Project 2

Heleine Fouda

2023-10-15

Setting up the environment

First Data Set: Gapminder

The data used in this analysis are from the Gapminder package. The main data frame gapminder has 1704 rows and 6 variables:country factor with 142 levels; continent factor with 5 levels; year ranges from 1952 to 2007 in increments of 5 years. For each of 142 countries, the package provides values for life expectancy, GDP per capita, and population, every five years, from 1952 to 2007. The research questions: how many unique countries does the data contain, by continent? which country has the highest life Expectancy and which one has the lowest? Also, by continent, which country experienced the sharpest 5-year drop in life expectancy and what was the drop? Finally, is the difference in median life expectancy between continents due to chance or is it real?

```
# Loading the data
library(gapminder)

# Importing the data
gapminder <- read_csv("https://raw.githubusercontent.com/Heleinef/Data-Science-Master_Heleine/main/gapm
```

```
## New names:
## Rows: 1704 Columns: 7
## -- Column specification
## ---------------------------------------- Delimiter: "," chr
## (2): country, continent dbl (5): ...1, year, lifeExp, pop,
## gdpPercap
## i Use `spec()` to retrieve the full column specification for
## this data. i Specify the column types or set `show_col_types =
## FALSE` to quiet this message.
## * `` -> `...1`
```

```
gapminder
```

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

```
## 10    10 Afghanistan Asia       1997    41.8 22227415      635.
## # i 1,694 more rows
```

Data exploration

Let's take a peek at the data

```
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 7
## $ ...1      <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14~
## $ country   <chr> "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <chr> "Asia", "Asia", "Asia", "Asia", "Asia", "Asia~
## $ year      <dbl> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 198~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.43~
## $ pop       <dbl> 8425333, 9240934, 10267083, 11537966, 1307946~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9~
```

```
head(gapminder)
```

```
## # A tibble: 6 x 7
##    ...1 country     continent  year lifeExp      pop gdpPercap
##   <dbl> <chr>       <chr>     <dbl>   <dbl>    <dbl>     <dbl>
## 1     1 Afghanistan Asia       1952    28.8  8425333      779.
## 2     2 Afghanistan Asia       1957    30.3  9240934      821.
## 3     3 Afghanistan Asia       1962    32.0 10267083      853.
## 4     4 Afghanistan Asia       1967    34.0 11537966      836.
## 5     5 Afghanistan Asia       1972    36.1 13079460      740.
## 6     6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
str(gapminder)
```

```
## spc_tbl_ [1,704 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1     : num [1:1704] 1 2 3 4 5 6 7 8 9 10 ...
##  $ country  : chr [1:1704] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr [1:1704] "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : num [1:1704] 1952 1957 1962 1967 1972 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : num [1:1704] 8425333 9240934 10267083 11537966 13079460 ...
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   country = col_character(),
##   ..   continent = col_character(),
##   ..   year = col_double(),
##   ..   lifeExp = col_double(),
##   ..   pop = col_double(),
##   ..   gdpPercap = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

How many unique countries does the data contain, by continent?

```
gapminder|>
  group_by(continent)|>
```

```
  summarize(n_obs = n(),
            n_countries = n_distinct(country))
```

```
## # A tibble: 5 x 3
##   continent n_obs n_countries
##   <chr>     <int>       <int>
## 1 Africa      624          52
## 2 Americas    300          25
## 3 Asia        396          33
## 4 Europe      360          30
## 5 Oceania      24           2
```

```
# Label: Summary - gapminder
summary(gapminder)
```

```
##       ...1          country           continent
##  Min.   :   1.0   Length:1704        Length:1704
##  1st Qu.: 426.8   Class :character   Class :character
##  Median : 852.5   Mode  :character   Mode  :character
##  Mean   : 852.5
##  3rd Qu.:1278.2
##  Max.   :1704.0
##       year         lifeExp           pop
##  Min.   :1952   Min.   :23.60   Min.   :6.001e+04
##  1st Qu.:1966   1st Qu.:48.20   1st Qu.:2.794e+06
##  Median :1980   Median :60.71   Median :7.024e+06
##  Mean   :1980   Mean   :59.47   Mean   :2.960e+07
##  3rd Qu.:1993   3rd Qu.:70.85   3rd Qu.:1.959e+07
##  Max.   :2007   Max.   :82.60   Max.   :1.319e+09
##    gdpPercap
##  Min.   :   241.2
##  1st Qu.:  1202.1
##  Median :  3531.8
##  Mean   :  7215.3
##  3rd Qu.:  9325.5
##  Max.   :113523.1
```

```
# Label : Standard deviation of gdpPercap
sd(gapminder$gdpPercap)
```

```
## [1] 9857.455
```

```
# Label: variance - gdpPercap
var(gapminder$gdpPercap)
```

```
## [1] 97169410
```

```
# Label : Standard deviation of lifeExp
sd(gapminder$lifeExp)
```

```
## [1] 12.91711
```

```
# label :  variance - lifeExp
var(gapminder$lifeExp)
```

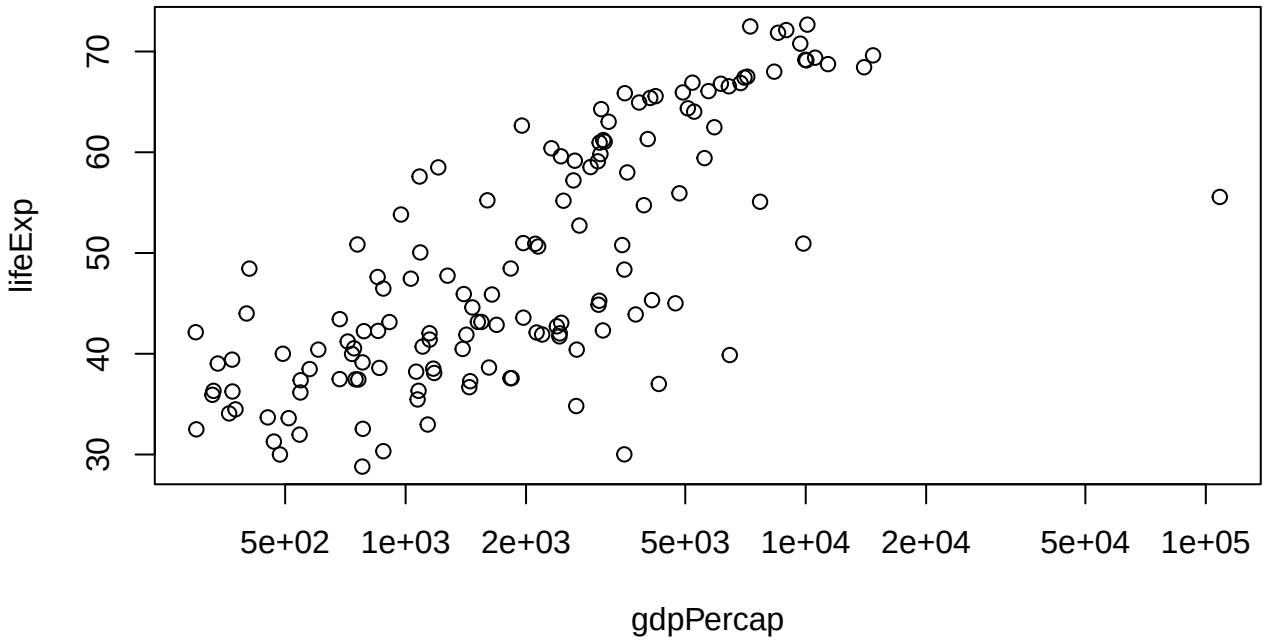
```
## [1] 166.8517
```

```r
# label aggregate life expectancy
aggregate(lifeExp ~ continent, gapminder, median)
```

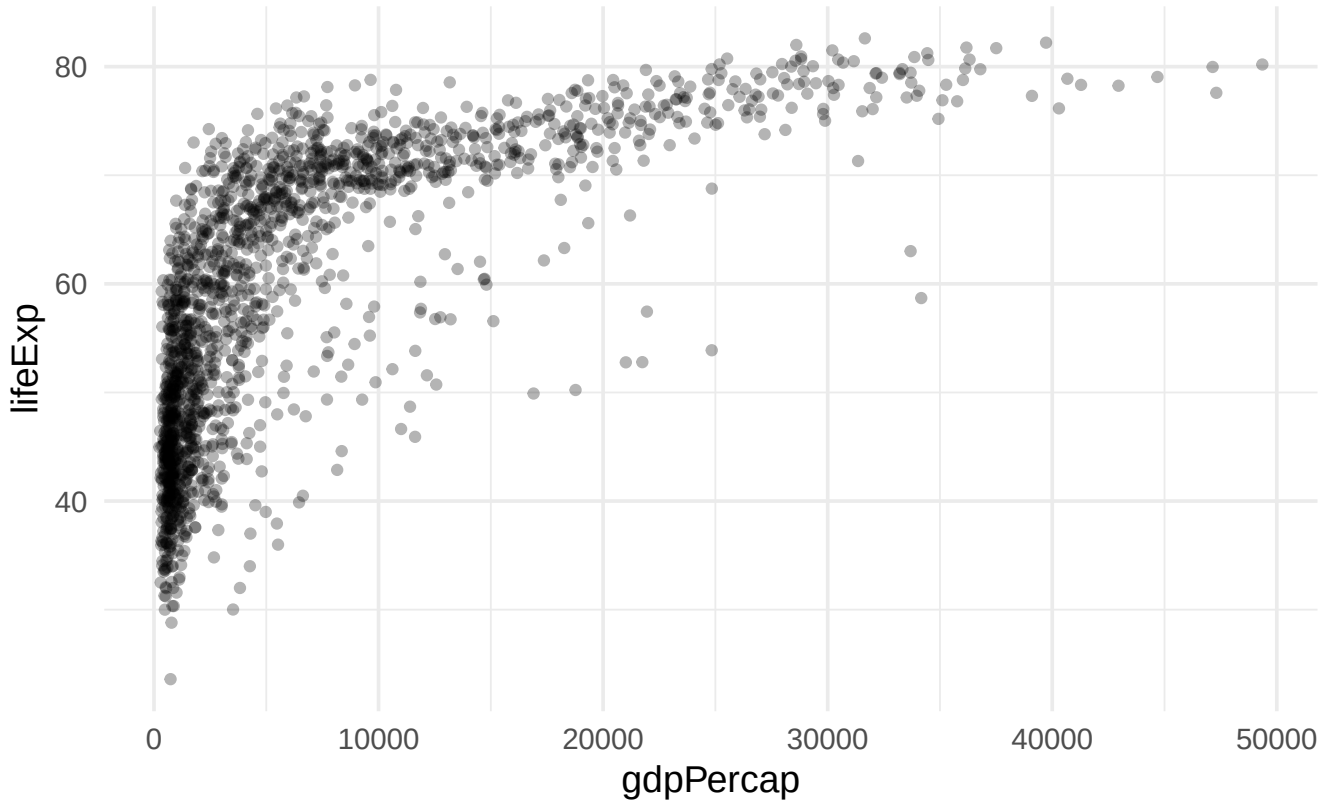
```
##   continent lifeExp
## 1    Africa 47.7920
## 2  Americas 67.0480
## 3      Asia 61.7915
## 4    Europe 72.2410
## 5   Oceania 73.6650
```

```r
plot(lifeExp ~ gdpPercap, gapminder, subset = year == 2007, log = "x")
```

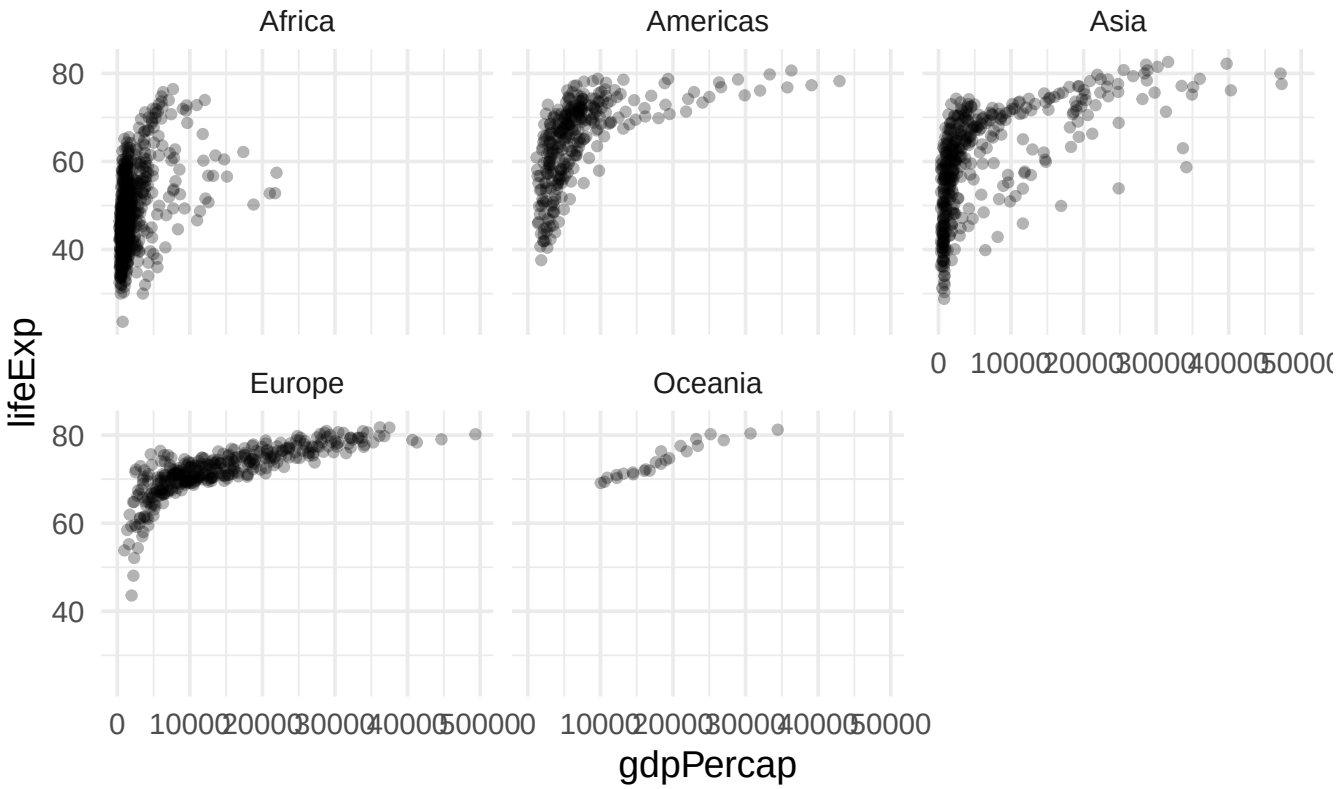
```r
plot(lifeExp ~ gdpPercap, gapminder, subset = year == 1952, log = "x")
```




```r
# plot, gdpPercap < 50000
library(ggplot2)
gapminder|>
  filter(gdpPercap <50000) |>
  ggplot(aes(x = gdpPercap, y = lifeExp))+
  geom_point(alpha =0.3)
```




```
# Label : Facet-wrap, plot- gdpPercap < 50000
library(ggplot2)
gapminder|>
  filter(gdpPercap <50000) |>
  ggplot(aes(x = gdpPercap, y = lifeExp))+
  geom_point(alpha =0.3) +
facet_wrap(~continent)
```



Let's examine the correlation between life expectancy and GDP per capita

```
# label: Correlation test
correlation <-cor(gapminder$lifeExp,gapminder$gdpPercap)
correlation
```

```
## [1] 0.5837062
```

Data transformation

Let's transform the gapminder long format into a large format:

```
# Let's first assign the data to a new object called "data"
data <- select (gapminder,country, year, lifeExp)
data
```

```
## # A tibble: 1,704 x 3
##    country      year lifeExp
##    <chr>       <dbl>   <dbl>
##  1 Afghanistan  1952    28.8
##  2 Afghanistan  1957    30.3
##  3 Afghanistan  1962    32.0
##  4 Afghanistan  1967    34.0
##  5 Afghanistan  1972    36.1
##  6 Afghanistan  1977    38.4
##  7 Afghanistan  1982    39.9
##  8 Afghanistan  1987    40.8
##  9 Afghanistan  1992    41.7
## 10 Afghanistan  1997    41.8
## # i 1,694 more rows
```

Let's transform the long gapminder format into a wide format

```
# From a long format into a wide format
gapminder_wider <- data|>
  pivot_wider(names_from = year,
              values_from = lifeExp)
gapminder_wider
```

```
## # A tibble: 142 x 13
##    country     `1952` `1957` `1962` `1967` `1972` `1977` `1982`
##    <chr>        <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 Afghanistan   28.8   30.3   32.0   34.0   36.1   38.4   39.9
##  2 Albania       55.2   59.3   64.8   66.2   67.7   68.9   70.4
##  3 Algeria       43.1   45.7   48.3   51.4   54.5   58.0   61.4
##  4 Angola        30.0   32.0   34     36.0   37.9   39.5   39.9
##  5 Argentina     62.5   64.4   65.1   65.6   67.1   68.5   69.9
##  6 Australia     69.1   70.3   70.9   71.1   71.9   73.5   74.7
##  7 Austria       66.8   67.5   69.5   70.1   70.6   72.2   73.2
##  8 Bahrain       50.9   53.8   56.9   59.9   63.3   65.6   69.1
##  9 Bangladesh    37.5   39.3   41.2   43.5   45.3   46.9   50.0
## 10 Belgium       68     69.2   70.2   70.9   71.4   72.8   73.9
## # i 132 more rows
## # i 5 more variables: `1987` <dbl>, `1992` <dbl>, `1997` <dbl>,
## #   `2002` <dbl>, `2007` <dbl>
```

Let's get back to the longer format for analysis convenience Pivot_longer

```
# From a wide format into a long format
gapminder_long <- gapminder_wider|>
  pivot_longer(2:13,
               names_to = "year",
               values_to = "lifeExp")

gapminder_long
```

```
## # A tibble: 1,704 x 3
##    country     year  lifeExp
##    <chr>       <chr>   <dbl>
##  1 Afghanistan 1952     28.8
##  2 Afghanistan 1957     30.3
##  3 Afghanistan 1962     32.0
##  4 Afghanistan 1967     34.0
##  5 Afghanistan 1972     36.1
##  6 Afghanistan 1977     38.4
##  7 Afghanistan 1982     39.9
##  8 Afghanistan 1987     40.8
##  9 Afghanistan 1992     41.7
## 10 Afghanistan 1997     41.8
## # i 1,694 more rows
```

Data analysis (using the functions filter, select, mutate, arrange and group_by)

Highest life expectancy found in 2007 Japan

```
#  Label: Tidying the data using the select, arrange & mutate functions
gapminder|>
  select(country, gdpPercap, lifeExp)|>
  filter(country == "Africa") |>
```

```
  mutate(Life_Expectancy = lifeExp, GDP_per_capita =gdpPercap)
```

```
## # A tibble: 0 x 5
## # i 5 variables: country <chr>, gdpPercap <dbl>, lifeExp <dbl>,
## #   Life_Expectancy <dbl>, GDP_per_capita <dbl>
```

```
 arrange(gapminder,desc(lifeExp))
```

```
## # A tibble: 1,704 x 7
##     ...1 country         continent  year lifeExp    pop gdpPercap
##    <dbl> <chr>           <chr>     <dbl>   <dbl>  <dbl>     <dbl>
##  1   804 Japan           Asia       2007    82.6 1.27e8    31656.
##  2   672 Hong Kong, Chi~ Asia       2007    82.2 6.98e6    39725.
##  3   803 Japan           Asia       2002    82   1.27e8    28605.
##  4   696 Iceland         Europe     2007    81.8 3.02e5    36181.
##  5  1488 Switzerland     Europe     2007    81.7 7.55e6    37506.
##  6   671 Hong Kong, Chi~ Asia       2002    81.5 6.76e6    30209.
##  7    72 Australia       Oceania    2007    81.2 2.04e7    34435.
##  8  1428 Spain           Europe     2007    80.9 4.04e7    28821.
##  9  1476 Sweden          Europe     2007    80.9 9.03e6    33860.
## 10   768 Israel          Asia       2007    80.7 6.43e6    25523.
## # i 1,694 more rows
```

Lowest life expectancy found in 1992 Rwanda

```
# Label: country with the lowest life expectancy
gapminder|>
  select(country, gdpPercap, lifeExp)|>
  filter(country == "Africa") |>
  mutate(Life_Expectancy = lifeExp, GDP_per_capita =gdpPercap)
```

```
## # A tibble: 0 x 5
## # i 5 variables: country <chr>, gdpPercap <dbl>, lifeExp <dbl>,
## #   Life_Expectancy <dbl>, GDP_per_capita <dbl>
```

```
 arrange(gapminder,(lifeExp))
```

```
## # A tibble: 1,704 x 7
##     ...1 country      continent  year lifeExp     pop gdpPercap
##    <dbl> <chr>        <chr>     <dbl>   <dbl>   <dbl>     <dbl>
##  1  1293 Rwanda       Africa     1992    23.6 7290203      737.
##  2     1 Afghanistan  Asia       1952    28.8 8425333      779.
##  3   553 Gambia       Africa     1952    30    284320      485.
##  4    37 Angola       Africa     1952    30.0 4232095     3521.
##  5  1345 Sierra Leone Africa     1952    30.3 2143249      880.
##  6     2 Afghanistan  Asia       1957    30.3 9240934      821.
##  7   222 Cambodia     Asia       1977    31.2 6978607      525.
##  8  1033 Mozambique   Africa     1952    31.3 6446316      469.
##  9  1346 Sierra Leone Africa     1957    31.6 2295678     1004.
## 10   193 Burkina Faso Africa     1952    32.0 4469979      543.
## # i 1,694 more rows
```

Median life expectancy across continents

```
# Label: Median life expectancy across continents in 2007
gapminder|>
  filter(year == 2007) |>
```

```
  group_by(continent) |>
  summarise(lifeExp = median(lifeExp))
```

```
## # A tibble: 5 x 2
##   continent lifeExp
##   <chr>       <dbl>
## 1 Africa       52.9
## 2 Americas     72.9
## 3 Asia         72.4
## 4 Europe       78.6
## 5 Oceania      80.7
```

Which country experienced the sharpest 5-year drop in life expectancy?

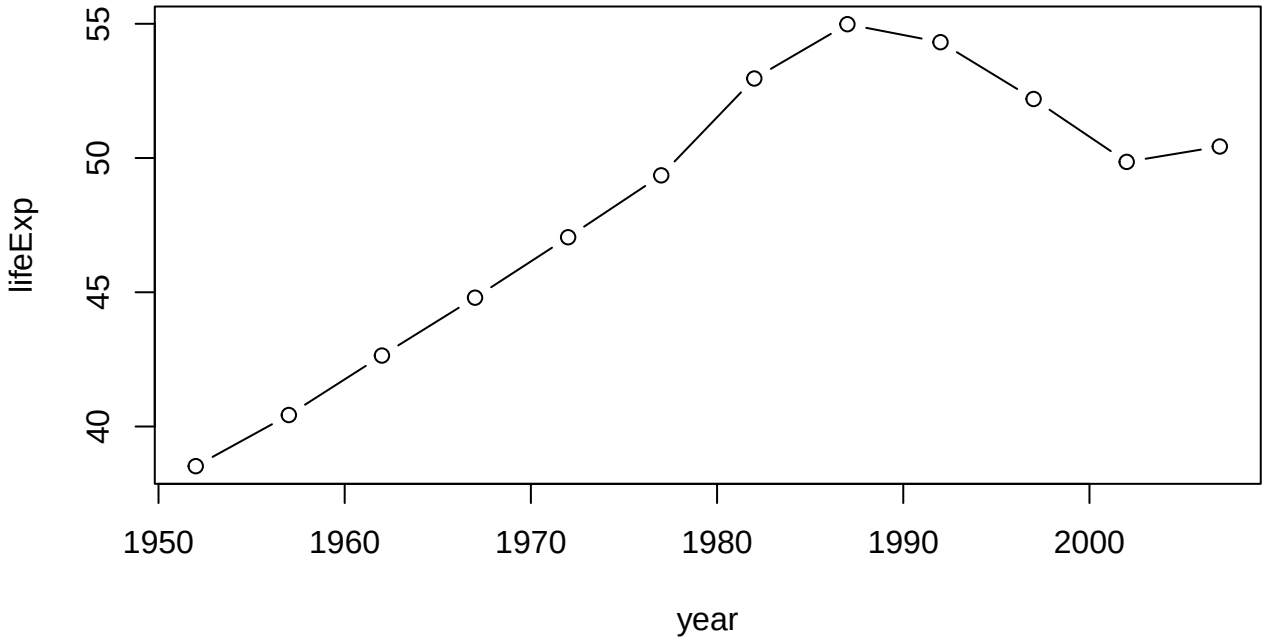
```
# label: Country with the sharpest drop in life expectancy
gapminder|>
  group_by(continent, country)|>
  select(country, year, continent, lifeExp)|>
  mutate(le_delta = lifeExp - lag(lifeExp)) |>
  summarize(worst_le_delta = min(le_delta, na.rm = TRUE))|>
  filter(min_rank(worst_le_delta) < 2)|> arrange(worst_le_delta)
```

```
## `summarise()` has grouped output by 'continent'. You can
## override using the `.groups` argument.
```

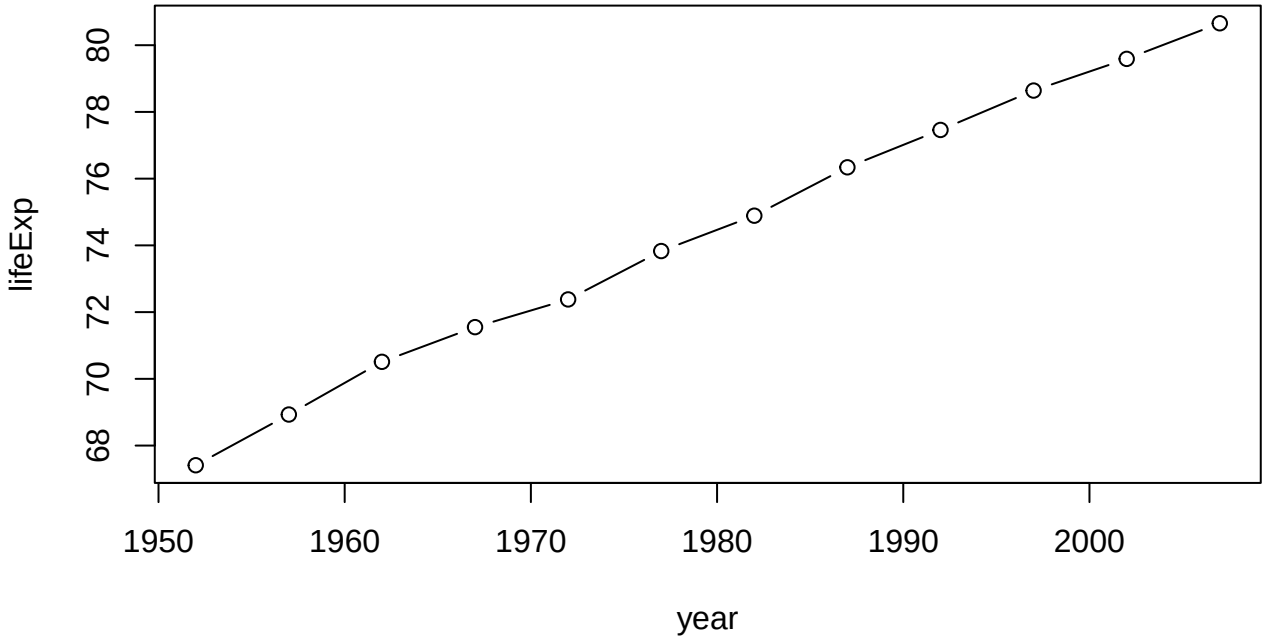
```
## # A tibble: 5 x 3
## # Groups:   continent [5]
##   continent country     worst_le_delta
##   <chr>     <chr>                <dbl>
## 1 Africa    Rwanda             -20.4
## 2 Asia      Cambodia            -9.10
## 3 Americas  El Salvador         -1.51
## 4 Europe    Montenegro          -1.46
## 5 Oceania   Australia            0.170
```

Let's get more specific and compare the median life expectancy in Cameroon, Central Africa and in France (Western Europe)

```
# label: Evolution of life Expectancy in Cameroon
plot(lifeExp ~ year, gapminder, subset = country == "Cameroon", type = "b")
```




```
# label: Evolution of life Expectancy in France
plot(lifeExp ~ year, gapminder, subset = country == "France", type = "b")
```




```
# Let's first create a smaller gapminder dataset made of only Cameroon and France

gapminder_small <-gapminder|>
  filter (country== "Cameroon" | country == "France")
gapminder_small
```

```
## # A tibble: 24 x 7
##     ...1 country  continent  year lifeExp      pop gdpPercap
##    <dbl> <chr>    <chr>     <dbl>   <dbl>    <dbl>     <dbl>
##  1   229 Cameroon Africa     1952    38.5  5009067     1173.
```

```
##  2   230 Cameroon Africa     1957    40.4  5359923     1313.
##  3   231 Cameroon Africa     1962    42.6  5793633     1400.
##  4   232 Cameroon Africa     1967    44.8  6335506     1508.
##  5   233 Cameroon Africa     1972    47.0  7021028     1684.
##  6   234 Cameroon Africa     1977    49.4  7959865     1783.
##  7   235 Cameroon Africa     1982    53.0  9250831     2368.
##  8   236 Cameroon Africa     1987    55.0 10780667     2603.
##  9   237 Cameroon Africa     1992    54.3 12467171     1793.
## 10   238 Cameroon Africa     1997    52.2 14195809     1694.
## # i 14 more rows
```

Pivot-wide

```
# Let pivot-wide gapminder_small
gapminder_wider2 <- gapminder_small|>
  pivot_wider(names_from = year,
              values_from = lifeExp)
gapminder_wider2
```

```
## # A tibble: 24 x 17
##     ...1 country  continent    pop gdpPercap `1952` `1957` `1962`
##    <dbl> <chr>    <chr>      <dbl>     <dbl>  <dbl>  <dbl>  <dbl>
##  1   229 Cameroon Africa    5.01e6     1173.   38.5   NA     NA
##  2   230 Cameroon Africa    5.36e6     1313.   NA     40.4   NA
##  3   231 Cameroon Africa    5.79e6     1400.   NA     NA     42.6
##  4   232 Cameroon Africa    6.34e6     1508.   NA     NA     NA
##  5   233 Cameroon Africa    7.02e6     1684.   NA     NA     NA
##  6   234 Cameroon Africa    7.96e6     1783.   NA     NA     NA
##  7   235 Cameroon Africa    9.25e6     2368.   NA     NA     NA
##  8   236 Cameroon Africa    1.08e7     2603.   NA     NA     NA
##  9   237 Cameroon Africa    1.25e7     1793.   NA     NA     NA
## 10   238 Cameroon Africa    1.42e7     1694.   NA     NA     NA
## # i 14 more rows
## # i 9 more variables: `1967` <dbl>, `1972` <dbl>, `1977` <dbl>,
## #   `1982` <dbl>, `1987` <dbl>, `1992` <dbl>, `1997` <dbl>,
## #   `2002` <dbl>, `2007` <dbl>
```

Median life expectancy in Cameroon & France

```
# Label: Median life expectancy in Cameroon & France

gapminder|>
  select(country, lifeExp)|>
  filter (country == "Cameroon"|country == "France")|>
  group_by( country)|>
  summarise(Median_lifeExp = median (lifeExp))
```

```
## # A tibble: 2 x 2
##   country  Median_lifeExp
##   <chr>             <dbl>
## 1 Cameroon           49.6
## 2 France             74.4
```

Since we've observed a difference between the median life expectancy in Cameroon and in France. Let's check (using a T- test),if that difference is due to chance.

```r
# Let's create a new data frame named df1
df1 <- gapminder|>
  select(country, lifeExp)|>
  filter(country == "Cameroon" | country == "France")

# Let's conduct a t-test on df1
t.test(data = df1, lifeExp~country)
```

```
##
##  Welch Two Sample t-test
##
## data:  lifeExp by country
## t = -13.052, df = 20.851, p-value = 1.685e-11
## alternative hypothesis: true difference in means between group Cameroon and group France is not equal
## 95 percent confidence interval:
##  -30.40003 -22.04080
## sample estimates:
## mean in group Cameroon   mean in group France
##               48.12850               74.34892
```

The Key Findings

Our analysis reveals the following findings:

1. There is a strong and positive correlation between life expectancy and GDP per capita. The correlation test estimates its value at 0.5837062.
2. There is a difference in life expectancy between continents;
3. The observed difference in life expectancy between Cameroon (mean = 48.12850, median = 49.6055) and France (mean = 74.34892, median = 74.3600) is true and statistically significant. The difference is confirmed by a two tails t-test that reveals within a 95 percent confidence interval that the difference in life expectancy between Cameroon & France countries ranges from -30.40003 to -22.04080.
4. Evolution in life expectancy in France from 1952 to 2007 has followed a steady and positive upward linear trend. In Cameroon, the one also observes an upward trend but with a sharp increase in life expectancy during the 1990s. It would be interesting to find out what caused that sharp increase.
5. The highest life expectancy was found in Japan in 2007 at 82.60300 and the lowest life expectancy was found in 1992 Rwanda at 23.59900 . Rwanda is also the country that experienced the sharpest 5-year drop in life expectancy at -20.421

Second Data Set : Hotels Bookings

The research questions: This analysis attempts to evaluate and to visualize the average daily rate (ADR), booking trends and patterns at two different hotels (one resort, one city hotel) from 2015 to 2017. The data used in this analysis are from an open hotel booking demand data by Antonio, Almeida and Nunes, 2017

```r
# Importing the data
hotels <-read_csv("https://raw.githubusercontent.com/Heleinef/Data-Science-Master_Heleine/main/Hotel_Bo
hotels
```

```
## # A tibble: 119,390 x 36
##    hotel        is_canceled lead_time arrival_date_year
##    <chr>              <dbl>     <dbl>             <dbl>
##  1 Resort Hotel           0       342              2015
##  2 Resort Hotel           0       737              2015
```

```
##  3 Resort Hotel           0         7              2015
##  4 Resort Hotel           0        13              2015
##  5 Resort Hotel           0        14              2015
##  6 Resort Hotel           0        14              2015
##  7 Resort Hotel           0         0              2015
##  8 Resort Hotel           0         9              2015
##  9 Resort Hotel           1        85              2015
## 10 Resort Hotel           1        75              2015
## # i 119,380 more rows
## # i 32 more variables: arrival_date_month <chr>,
## #   arrival_date_week_number <dbl>,
## #   arrival_date_day_of_month <dbl>,
## #   stays_in_weekend_nights <dbl>, stays_in_week_nights <dbl>,
## #   adults <dbl>, children <dbl>, babies <dbl>, meal <chr>,
## #   country <chr>, market_segment <chr>, ...
```

Data exploration

Let's take a peek at the data

```
glimpse(hotels)
```

```
## Rows: 119,390
## Columns: 36
## $ hotel                         <chr> "Resort Hotel", "Resort ~
## $ is_canceled                   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ lead_time                     <dbl> 342, 737, 7, 13, 14, 14,~
## $ arrival_date_year             <dbl> 2015, 2015, 2015, 2015, ~
## $ arrival_date_month            <chr> "July", "July", "July", ~
## $ arrival_date_week_number      <dbl> 27, 27, 27, 27, 27, 27, ~
## $ arrival_date_day_of_month     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ stays_in_weekend_nights       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ stays_in_week_nights          <dbl> 0, 0, 1, 1, 2, 2, 2, 2, ~
## $ adults                        <dbl> 2, 2, 1, 1, 2, 2, 2, 2, ~
## $ children                      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ babies                        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ meal                          <chr> "BB", "BB", "BB", "BB", ~
## $ country                       <chr> "PRT", "PRT", "GBR", "GB~
## $ market_segment                <chr> "Direct", "Direct", "Dir~
## $ distribution_channel          <chr> "Direct", "Direct", "Dir~
## $ is_repeated_guest             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ previous_cancellations        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ previous_bookings_not_canceled <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ reserved_room_type            <chr> "C", "C", "A", "A", "A",~
## $ assigned_room_type            <chr> "C", "C", "C", "A", "A",~
## $ booking_changes               <dbl> 3, 4, 0, 0, 0, 0, 0, 0, ~
## $ deposit_type                  <chr> "No Deposit", "No Deposi~
## $ agent                         <dbl> NA, NA, NA, 304, 240, 24~
## $ company                       <dbl> NA, NA, NA, NA, NA, NA, ~
## $ days_in_waiting_list          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ customer_type                 <chr> "Transient", "Transient"~
## $ adr                           <dbl> 0.00, 0.00, 75.00, 75.00~
## $ required_car_parking_spaces   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ total_of_special_requests     <dbl> 0, 0, 0, 0, 1, 1, 0, 1, ~
## $ reservation_status            <chr> "Check-Out", "Check-Out"~
```

```
## $ reservation_status_date        <chr> "7/1/15", "7/1/15", "7/2~
## $ name                           <chr> "Ernest Barnes", "Andrea~
## $ email                          <chr> "Ernest.Barnes31@outlook~
## $ `phone-number`                 <chr> "669-792-1661", "858-637~
## $ credit_card                    <chr> "************4322", "***~
```

```
head(hotels)
```

```
## # A tibble: 6 x 36
##   hotel        is_canceled lead_time arrival_date_year
##   <chr>              <dbl>     <dbl>             <dbl>
## 1 Resort Hotel           0       342              2015
## 2 Resort Hotel           0       737              2015
## 3 Resort Hotel           0         7              2015
## 4 Resort Hotel           0        13              2015
## 5 Resort Hotel           0        14              2015
## 6 Resort Hotel           0        14              2015
## # i 32 more variables: arrival_date_month <chr>,
## #   arrival_date_week_number <dbl>,
## #   arrival_date_day_of_month <dbl>,
## #   stays_in_weekend_nights <dbl>, stays_in_week_nights <dbl>,
## #   adults <dbl>, children <dbl>, babies <dbl>, meal <chr>,
## #   country <chr>, market_segment <chr>,
## #   distribution_channel <chr>, is_repeated_guest <dbl>, ...
```

```
str(hotels)
```

```
## spc_tbl_ [119,390 x 36] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ hotel                         : chr [1:119390] "Resort Hotel" "Resort Hotel" "Resort Hotel" "Reso
##  $ is_canceled                   : num [1:119390] 0 0 0 0 0 0 0 0 1 1 ...
##  $ lead_time                     : num [1:119390] 342 737 7 13 14 14 0 9 85 75 ...
##  $ arrival_date_year             : num [1:119390] 2015 2015 2015 2015 2015 ...
##  $ arrival_date_month            : chr [1:119390] "July" "July" "July" "July" ...
##  $ arrival_date_week_number      : num [1:119390] 27 27 27 27 27 27 27 27 27 27 ...
##  $ arrival_date_day_of_month     : num [1:119390] 1 1 1 1 1 1 1 1 1 1 ...
##  $ stays_in_weekend_nights       : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ stays_in_week_nights          : num [1:119390] 0 0 1 1 2 2 2 2 3 3 ...
##  $ adults                        : num [1:119390] 2 2 1 1 2 2 2 2 2 2 ...
##  $ children                      : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ babies                        : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ meal                          : chr [1:119390] "BB" "BB" "BB" "BB" ...
##  $ country                       : chr [1:119390] "PRT" "PRT" "GBR" "GBR" ...
##  $ market_segment                : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
##  $ distribution_channel          : chr [1:119390] "Direct" "Direct" "Direct" "Corporate" ...
##  $ is_repeated_guest             : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_cancellations        : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ previous_bookings_not_canceled: num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ reserved_room_type            : chr [1:119390] "C" "C" "A" "A" ...
##  $ assigned_room_type            : chr [1:119390] "C" "C" "C" "A" ...
##  $ booking_changes               : num [1:119390] 3 4 0 0 0 0 0 0 0 0 ...
##  $ deposit_type                  : chr [1:119390] "No Deposit" "No Deposit" "No Deposit" "No Deposit
##  $ agent                         : num [1:119390] NA NA NA 304 240 240 NA 303 240 15 ...
##  $ company                       : num [1:119390] NA NA NA NA NA NA NA NA NA NA ...
##  $ days_in_waiting_list          : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ customer_type                 : chr [1:119390] "Transient" "Transient" "Transient" "Transient" ..
##  $ adr                           : num [1:119390] 0 0 75 75 98 ...
```

```
##  $ required_car_parking_spaces   : num [1:119390] 0 0 0 0 0 0 0 0 0 0 ...
##  $ total_of_special_requests     : num [1:119390] 0 0 0 0 1 1 0 1 1 0 ...
##  $ reservation_status            : chr [1:119390] "Check-Out" "Check-Out" "Check-Out" "Check-Out" ..
##  $ reservation_status_date       : chr [1:119390] "7/1/15" "7/1/15" "7/2/15" "7/2/15" ...
##  $ name                          : chr [1:119390] "Ernest Barnes" "Andrea Baker" "Rebecca Parker" "La
##  $ email                         : chr [1:119390] "Ernest.Barnes31@outlook.com" "Andrea_Baker94@aol.
##  $ phone-number                  : chr [1:119390] "669-792-1661" "858-637-6955" "652-885-2745" "364-
##  $ credit_card                   : chr [1:119390] "************4322" "************9157" "**********
##  - attr(*, "spec")=
##   .. cols(
##   ..   hotel = col_character(),
##   ..   is_canceled = col_double(),
##   ..   lead_time = col_double(),
##   ..   arrival_date_year = col_double(),
##   ..   arrival_date_month = col_character(),
##   ..   arrival_date_week_number = col_double(),
##   ..   arrival_date_day_of_month = col_double(),
##   ..   stays_in_weekend_nights = col_double(),
##   ..   stays_in_week_nights = col_double(),
##   ..   adults = col_double(),
##   ..   children = col_double(),
##   ..   babies = col_double(),
##   ..   meal = col_character(),
##   ..   country = col_character(),
##   ..   market_segment = col_character(),
##   ..   distribution_channel = col_character(),
##   ..   is_repeated_guest = col_double(),
##   ..   previous_cancellations = col_double(),
##   ..   previous_bookings_not_canceled = col_double(),
##   ..   reserved_room_type = col_character(),
##   ..   assigned_room_type = col_character(),
##   ..   booking_changes = col_double(),
##   ..   deposit_type = col_character(),
##   ..   agent = col_double(),
##   ..   company = col_double(),
##   ..   days_in_waiting_list = col_double(),
##   ..   customer_type = col_character(),
##   ..   adr = col_double(),
##   ..   required_car_parking_spaces = col_double(),
##   ..   total_of_special_requests = col_double(),
##   ..   reservation_status = col_character(),
##   ..   reservation_status_date = col_character(),
##   ..   name = col_character(),
##   ..   email = col_character(),
##   ..   `phone-number` = col_character(),
##   ..   credit_card = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(hotels)
```

```
##     hotel            is_canceled       lead_time
##  Length:119390      Min.   :0.0000   Min.   :  0
##  Class :character   1st Qu.:0.0000   1st Qu.: 18
##  Mode  :character   Median :0.0000   Median : 69
```

```
##                    Mean   :0.3704   Mean   :104
##                    3rd Qu.:1.0000   3rd Qu.:160
##                    Max.   :1.0000   Max.   :737
##
##  arrival_date_year arrival_date_month arrival_date_week_number
##  Min.   :2015      Length:119390      Min.   : 1.00
##  1st Qu.:2016      Class :character   1st Qu.:16.00
##  Median :2016      Mode  :character   Median :28.00
##  Mean   :2016                         Mean   :27.17
##  3rd Qu.:2017                         3rd Qu.:38.00
##  Max.   :2017                         Max.   :53.00
##
##  arrival_date_day_of_month stays_in_weekend_nights
##  Min.   : 1.0              Min.   : 0.0000
##  1st Qu.: 8.0              1st Qu.: 0.0000
##  Median :16.0              Median : 1.0000
##  Mean   :15.8              Mean   : 0.9276
##  3rd Qu.:23.0              3rd Qu.: 2.0000
##  Max.   :31.0              Max.   :19.0000
##
##  stays_in_week_nights     adults          children
##  Min.   : 0.0         Min.   : 0.000   Min.   : 0.0000
##  1st Qu.: 1.0         1st Qu.: 2.000   1st Qu.: 0.0000
##  Median : 2.0         Median : 2.000   Median : 0.0000
##  Mean   : 2.5         Mean   : 1.856   Mean   : 0.1039
##  3rd Qu.: 3.0         3rd Qu.: 2.000   3rd Qu.: 0.0000
##  Max.   :50.0         Max.   :55.000   Max.   :10.0000
##                                        NA's   :4
##      babies              meal             country
##  Min.   : 0.000000   Length:119390      Length:119390
##  1st Qu.: 0.000000   Class :character   Class :character
##  Median : 0.000000   Mode  :character   Mode  :character
##  Mean   : 0.007949
##  3rd Qu.: 0.000000
##  Max.   :10.000000
##
##  market_segment     distribution_channel is_repeated_guest
##  Length:119390      Length:119390        Min.   :0.00000
##  Class :character   Class :character     1st Qu.:0.00000
##  Mode  :character   Mode  :character     Median :0.00000
##                                          Mean   :0.03191
##                                          3rd Qu.:0.00000
##                                          Max.   :1.00000
##
##  previous_cancellations previous_bookings_not_canceled
##  Min.   : 0.00000       Min.   : 0.0000
##  1st Qu.: 0.00000       1st Qu.: 0.0000
##  Median : 0.00000       Median : 0.0000
##  Mean   : 0.08712       Mean   : 0.1371
##  3rd Qu.: 0.00000       3rd Qu.: 0.0000
##  Max.   :26.00000       Max.   :72.0000
##
##  reserved_room_type assigned_room_type booking_changes
##  Length:119390      Length:119390      Min.   : 0.0000
```

```
##  Class :character   Class :character   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median : 0.0000
##                                        Mean   : 0.2211
##                                        3rd Qu.: 0.0000
##                                        Max.   :21.0000
##
##  deposit_type           agent           company
##  Length:119390      Min.   :  1.00   Min.   :  6.0
##  Class :character   1st Qu.:  9.00   1st Qu.: 62.0
##  Mode  :character   Median : 14.00   Median :179.0
##                     Mean   : 86.69   Mean   :189.3
##                     3rd Qu.:229.00   3rd Qu.:270.0
##                     Max.   :535.00   Max.   :543.0
##                     NA's   :16340    NA's   :112593
##  days_in_waiting_list customer_type           adr
##  Min.   :  0.000      Length:119390      Min.   :  -6.38
##  1st Qu.:  0.000      Class :character   1st Qu.:  69.29
##  Median :  0.000      Mode  :character   Median :  94.58
##  Mean   :  2.321                         Mean   : 101.83
##  3rd Qu.:  0.000                         3rd Qu.: 126.00
##  Max.   :391.000                         Max.   :5400.00
##
##  required_car_parking_spaces total_of_special_requests
##  Min.   :0.00000             Min.   :0.0000
##  1st Qu.:0.00000             1st Qu.:0.0000
##  Median :0.00000             Median :0.0000
##  Mean   :0.06252             Mean   :0.5714
##  3rd Qu.:0.00000             3rd Qu.:1.0000
##  Max.   :8.00000             Max.   :5.0000
##
##  reservation_status reservation_status_date     name
##  Length:119390      Length:119390           Length:119390
##  Class :character   Class :character        Class :character
##  Mode  :character   Mode  :character        Mode  :character
##
##
##
##
##     email           phone-number       credit_card
##  Length:119390      Length:119390      Length:119390
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

Data transformation :

Let's select our variables of interest in this analysis

```
# Selecting our variables of interest
hotels_small <- hotels|>
  select(hotel, adr, stays_in_week_nights,
stays_in_weekend_nights, adults, children,
```

```
arrival_date_month, arrival_date_year, is_canceled)
hotels_small
```

```
## # A tibble: 119,390 x 9
##    hotel   adr stays_in_week_nights stays_in_weekend_nig~1 adults
##    <chr> <dbl>                <dbl>                  <dbl>  <dbl>
##  1 Reso~    0                    0                      0      2
##  2 Reso~    0                    0                      0      2
##  3 Reso~   75                    1                      0      1
##  4 Reso~   75                    1                      0      1
##  5 Reso~   98                    2                      0      2
##  6 Reso~   98                    2                      0      2
##  7 Reso~  107                    2                      0      2
##  8 Reso~  103                    2                      0      2
##  9 Reso~   82                    3                      0      2
## 10 Reso~  106.                   3                      0      2
## # i 119,380 more rows
## # i abbreviated name: 1: stays_in_weekend_nights
## # i 4 more variables: children <dbl>, arrival_date_month <chr>,
## #   arrival_date_year <dbl>, is_canceled <dbl>
```

Let's make a pivot longer out of our selected variables for easy analysis

```
# Label: pivot -longer
hotels_long <- hotels_small|>
  pivot_longer(
    cols = c(adr, stays_in_week_nights, stays_in_weekend_nights, adults, children),
    names_to = "variable",
    values_to = "value"
  )
hotels_long
```

```
## # A tibble: 596,950 x 6
##    hotel        arrival_date_month arrival_date_year is_canceled
##    <chr>        <chr>                          <dbl>       <dbl>
##  1 Resort Hotel July                           2015           0
##  2 Resort Hotel July                           2015           0
##  3 Resort Hotel July                           2015           0
##  4 Resort Hotel July                           2015           0
##  5 Resort Hotel July                           2015           0
##  6 Resort Hotel July                           2015           0
##  7 Resort Hotel July                           2015           0
##  8 Resort Hotel July                           2015           0
##  9 Resort Hotel July                           2015           0
## 10 Resort Hotel July                           2015           0
## # i 596,940 more rows
## # i 2 more variables: variable <chr>, value <dbl>
```

Data analysis

Now, let's calculate the summary statistics for stays_in_week_nights and stays_in_weekend_nights Mean

```
# First, let's find the mean
hotels |>
  summarise(across(.cols = starts_with("stays"),  mean))
```

```
## # A tibble: 1 x 2
##   stays_in_weekend_nights stays_in_week_nights
##                     <dbl>                <dbl>
## 1                   0.928                 2.50
```

Median

```
# Now, let's find the median
hotels |>
  summarise(across(.cols = starts_with("stays"),  median))
```

```
## # A tibble: 1 x 2
##   stays_in_weekend_nights stays_in_week_nights
##                     <dbl>                <dbl>
## 1                       1                    2
```

Mean and standard deviation

```
# Calculating the both the mean and the standard deviation
hotels |>
  summarise(across(.cols = starts_with("stays"),  list(mean, sd)))
```

```
## # A tibble: 1 x 4
##   stays_in_weekend_nights_1 stays_in_weekend_nights_2
##                       <dbl>                     <dbl>
## 1                     0.928                     0.999
## # i 2 more variables: stays_in_week_nights_1 <dbl>,
## #   stays_in_week_nights_2 <dbl>
```

Let's calculate the total number of guests for each booking

```
# label: Number of guests per booking
hotels |>
  select(adults, children, babies) |>
  rowwise() |>
  mutate(guests = sum(c(adults, children, babies))) |>
  filter(adults > 0, children > 0, babies > 0)
```

```
## # A tibble: 172 x 4
## # Rowwise:
##    adults children babies guests
##     <dbl>    <dbl>  <dbl>  <dbl>
##  1      2        1      1      4
##  2      2        1      1      4
##  3      2        1      1      4
##  4      2        1      1      4
##  5      2        1      1      4
##  6      2        1      1      4
##  7      2        1      1      4
##  8      2        2      1      5
##  9      2        2      1      5
## 10      1        2      1      4
## # i 162 more rows
```

Let's calculate the average daily rate at both hotels

```
# The average daily rate
 Average_Price <- mean(hotels$adr)
Average_Price
```

```
## [1] 101.8311
```

```
# Most expensive months
hotels_small |>
  filter(hotel == "City Hotel" | hotel == "Resort Hotel") |>
group_by(arrival_date_month)|>
  mutate("Average_Price" = mean(adr))|>
  arrange(desc(arrival_date_month))
```

```
## # A tibble: 119,390 x 10
## # Groups:   arrival_date_month [12]
##    hotel   adr stays_in_week_nights stays_in_weekend_nig~1 adults
##    <chr> <dbl>                <dbl>                   <dbl>  <dbl>
##  1 Reso~  123                    2                       0      2
##  2 Reso~   98                    2                       0      2
##  3 Reso~  151                    3                       0      2
##  4 Reso~  135.                   3                       0      2
##  5 Reso~  153                    3                       0      2
##  6 Reso~  138.                   3                       0      2
##  7 Reso~  111                    4                       0      2
##  8 Reso~  123                    4                       0      2
##  9 Reso~  119                    4                       0      2
## 10 Reso~  155                    4                       0      2
## # i 119,380 more rows
## # i abbreviated name: 1: stays_in_weekend_nights
## # i 5 more variables: children <dbl>, arrival_date_month <chr>,
## #   arrival_date_year <dbl>, is_canceled <dbl>,
## #   Average_Price <dbl>
```

```
hotels_small
```

```
## # A tibble: 119,390 x 9
##    hotel   adr stays_in_week_nights stays_in_weekend_nig~1 adults
##    <chr> <dbl>                <dbl>                   <dbl>  <dbl>
##  1 Reso~    0                    0                       0      2
##  2 Reso~    0                    0                       0      2
##  3 Reso~   75                    1                       0      1
##  4 Reso~   75                    1                       0      1
##  5 Reso~   98                    2                       0      2
##  6 Reso~   98                    2                       0      2
##  7 Reso~  107                    2                       0      2
##  8 Reso~  103                    2                       0      2
##  9 Reso~   82                    3                       0      2
## 10 Reso~  106.                   3                       0      2
## # i 119,380 more rows
## # i abbreviated name: 1: stays_in_weekend_nights
## # i 4 more variables: children <dbl>, arrival_date_month <chr>,
## #   arrival_date_year <dbl>, is_canceled <dbl>
```

Let's find out which hotel has charged its clients the highest price

```
# Highest daily daily rate charged
hotels_small |>
  filter(hotel == "City Hotel" | hotel == "Resort Hotel") |>
group_by(hotel)|>
  mutate("Average_Price" = mean(adr))|>
  arrange(desc(adr))
```

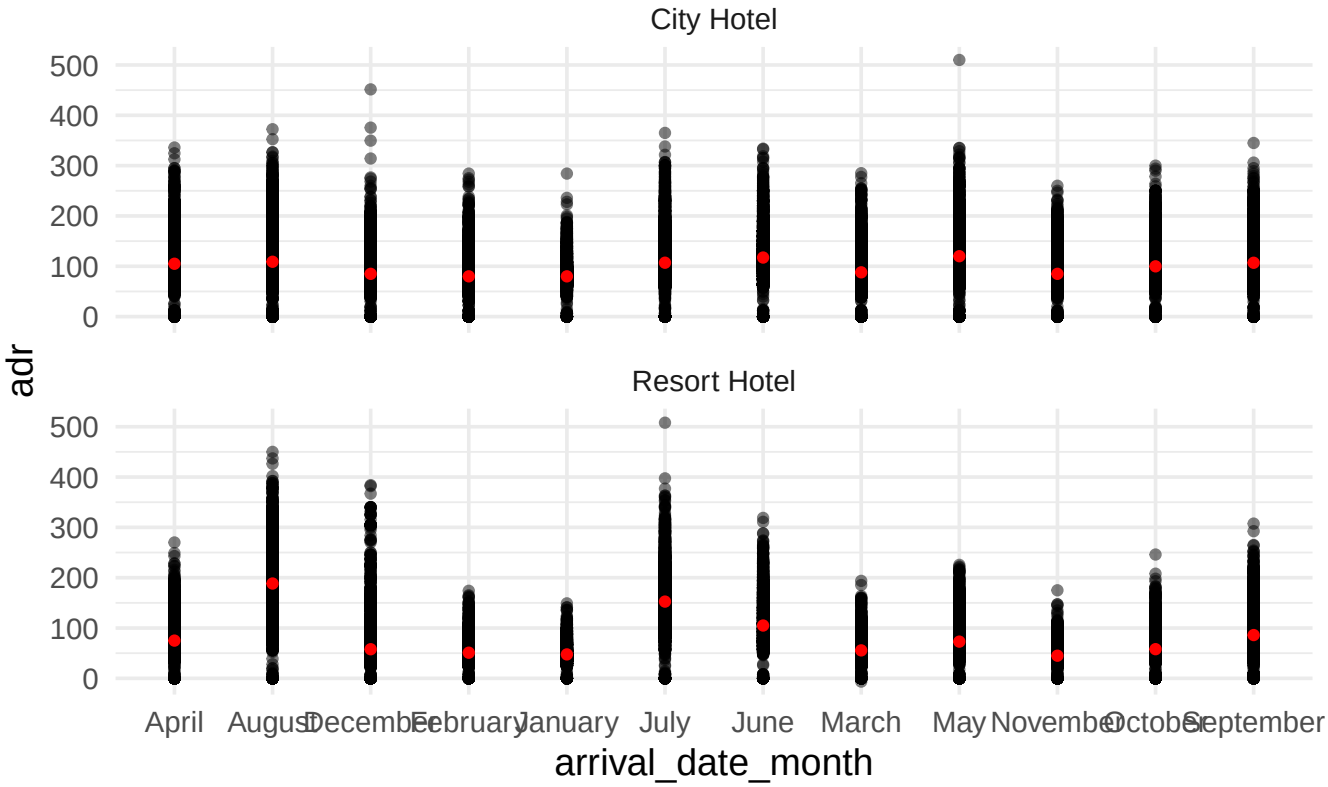
```
## # A tibble: 119,390 x 10
## # Groups:   hotel [2]
##    hotel   adr stays_in_week_nights stays_in_weekend_nig~1 adults
##    <chr> <dbl>                <dbl>                  <dbl>  <dbl>
##  1 City~ 5400                   1                      0      2
##  2 City~  510                   1                      0      1
##  3 Reso~  508                   1                      0      2
##  4 City~  452.                  1                      1      2
##  5 Reso~  450                  10                      4      2
##  6 Reso~  437                   4                      2      2
##  7 Reso~  426.                  6                      2      2
##  8 Reso~  402                   3                      2      3
##  9 Reso~  397.                  5                      3      3
## 10 Reso~  392                   8                      2      2
## # i 119,380 more rows
## # i abbreviated name: 1: stays_in_weekend_nights
## # i 5 more variables: children <dbl>, arrival_date_month <chr>,
## #   arrival_date_year <dbl>, is_canceled <dbl>,
## #   Average_Price <dbl>
```

```
hotels_small
```

```
## # A tibble: 119,390 x 9
##    hotel   adr stays_in_week_nights stays_in_weekend_nig~1 adults
##    <chr> <dbl>                <dbl>                  <dbl>  <dbl>
##  1 Reso~    0                    0                      0      2
##  2 Reso~    0                    0                      0      2
##  3 Reso~   75                    1                      0      1
##  4 Reso~   75                    1                      0      1
##  5 Reso~   98                    2                      0      2
##  6 Reso~   98                    2                      0      2
##  7 Reso~  107                    2                      0      2
##  8 Reso~  103                    2                      0      2
##  9 Reso~   82                    3                      0      2
## 10 Reso~  106.                   3                      0      2
## # i 119,380 more rows
## # i abbreviated name: 1: stays_in_weekend_nights
## # i 4 more variables: children <dbl>, arrival_date_month <chr>,
## #   arrival_date_year <dbl>, is_canceled <dbl>
```

Data visualization

```
hotels |>
  filter(adr < 4000) |>
  ggplot(aes(x = arrival_date_month, y = adr)) +
  geom_point(alpha = 0.5) +
  geom_point(
    stat = "summary", fun = "median",
    colour = "red") +
  facet_wrap(~ hotel, ncol = 1)
```



Let's calculate the summary statistics of bookings that were canceled at both hotels between 2015 and 2017

```
# All cancelled bookings from 2015 to 2017 at the two hotels
hotels |>
  group_by(hotel, is_canceled) |>
  summarise(
    across(.cols = starts_with("stays"),  list(mean = mean, sd = sd), .names = "{.fn}_{.col}")
    )
```

```
## `summarise()` has grouped output by 'hotel'. You can override
## using the `.groups` argument.
```

```
## # A tibble: 4 x 6
## # Groups:   hotel [2]
##   hotel is_canceled mean_stays_in_weeken~1 sd_stays_in_weekend_~2
##   <chr>       <dbl>                  <dbl>                  <dbl>
## 1 City~           0                  0.801                  0.862
## 2 City~           1                  0.788                  0.917
## 3 Reso~           0                  1.13                   1.14
## 4 Reso~           1                  1.34                   1.14
## # i abbreviated names: 1: mean_stays_in_weekend_nights,
## #   2: sd_stays_in_weekend_nights
## # i 2 more variables: mean_stays_in_week_nights <dbl>,
## #   sd_stays_in_week_nights <dbl>
```

```
hotels_summary <- hotels |>
  group_by(hotel, is_canceled) |>
  summarise(
    across(
      .cols = starts_with("stays"),
      list(mean = mean),
```

```
      .names = "{.fn}_{.col}"
    ),
    .groups = "drop"
  )

hotels_summary
```

```
## # A tibble: 4 x 4
##   hotel is_canceled mean_stays_in_weeken~1 mean_stays_in_week_n~2
##   <chr>       <dbl>                  <dbl>                  <dbl>
## 1 City~           0                  0.801                   2.12
## 2 City~           1                  0.788                   2.27
## 3 Reso~           0                  1.13                    3.01
## 4 Reso~           1                  1.34                    3.44
## # i abbreviated names: 1: mean_stays_in_weekend_nights,
## #   2: mean_stays_in_week_nights
```

Let's now break down the above summary statistics by year:

Bookings cancelled in 2015

```
cancelled_2015 <- hotels |>
  filter(arrival_date_year == 2015)|>
  group_by(hotel)|>
  summarise(Cancelled = is_canceled)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was
## deprecated in dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember
##   that `reframe()` always returns an ungrouped data frame and
##   adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this
## warning was generated.
```

```
## `summarise()` has grouped output by 'hotel'. You can override
## using the `.groups` argument.
```

```
cancelled_2015
```

```
## # A tibble: 21,996 x 2
## # Groups:   hotel [2]
##    hotel      Cancelled
##    <chr>          <dbl>
##  1 City Hotel         0
##  2 City Hotel         1
##  3 City Hotel         1
##  4 City Hotel         1
##  5 City Hotel         1
##  6 City Hotel         1
##  7 City Hotel         0
##  8 City Hotel         1
##  9 City Hotel         1
## 10 City Hotel         1
## # i 21,986 more rows
```

```
distinct(cancelled_2015)
```

```
## # A tibble: 4 x 2
## # Groups:   hotel [2]
##   hotel        Cancelled
##   <chr>            <dbl>
## 1 City Hotel           0
## 2 City Hotel           1
## 3 Resort Hotel         0
## 4 Resort Hotel         1
```

Bookings cancelled in 2016

```
cancelled_2016 <- hotels |>
  filter(arrival_date_year == 2016)|>
  group_by(hotel)|>
  summarise(Cancelled = is_canceled)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was
## deprecated in dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember
##   that `reframe()` always returns an ungrouped data frame and
##   adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this
## warning was generated.
```

```
## `summarise()` has grouped output by 'hotel'. You can override
## using the `.groups` argument.
```

```
cancelled_2016
```

```
## # A tibble: 56,707 x 2
## # Groups:   hotel [2]
##    hotel      Cancelled
##    <chr>          <dbl>
##  1 City Hotel         1
##  2 City Hotel         0
##  3 City Hotel         0
##  4 City Hotel         0
##  5 City Hotel         0
##  6 City Hotel         0
##  7 City Hotel         0
##  8 City Hotel         0
##  9 City Hotel         0
## 10 City Hotel         0
## # i 56,697 more rows
```

```
distinct(cancelled_2016)
```

```
## # A tibble: 4 x 2
## # Groups:   hotel [2]
##   hotel        Cancelled
##   <chr>            <dbl>
## 1 City Hotel           1
## 2 City Hotel           0
## 3 Resort Hotel         0
```

```
## 4 Resort Hotel         1
```

Bookings cancelled in 2017

```
cancelled_2017 <- hotels |>
  filter(arrival_date_year == 2016)|>
  group_by(hotel)|>
  summarise(Cancelled = is_canceled)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was
## deprecated in dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember
##   that `reframe()` always returns an ungrouped data frame and
##   adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this
## warning was generated.
```

```
## `summarise()` has grouped output by 'hotel'. You can override
## using the `.groups` argument.
```

```
cancelled_2017
```

```
## # A tibble: 56,707 x 2
## # Groups:   hotel [2]
##    hotel      Cancelled
##    <chr>          <dbl>
##  1 City Hotel         1
##  2 City Hotel         0
##  3 City Hotel         0
##  4 City Hotel         0
##  5 City Hotel         0
##  6 City Hotel         0
##  7 City Hotel         0
##  8 City Hotel         0
##  9 City Hotel         0
## 10 City Hotel         0
## # i 56,697 more rows
```

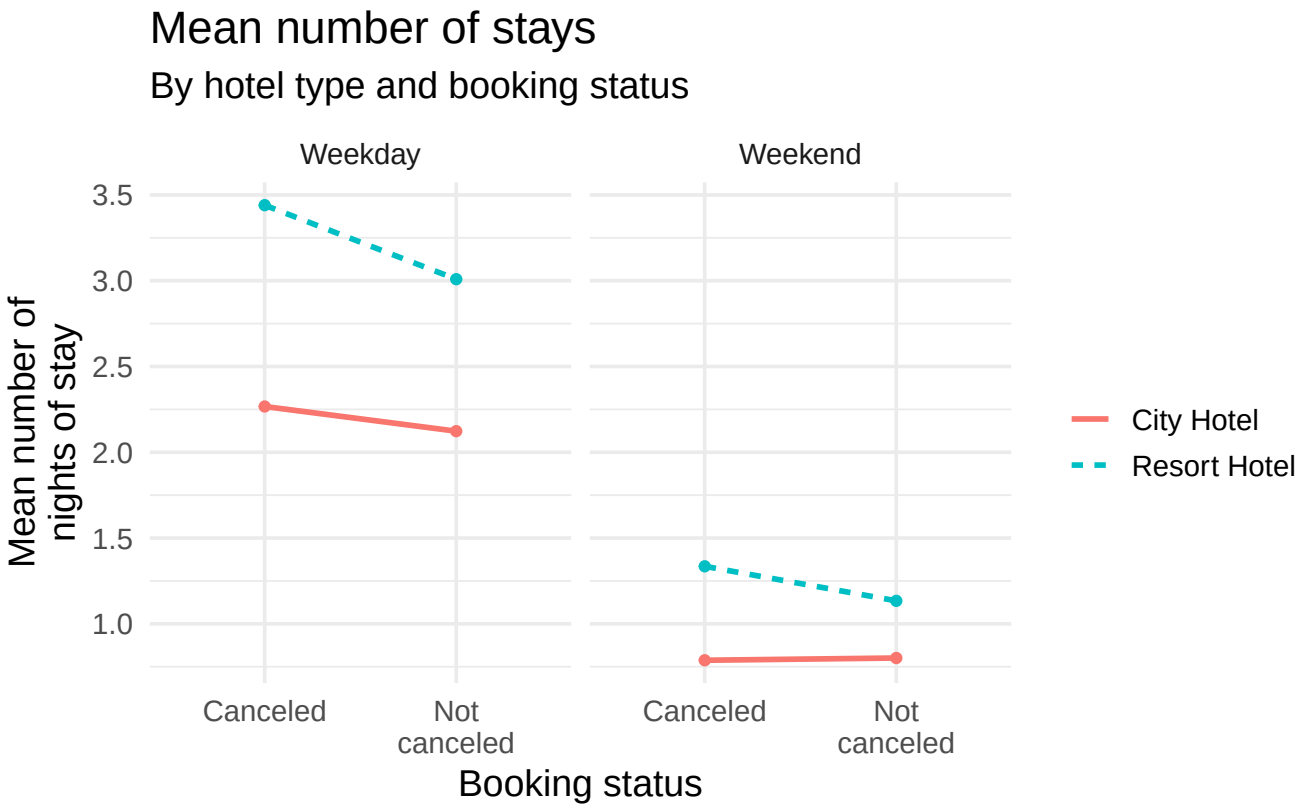
```
distinct(cancelled_2017)
```

```
## # A tibble: 4 x 2
## # Groups:   hotel [2]
##   hotel        Cancelled
##   <chr>            <dbl>
## 1 City Hotel           1
## 2 City Hotel           0
## 3 Resort Hotel         0
## 4 Resort Hotel         1
```

Let's plot a graphic of the hotels summary statistics

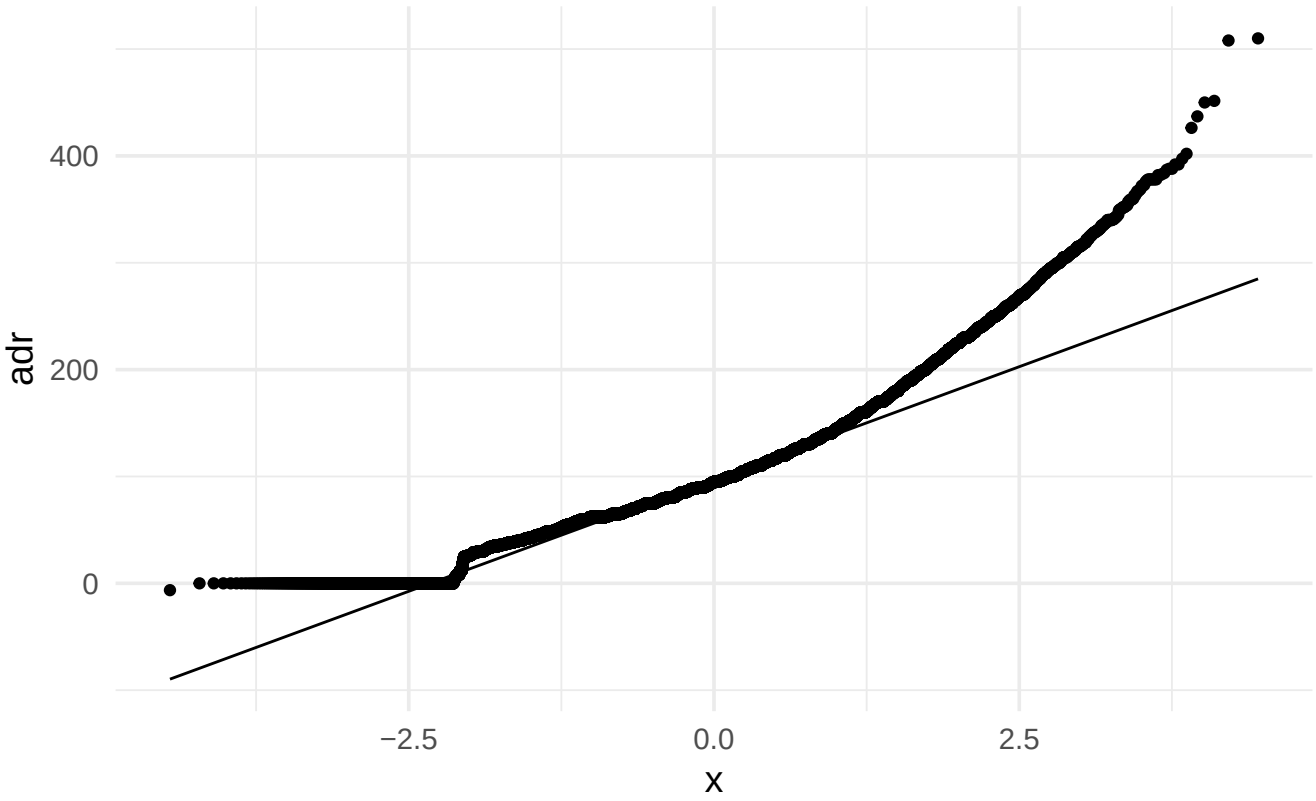
```
library(stringr)
## geom_point & geom_line - summary statistics
hotels_summary |>
  mutate(is_canceled = if_else(is_canceled == 0, "Not canceled", "Canceled")) |>
  pivot_longer(cols = starts_with("mean"),
               names_to = "day_type",
```

```
              values_to = "mean_stays",
              names_prefix = "mean_stays_in_") |>
  mutate(
    day_type = if_else(str_detect(day_type, "weekend"), "Weekend", "Weekday")
    ) |>
  ggplot(aes(x = str_wrap(is_canceled, 10), y = mean_stays,
             group = hotel, color = hotel)) +
  geom_point(show.legend = FALSE) +
  geom_line(aes(linetype = hotel), linewidth = 1) +
  facet_wrap(~day_type) +
  labs(
    x = "Booking status",
    y = "Mean number of\nnights of stay",
    color = NULL, linetype = NULL,
    title = "Mean number of stays",
    subtitle = "By hotel type and booking status")
```



Let's explore the trend in daily rate < 4000

```
# label: Daily rate trends at the two hotels
hotels |>
  filter(adr < 4000) |>
  ggplot(aes(sample = adr)) +
  stat_qq() +
  stat_qq_line() +
  labs(y = "adr")
```



Key Findings

1. Hotel prices have followed an upward trend between 2015 & 2017 at the listed two hotels.
2. The average booking price at both hotels between 2015 & 2017 was
   $ 101.8311
3. The average number of adults per booking at either hotel is Mean = 1.856 ;
4. The average number of babies staying with adults at each booking is Mean = 0.007949;
5. The average cancellation at both hotels is Mean = 0.3704.
6. The most busiest months are also the most expensive, ie.,July and August (summer) and then December (the holiday season.).
7. City Hotel is the hotel that has charged the most at $5400.00 in March 2016. The booking was eventually cancelled.

```
?str_detect
```

Third Data Set:

The present data known as babynames was curated by the Social Security Administration (SSA). The data set provides the first names of all newborn Americans (US Baby Names) from 1880 to 2017 Version.

Our research question : This analysis will try to find out what names were popular and distinctive at the end of the 19th century and what names were less trendy at that time.

Data exploration

Let's take a peek at babies_names

```
glimpse(baby_names)
```

```
## Rows: 258,000
```

```
## Columns: 4
## $ year    <int> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880,~
## $ name    <chr> "John", "William", "James", "Charles", "George"~
## $ percent <dbl> 0.081541, 0.080511, 0.050057, 0.045167, 0.04329~
## $ sex     <chr> "boy", "boy", "boy", "boy", "boy", "boy", "boy"~
```

```
head (baby_names)
```

```
##   year    name  percent sex
## 1 1880    John 0.081541 boy
## 2 1880 William 0.080511 boy
## 3 1880   James 0.050057 boy
## 4 1880 Charles 0.045167 boy
## 5 1880  George 0.043292 boy
## 6 1880   Frank 0.027380 boy
```

```
str(baby_names)
```

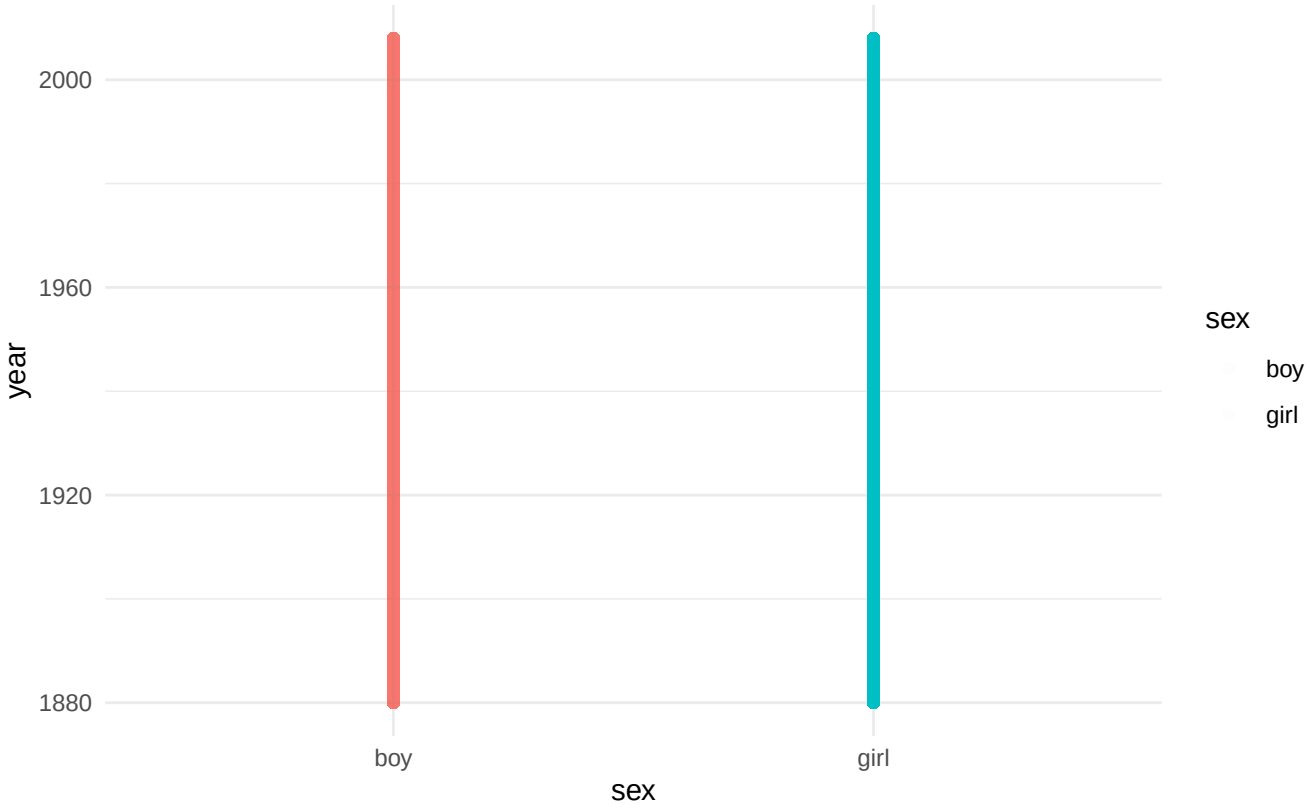
```
## 'data.frame':    258000 obs. of  4 variables:
##  $ year   : int  1880 1880 1880 1880 1880 1880 1880 1880 1880 1880 ...
##  $ name   : chr  "John" "William" "James" "Charles" ...
##  $ percent: num  0.0815 0.0805 0.0501 0.0452 0.0433 ...
##  $ sex    : chr  "boy" "boy" "boy" "boy" ...
```

```
# Label: Summary stats
summary(baby_names)
```

```
##       year          name              percent
##  Min.   :1880   Length:258000      Min.   :0.0000260
##  1st Qu.:1912   Class :character   1st Qu.:0.0000810
##  Median :1944   Mode  :character   Median :0.0001640
##  Mean   :1944                      Mean   :0.0008945
##  3rd Qu.:1976                      3rd Qu.:0.0005070
##  Max.   :2008                      Max.   :0.0815410
##      sex
##  Length:258000
##  Class :character
##  Mode  :character
##
##
##
```

Data visualization

```
ggplot(baby_names, aes(sex, y= year, colour = sex))+
geom_point(alpha = 0.01)+
  theme_minimal()
```



Data transformation: Pivot - wider

```
# Baby_names_wide
baby_names_wide <- baby_names |>
  pivot_wider(names_from = name, values_from = percent)

head(baby_names_wide)
```

```
## # A tibble: 6 x 6,784
##    year sex     John William  James Charles George  Frank Joseph
##   <int> <chr>  <dbl>   <dbl>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>
## 1  1880 boy   0.0815  0.0805 0.0501  0.0452 0.0433 0.0274 0.0222
## 2  1881 boy   0.0810  0.0787 0.0503  0.0428 0.0431 0.0262 0.0227
## 3  1882 boy   0.0783  0.0762 0.0483  0.0417 0.0426 0.0260 0.0219
## 4  1883 boy   0.0791  0.0746 0.0464  0.0429 0.0421 0.0265 0.0224
## 5  1884 boy   0.0765  0.0725 0.0464  0.0391 0.0404 0.0262 0.0221
## 6  1885 boy   0.0755  0.0694 0.0446  0.0397 0.0403 0.0265 0.0219
## # i 6,775 more variables: Thomas <dbl>, Henry <dbl>,
## #   Robert <dbl>, Edward <dbl>, Harry <dbl>, Walter <dbl>,
## #   Arthur <dbl>, Fred <dbl>, Albert <dbl>, Samuel <dbl>,
## #   David <dbl>, Louis <dbl>, Joe <dbl>, Charlie <dbl>,
## #   Clarence <dbl>, Richard <dbl>, Andrew <dbl>, Daniel <dbl>,
## #   Ernest <dbl>, Will <dbl>, Jesse <dbl>, Oscar <dbl>,
## #   Lewis <dbl>, Peter <dbl>, Benjamin <dbl>, ...
```

```
# Trendy names for boys in 1880
baby_names_wide |>
```

```
  filter (year == 1880) |>
  arrange(desc("name"))
```

```
## # A tibble: 2 x 6,784
##    year sex       John  William    James  Charles  George   Frank
##   <int> <chr>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>   <dbl>
## 1  1880 boy   0.0815   0.0805   0.0501   0.0452   4.33e-2 2.74e-2
## 2  1880 girl  0.000471 0.000307 0.000225 0.000113 2.66e-4 1.33e-4
## # i 6,776 more variables: Joseph <dbl>, Thomas <dbl>,
## #   Henry <dbl>, Robert <dbl>, Edward <dbl>, Harry <dbl>,
## #   Walter <dbl>, Arthur <dbl>, Fred <dbl>, Albert <dbl>,
## #   Samuel <dbl>, David <dbl>, Louis <dbl>, Joe <dbl>,
## #   Charlie <dbl>, Clarence <dbl>, Richard <dbl>, Andrew <dbl>,
## #   Daniel <dbl>, Ernest <dbl>, Will <dbl>, Jesse <dbl>,
## #   Oscar <dbl>, Lewis <dbl>, Peter <dbl>, Benjamin <dbl>, ...
```

```
library(dplyr)
# Popular baby girls names by the end of the 19th century
names_19th <- distinct(baby_names)|>
  filter (year == 1880 | year ==1900)|>
  group_by("sex") |>
  select(name, sex) |>
  arrange(desc(name))
```

```
## Adding missing grouping variables: `"sex"`
```

```
 print(names_19th )
```

```
## # A tibble: 4,000 x 3
## # Groups:   "sex" [1]
##    `"sex"` name   sex
##    <chr>   <chr>  <chr>
##  1 sex     Zula   girl
##  2 sex     Zula   girl
##  3 sex     Zora   girl
##  4 sex     Zora   girl
##  5 sex     Zona   girl
##  6 sex     Zona   girl
##  7 sex     Zollie boy
##  8 sex     Zola   girl
##  9 sex     Zola   girl
## 10 sex     Zoe    girl
## # i 3,990 more rows
```

```
# Less trendy baby names by the end of the 19th century
names_19th <- distinct(baby_names)|>
  filter (year == 1880 | year ==1900)|>
  group_by("sex") |>
  select(name, sex) |>
  arrange(name)
```

```
## Adding missing grouping variables: `"sex"`
```

```
 print(names_19th)
```

```
## # A tibble: 4,000 x 3
## # Groups:   "sex" [1]
```

```
##    `"sex"` name   sex
##    <chr>   <chr>  <chr>
##  1 sex     Aaron  boy
##  2 sex     Aaron  boy
##  3 sex     Ab     boy
##  4 sex     Abbie  girl
##  5 sex     Abbie  girl
##  6 sex     Abbott boy
##  7 sex     Abby   girl
##  8 sex     Abe    boy
##  9 sex     Abe    boy
## 10 sex     Abel   boy
## # i 3,990 more rows
```

```
# Poppular girl names in the end of the 19 century



girls_names_1880 <- baby_names |>
  filter(year == 1880, sex == "F")|>
  arrange(desc("baby_names"))


print(girls_names_1880)
```

```
## [1] year    name    percent sex
## <0 rows> (or 0-length row.names)
```

Key Findings

1. The most population baby boys name at the end of the nineteenth century were (in descending order), John, William , James, Charles, and George.
2. The most trendy names for baby girls was in descending order Zora , Zona, Zollie, Zola, and zoe.
3. The less trendy names were Aaron , Ab, Abe, Abel (for baby boys) and Abbie or Abby (for baby girls)