variables

Exploratory analysis

Now lets explore the relationship between the categorical varialbe we created for GDP (*gdp_per_capita_kat*), and the continent (*continent*). Lets use the data only from the latest day in the dataset.

The easiest was to get descriptive data of the relationship of two categorical variables is to produce a **crosstab** using the **table()** function.

```
table(COVID_data_latest$gdp_per_capita_kat, COVID_data_latest$continent)
```

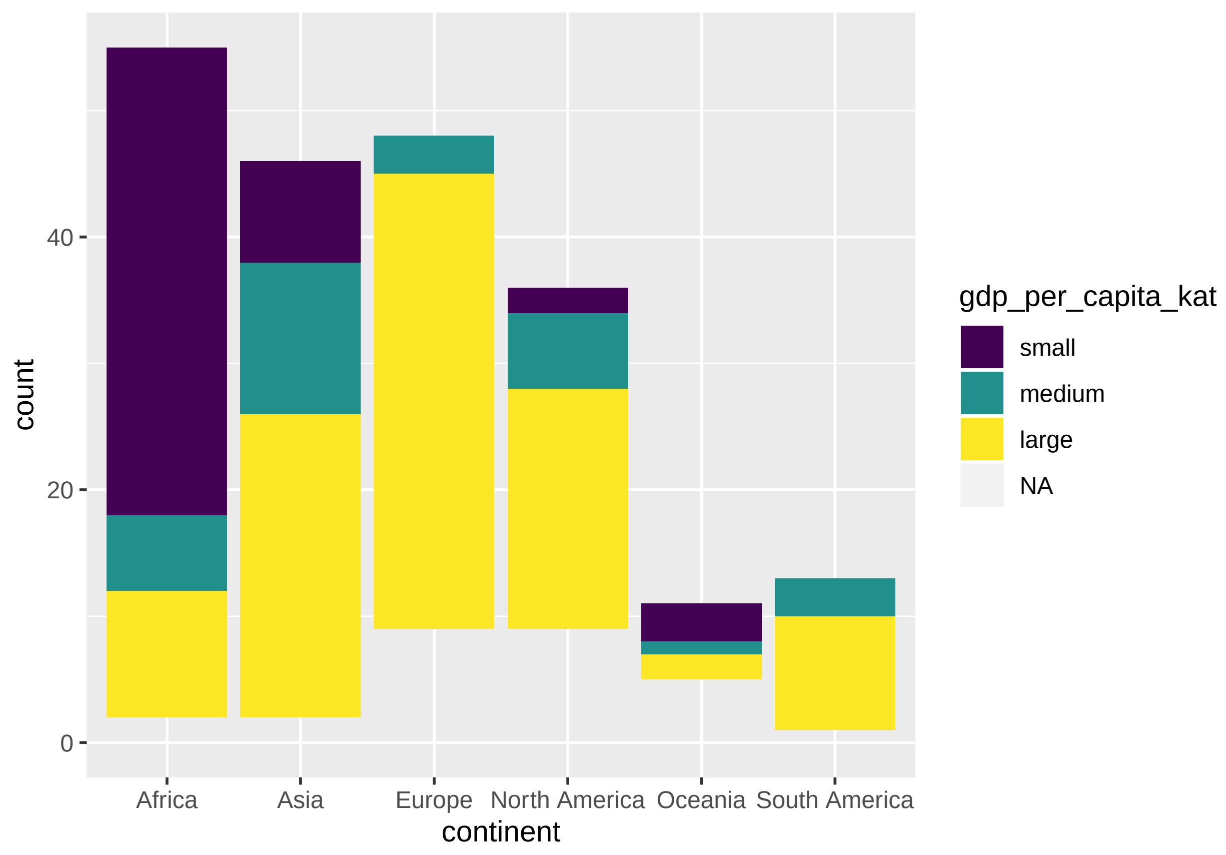
```
##
##          Africa Asia Europe North America Oceania South America
##   small      37    8      0             2       3             0
##   medium      6   12      3             6       1             3
##   large      10   24     36            19       2             9
```

This table shows that most of the small income countries are located in Africa, while most of the large income countries are located in Europe.

We can also use a barchart to visualize this relationship with **geom_bar()**.

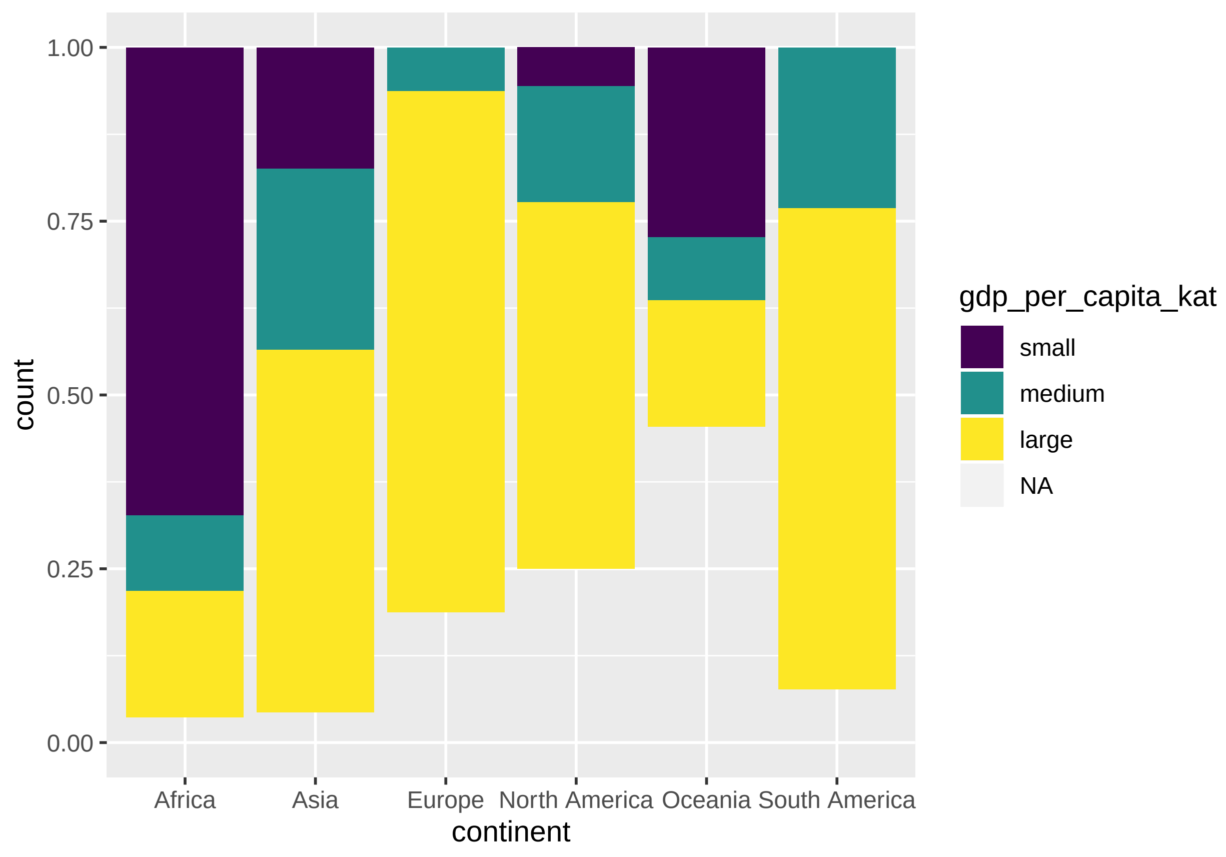
One of the methods is to use a **stacked bar chart**, which cna be created using **geom_bar()**. Since barcharts only accept vairalbe on the x axis, we can use the **"fill ="** parameter to add another variable element in the aes() function.

```
COVID_data_latest %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar()
```

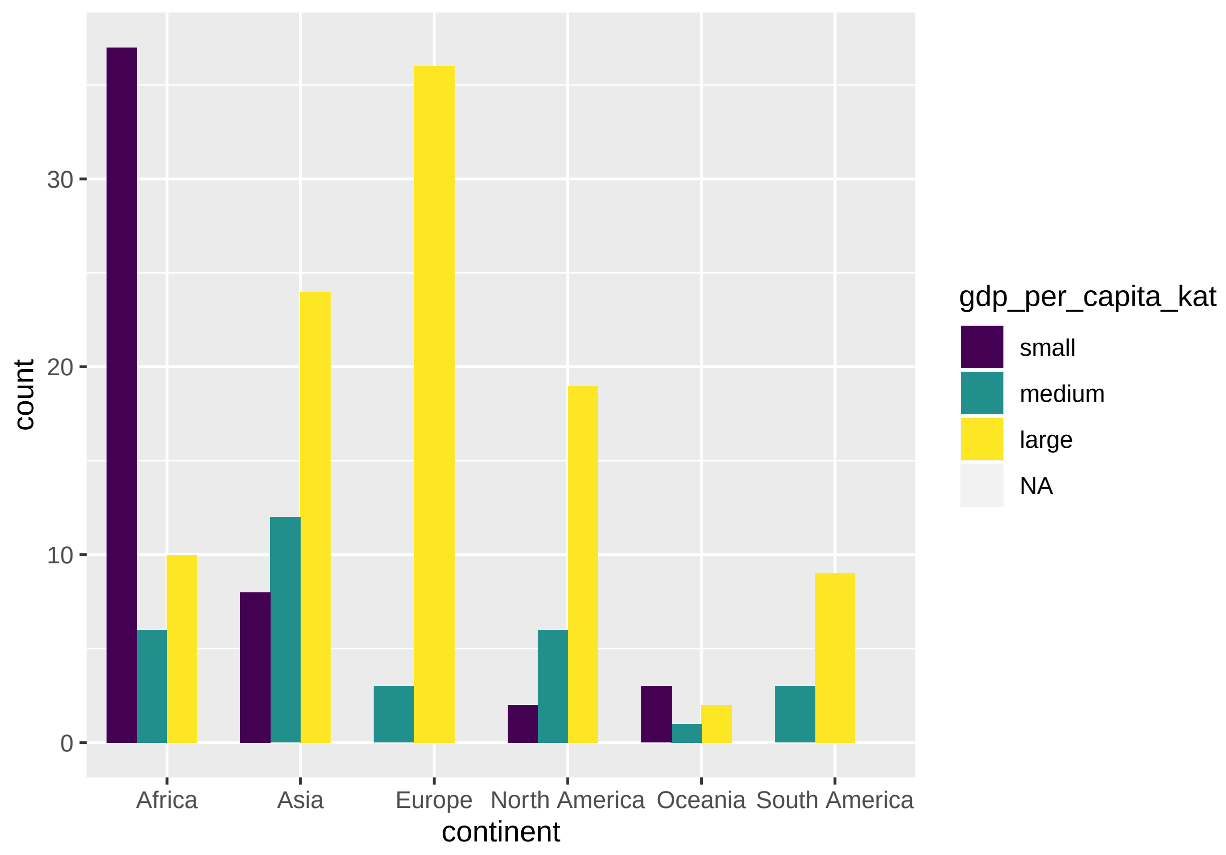


If the number of observations is very different for the different factor levels, the stacked barchart can be misleading or uninformative. So it is often useful to create different barcharts as well, for example a stacked barchart with y representing proportions instead of raw frequencies (use **position = "fill"** parameter within geom_bar() to achieve this.), or a dodged barchart (use **position = "dodge"** parameter within geom_bar() to achieve this).

```
COVID_data_latest %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill")
```



```r
COVID_data_latest %>%
ggplot() +
  aes(x = continent, fill = gdp_per_capita_kat) +
  geom_bar(position = "dodge")
```



________________*Practice*________________

Use the methods learned above to explore the relationship of **new_cases_per_million_kat** (created in the practice exercise above) and **continent** in the COVID_data_latest dataset. - use the **geom_bar()** geom for visualization - use different types of barcharts to give a more detailed picture of the relationship of the two variables. - What conclusion can we arrive at by looking at the plots?

____

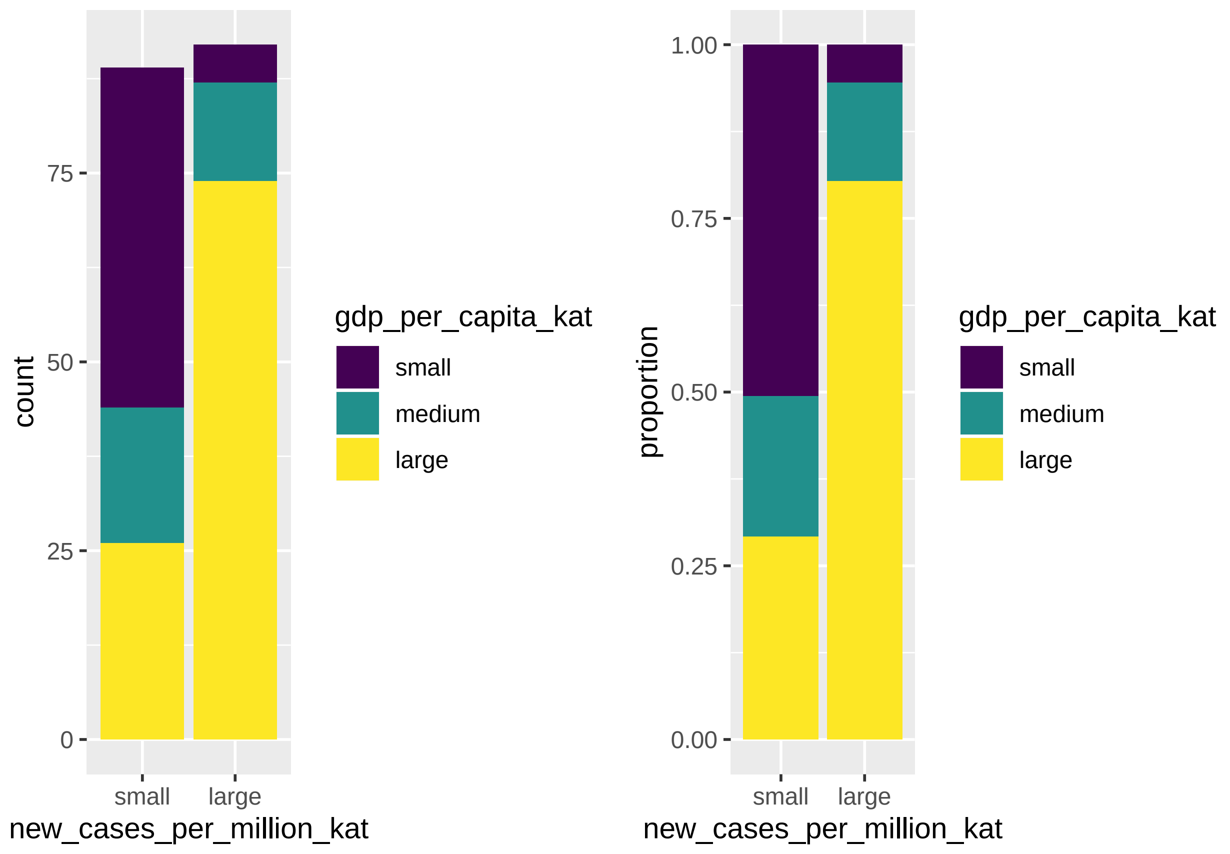
```
# If you havent created this variable yet, you can create the new_cases_per_million_kat variable in the

COVID_data = COVID_data %>%
  mutate(new_cases_per_million_kat = factor(
                                    case_when(new_cases_per_million < 20 ~ "small",
                                              new_cases_per_million >= 20 ~ "large"), ordered = T, le

levels(COVID_data$new_cases_per_million_kat)
```

```
## [1] "small" "large"
```

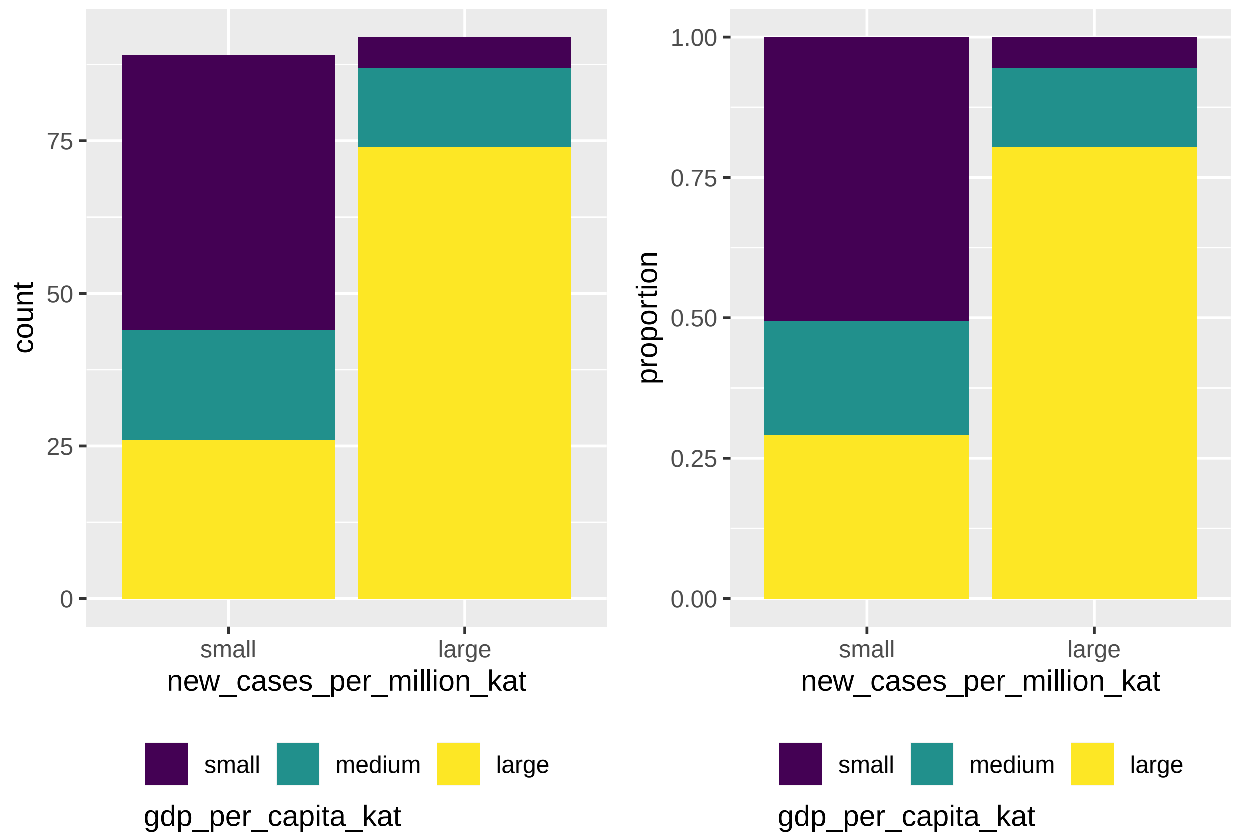
```
# The same for the COVID_data_latest dataset:

COVID_data_latest = COVID_data_latest %>%
  mutate(new_cases_per_million_kat = factor(
                                    case_when(new_cases_per_million < 20 ~ "small",
                                              new_cases_per_million >= 20 ~ "large"), ordered = T, le
```

We can control how and where the plot legend is displayed using the theme(legend.position) and the guides() functions. This can improve the interpretability of the plot.

```
barchart_plot_3 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar()


barchart_plot_4 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill") +
  ylab("proportion")

grid.arrange(barchart_plot_3, barchart_plot_4, ncol=2)
```

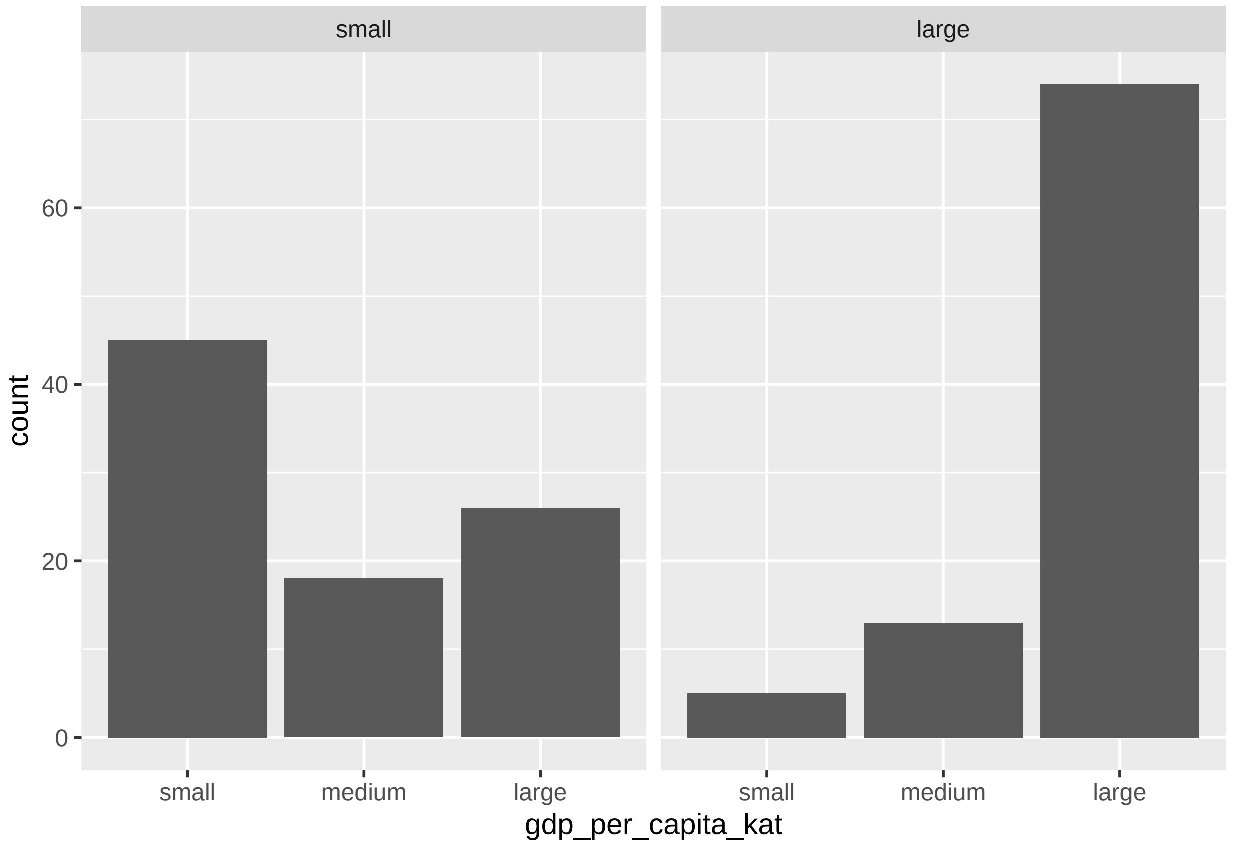


```
# a theme(legend.position) es a guides() funciok
# hasznalataval kontrollalhatjuk hogy hol es hogyan
# jelenjen meg a jelmagyarazat az abran

barchart_plot_3 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
    aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
    geom_bar() +
    theme(legend.position="bottom") +
    guides(fill = guide_legend(title.position = "bottom"))


barchart_plot_4 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = new_cases_per_million_kat, fill = gdp_per_capita_kat) +
  geom_bar(position = "fill") +
  theme(legend.position="bottom") +
  guides(fill = guide_legend(title.position = "bottom")) +
  ylab("proportion")

grid.arrange(barchart_plot_3, barchart_plot_4, ncol=2)
```



Yet another way of displaying plots of multiple variables is to put them on separate plot panels. We can use the **facet_wrap()** function to create this faceted (**multi-panel**) plot like in the example below:

```r
barchart_plot_6 <-
COVID_data_latest %>%
  select(new_cases_per_million_kat, gdp_per_capita_kat) %>%
  drop_na() %>%
ggplot() +
  aes(x = gdp_per_capita_kat) +
  geom_bar() +
  facet_wrap(~ new_cases_per_million_kat)

barchart_plot_6
```



Exploring the relationship if a categorical and a continuous variable

Lets explore gdp_per_capita on the different continents. In this case we will use gdp_per_capita as a continuous variable, and we would like to assess its relationship with a categoricl variable: continent.

We can start the exploration by producing descriptive tables. As we have learned earlier, this ispossible by using the combination of **group_by()** amd **summarize()** functions. There are also some countries with no data about gdp, so in order to get means and standard deviations we will need to drop these cases from the dataset using the **drop_na()** function.

We can do all this within one code chain using the pipe operator.

The summary table shows that the average gdp per capita in Africa is 5444, while in Europe it is 32750.

```r
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  group_by(continent) %>%
    summarize(mean = mean(gdp_per_capita),
              sd = sd(gdp_per_capita))
```

```
## # A tibble: 6 x 3
##   continent       mean     sd
##   <fct>          <dbl>  <dbl>
## 1 Africa          5444.  6183.
## 2 Asia           22185. 25406.
## 3 Europe         32750. 18526.
## 4 North America 21655. 15404.
## 5 Oceania        16548. 18775.
## 6 South America 13841.  5110.
```

Lets **visualize** this same relationship now.

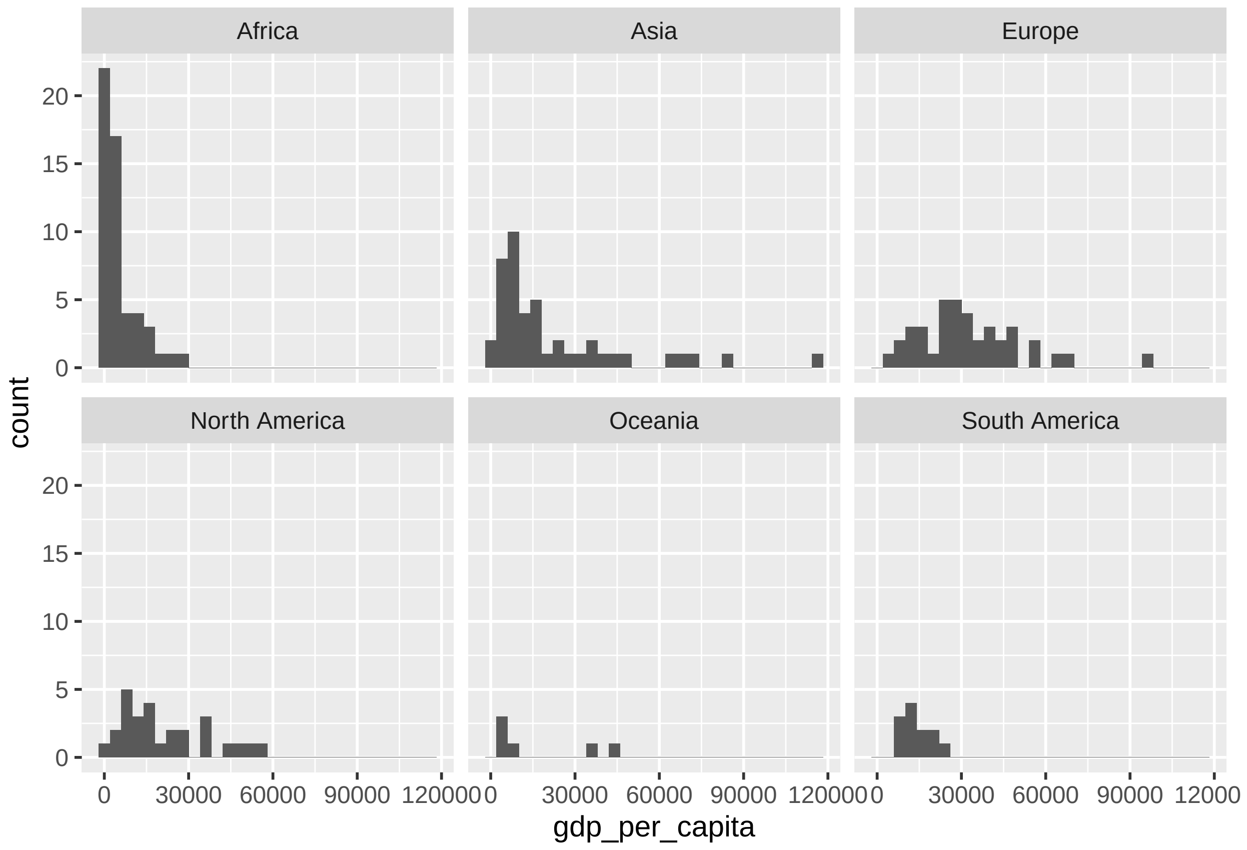
We have multipl options to choose from.

- we can us **facet_wrap()** and apply **geom_histogram()** or **geom_dotplot()**
- one of the most common solutions is to use **geom_boxplot()**
- we can also use **geom_density()** with colors representing the different continents.
- my favorite solution is the **geom_violin()**, combined with **geom_jitter()**

We can use multiple types of plots to get a more complex picture of the relationship.

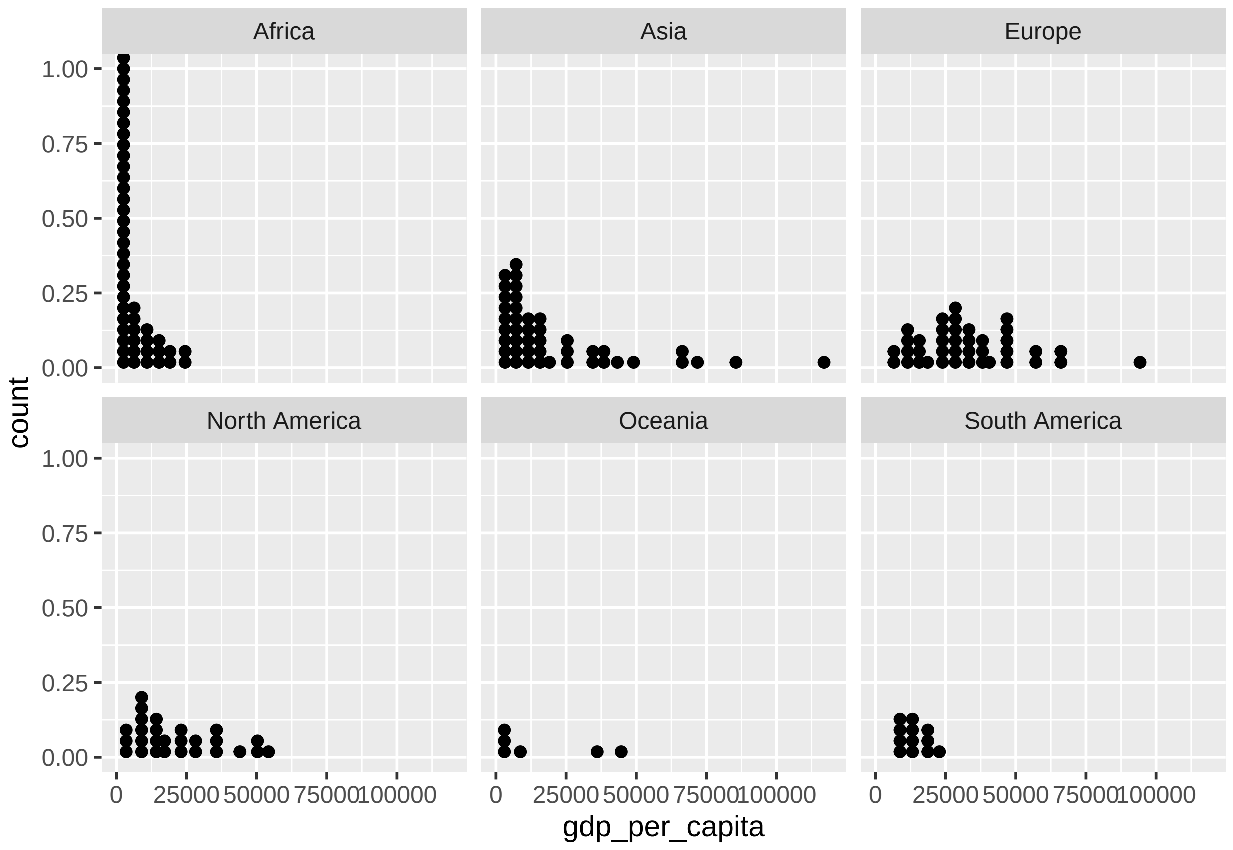
```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita) +
    geom_histogram() +
    facet_wrap(~ continent)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

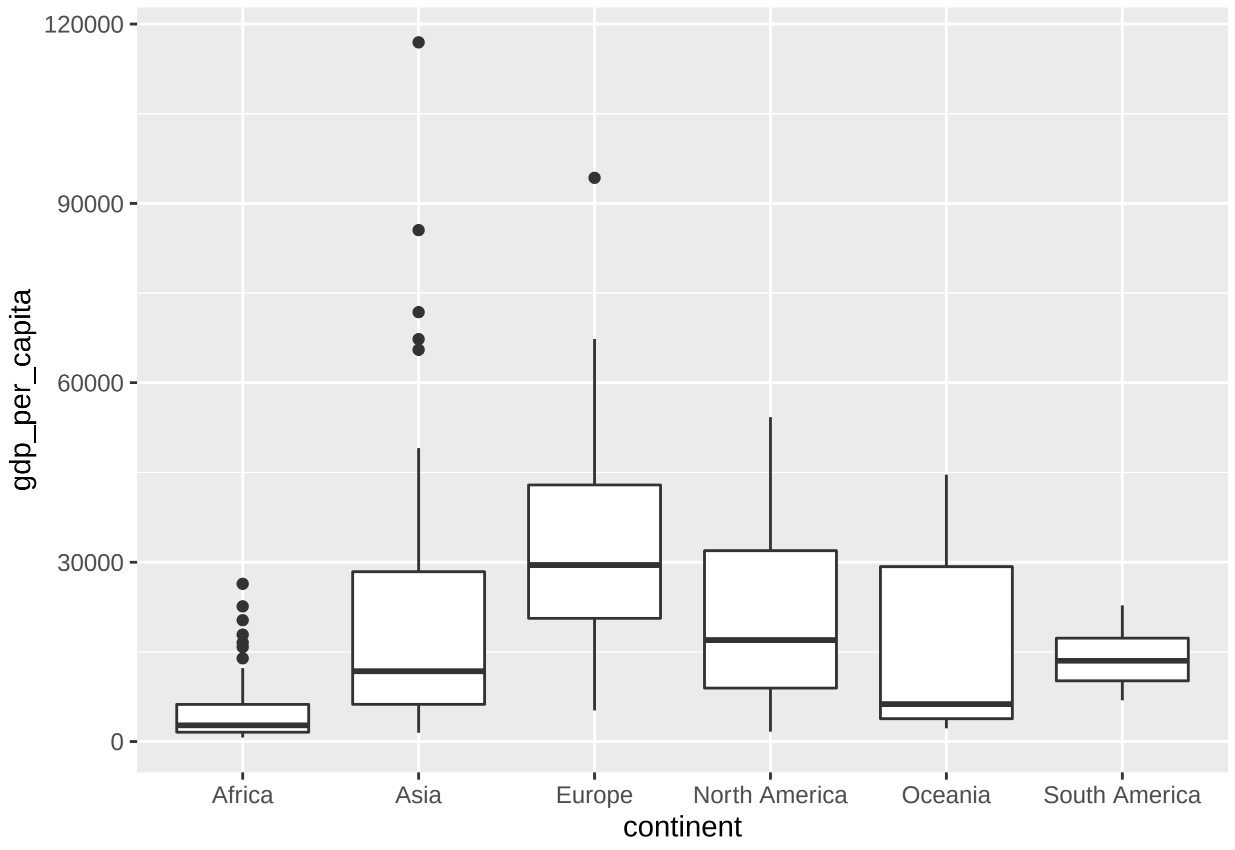


```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita) +
    geom_dotplot() +
    facet_wrap(~ continent)
```

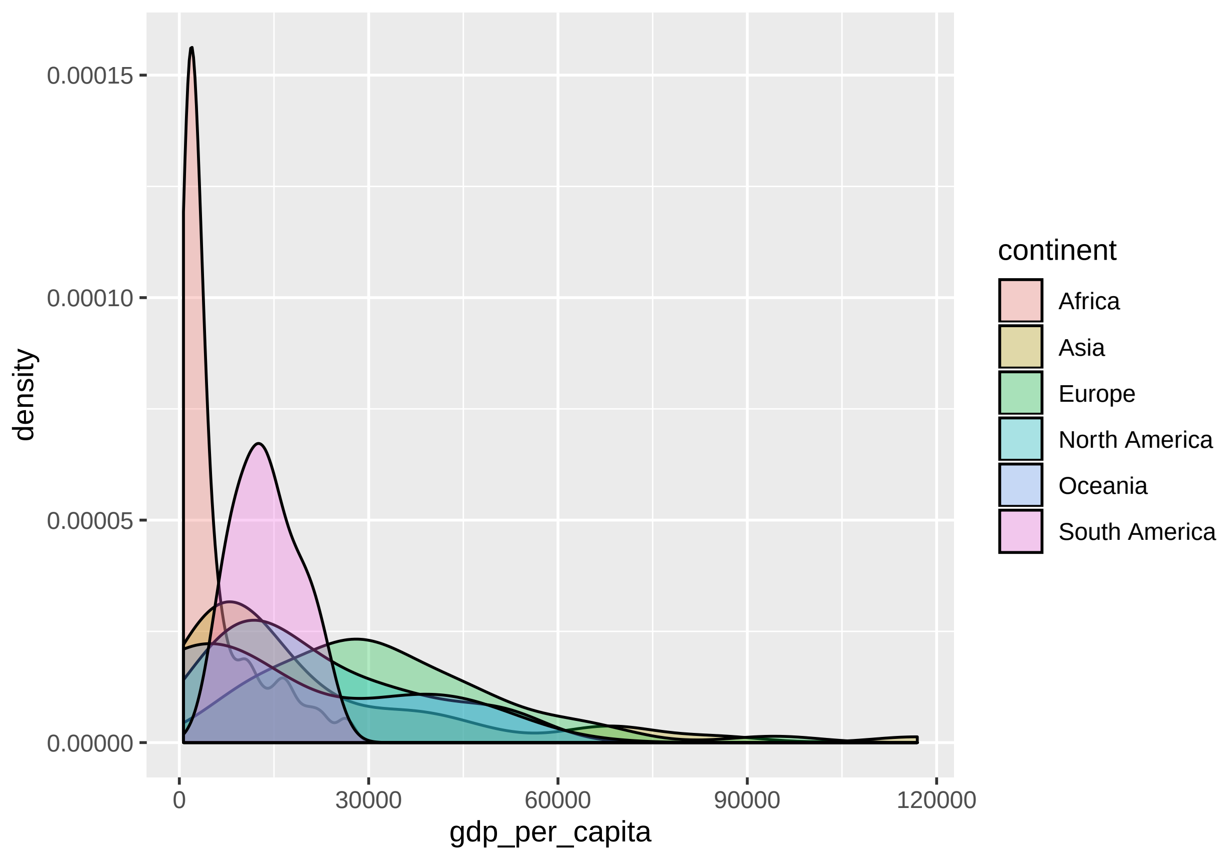
```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```



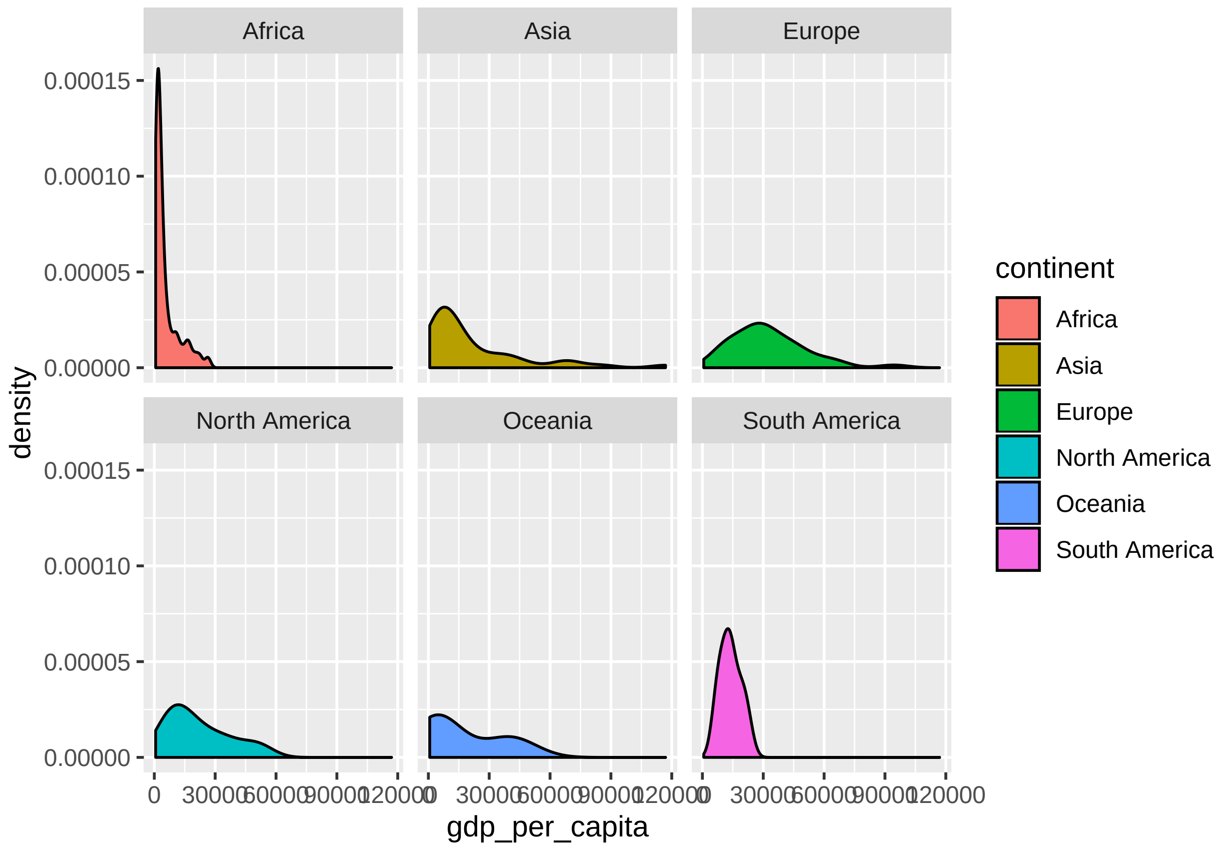
```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = continent, y = gdp_per_capita) +
    geom_boxplot()
```



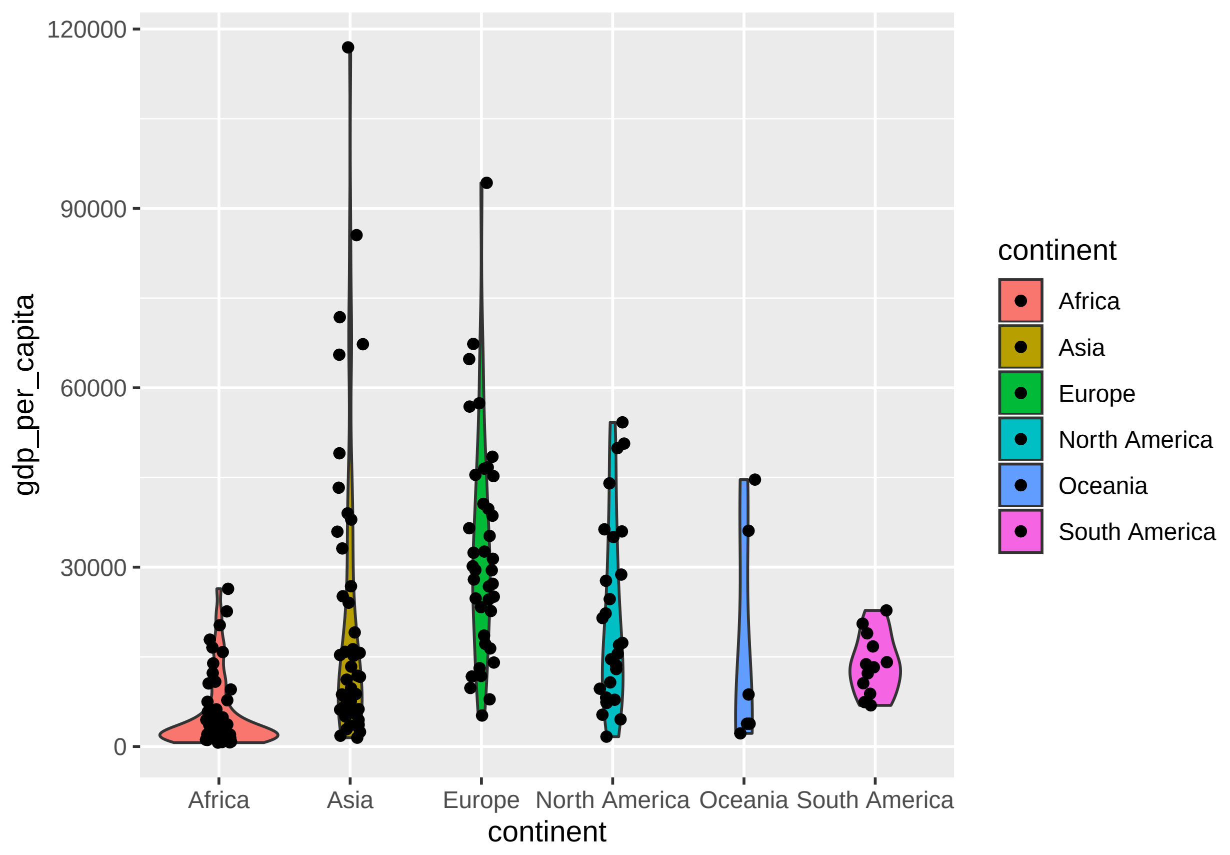
```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita, fill = continent) +
    geom_density(alpha = 0.3)
```



```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = gdp_per_capita, fill = continent) +
    geom_density()+
  facet_wrap(~continent)
```



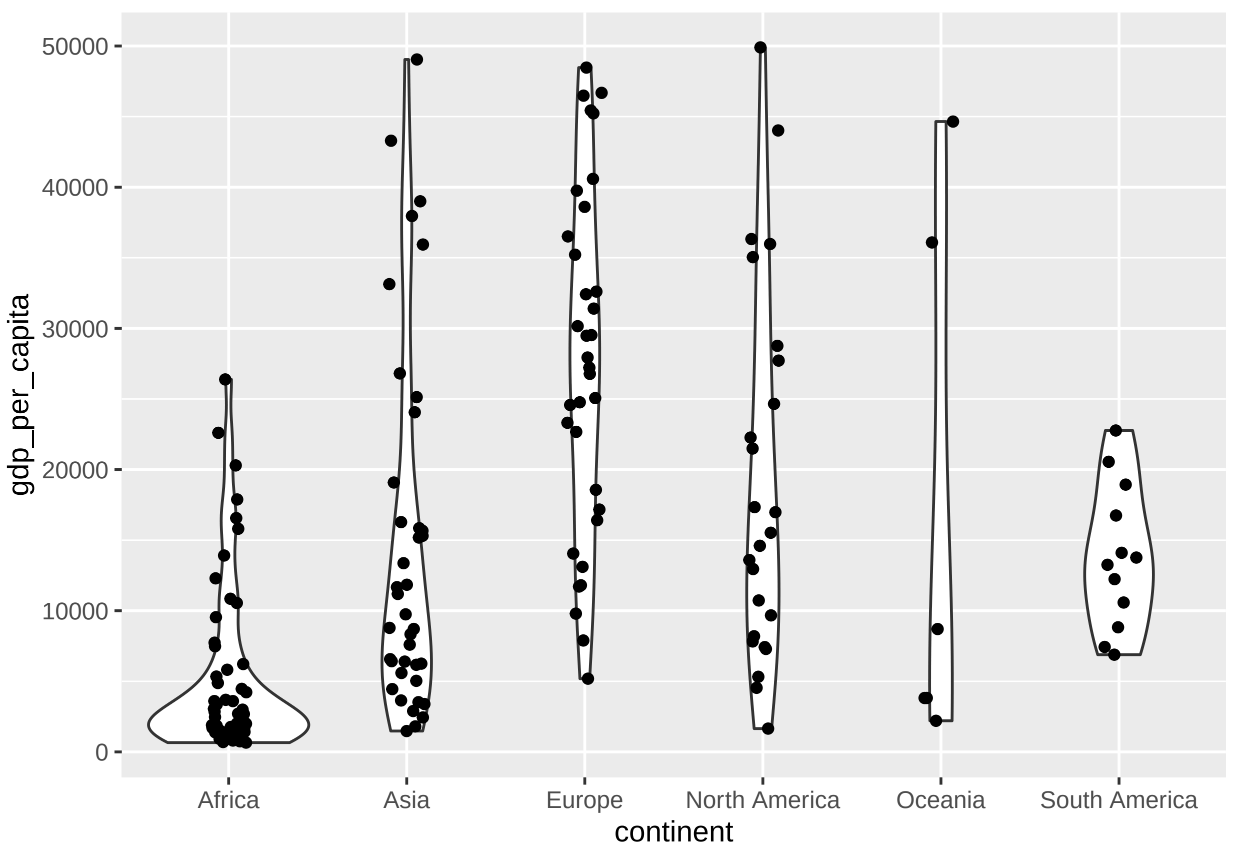
```r
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = continent, y = gdp_per_capita, fill = continent) +
    geom_violin() +
    geom_jitter(width = 0.1)
```



The plots add to the picture previously seen in the descriptive table. It seems that even though the average gdp in Asia is not as low as in Africa, this is partly due to a few **extreme cases** that probably have a high influence on the mean.

Due to this realization we might want to trim the cases with very high gdp values to get a more realistic picture about the descriptive statistics of each continent. We can insert the **filter()** function int hte previous code chuncks to do this trimming.

```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  filter(gdp_per_capita < 50000) %>%
    ggplot() +
      aes(x = continent, y = gdp_per_capita) +
      geom_violin() +
      geom_jitter(width = 0.1)
```



```
COVID_data_latest %>%
  select(continent, gdp_per_capita) %>%
  drop_na() %>%
  filter(gdp_per_capita < 50000) %>%
    group_by(continent) %>%
      summarize(mean = mean(gdp_per_capita),
                sd = sd(gdp_per_capita))
```

```
## # A tibble: 6 x 3
##   continent       mean     sd
##   <fct>          <dbl>  <dbl>
## 1 Africa         5444.  6183.
## 2 Asia          14591. 12710.
## 3 Europe        27546. 12213.
## 4 North America 19192. 13095.
## 5 Oceania       16548. 18775.
## 6 South America 13841.  5110.
```

We can take more valiralbes into account while plotting by using the facet_wrap() and facet_grid() functions.

________________*Practice*________________

- Use the above learned techniques to explore the relationship of **total_cases_per_million** and **gdp_per_capita_kat** in the COVID_data_latest dataset.

____

Exploring the relationship of two numerical variables

We ususally use the **correlation** coefficient to discribe the relationship between **two continuous variables**.

The **cor()** function is can be used to get the correlation coefficient. (note that we need to use the **drop_na()** function again to drop cases with missing data, otherwise the cor() function will return on NAs).

```
COVID_data_latest %>%
  select(new_cases_per_million, gdp_per_capita) %>%
  drop_na() %>%
      cor()
```

```
##                       new_cases_per_million gdp_per_capita
## new_cases_per_million              1.000000       0.342772
## gdp_per_capita                     0.342772       1.000000
```

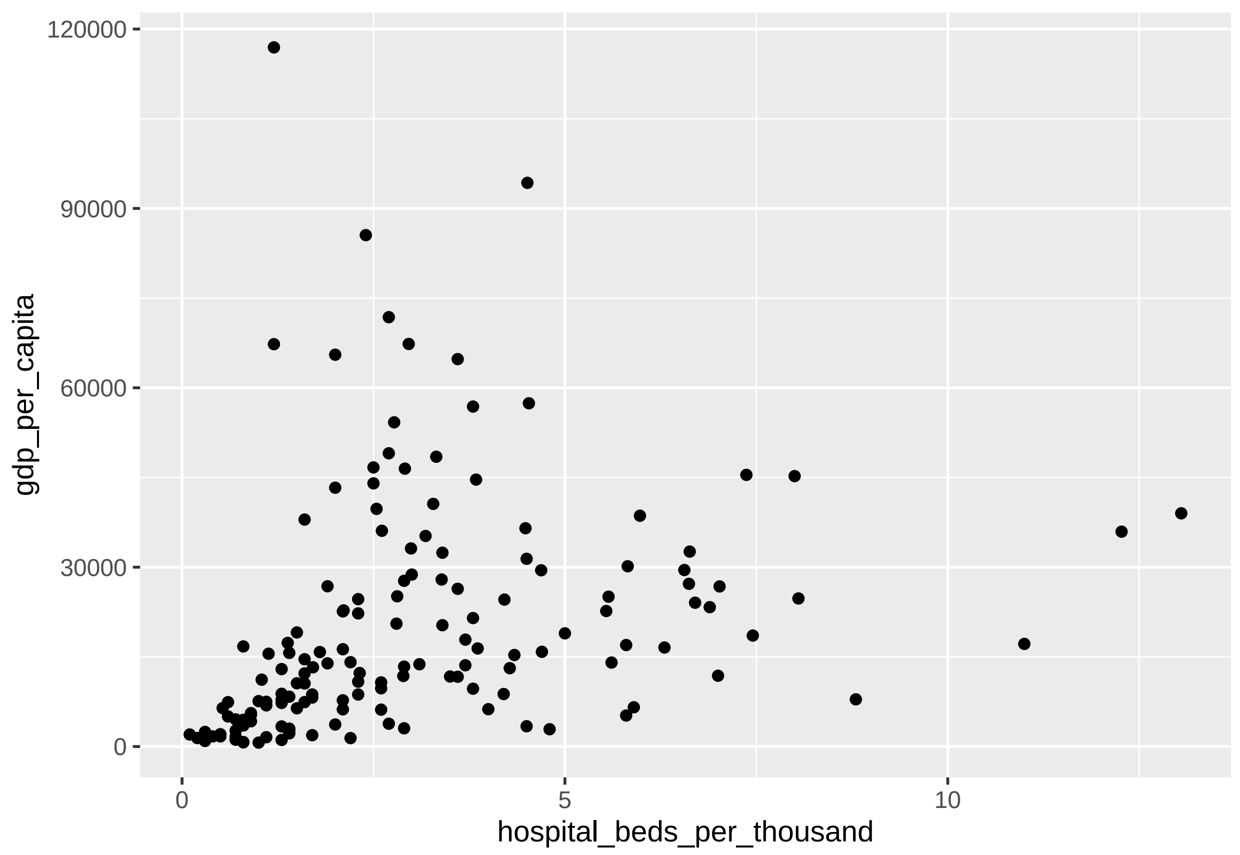
```
COVID_data_latest %>%
  select(new_cases_per_million, gdp_per_capita, hospital_beds_per_thousand) %>%
  drop_na() %>%
      cor()
```

```
##                            new_cases_per_million gdp_per_capita
## new_cases_per_million                  1.0000000      0.3308897
## gdp_per_capita                         0.3308897      1.0000000
## hospital_beds_per_thousand             0.3138501      0.2985625
##                            hospital_beds_per_thousand
## new_cases_per_million                       0.3138501
## gdp_per_capita                              0.2985625
## hospital_beds_per_thousand                  1.0000000
```

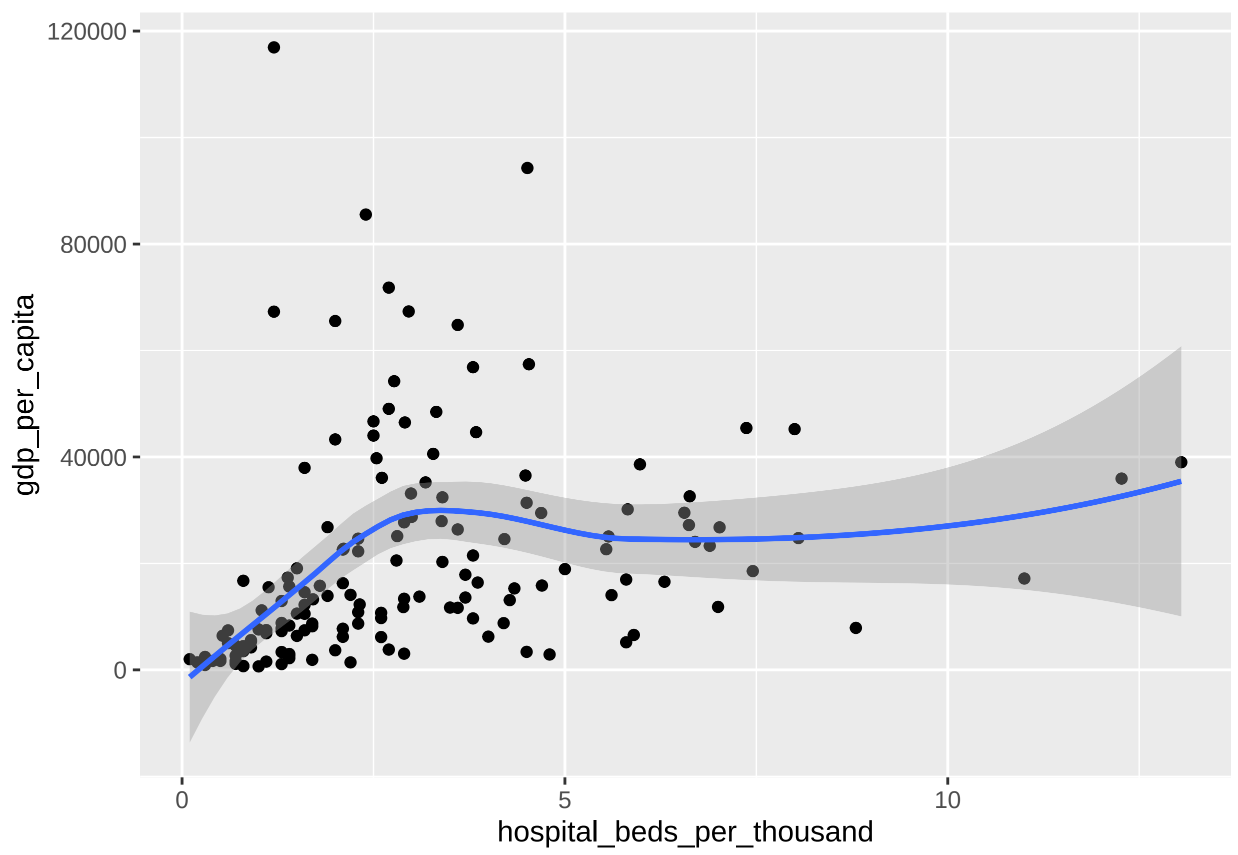
We usually use scatterplots (**geom_point()**) to **visualize** relationship between two continuous variables.

The **geom_smooth()** layer can also provide additional insight about the underlying relationship between the two variables. When using geom_smooth, the blue line represents the "trend line" or "regression line", while the grey area shows the confidence interval for this prediction line.

```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita) +
    geom_point()
```



```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita) +
    geom_point() +
    geom_smooth()
```



_________________*Practice*_________________

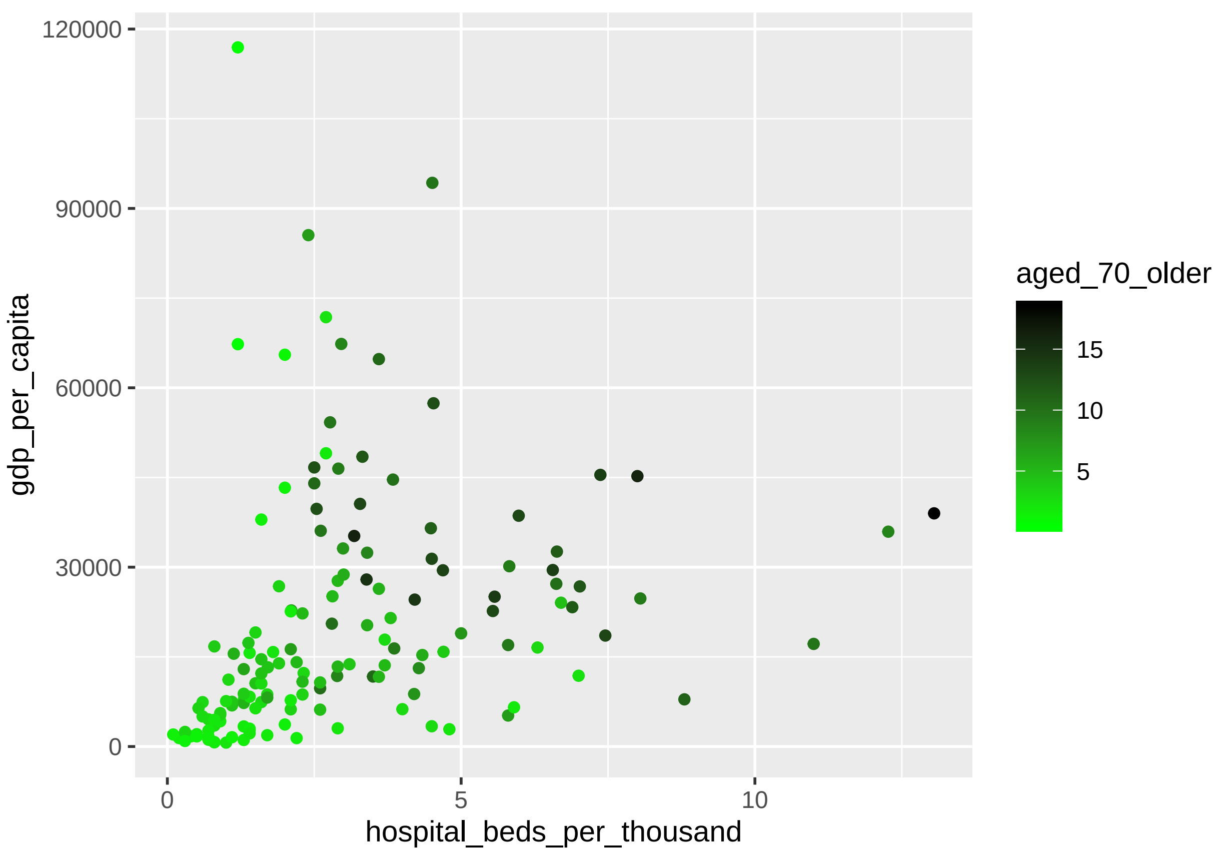
Explore the relationship between aged_70_older and gdp_per_capita in the COVID_data_latest.

- determine the correlation coefficient
- visualize the relationship

____

When exploring the relationship of three continuous variables we can use graded color as a new aesthetic for the third variable, just like in the example below:

```
COVID_data_latest %>%
  select(hospital_beds_per_thousand, gdp_per_capita, aged_70_older) %>%
  drop_na() %>%
  ggplot() +
    aes(x = hospital_beds_per_thousand, y = gdp_per_capita, col = aged_70_older) +
    geom_point()+
  scale_colour_gradientn(colours=c("green","black"))
```



120000
90000
60000
30000
0
gdp_per_capita
0
5
10
hospital_beds_per_thousand
aged_70_older
15
10
5