TODO

1. charger le dataset `us_city_populations` de la librairie `vizoR`

2. tracer un line chart de l'évolution de la population des villes US

3. Mettez en évidence les 5 plus grandes villes (hint: package gghighlight)

introduction gghighlight

4. Appliquez les principes de design de Tufte

5. BONUS: affichez le nom des villes directement à la fin de la ligne

6. Réalisez un streamgraph des 5 plus grandes villes US (hint: package ggTimeSeries)

TODO 2

Trouver une 3e visualization pertinente pour montrer l'évolution de la population des villes US.

```
data("us_city_populations")

n_cities = 5

last_ranks <- us_city_populations %>%
  filter(Year == max(Year)) %>%
  mutate(last_rank = Rank) %>%
  select(City, last_rank)

to_plot <- left_join(us_city_populations, last_ranks, by= 'City')

right_axis <-  to_plot %>%
  group_by(City) %>%
  top_n(1, Year) %>%
  ungroup() %>%
  top_n(n_cities, -last_rank)

ends <- right_axis %>%
  pull(Population)

labels <-  right_axis %>%
  pull(City)
```

```
p <- ggplot(to_plot, aes(x=Year, y = Population,
                         group = City, color = City)) +
  geom_line(size=1) +
  scale_x_continuous("", expand=c(0,0))+
  scale_y_continuous("",
                     labels=scales::comma_format(big.mark = " "),
                     sec.axis = sec_axis(~ .,
                                         breaks = ends,
                                         labels = labels ))+
  scale_color_viridis_d()+
  theme_elegant_dark()+
  theme(legend.position = "none",
        plot.margin = unit(c(1,3,1,1), "lines"),
        axis.line.y = element_blank(),
        axis.line.x = element_blank(),
        axis.ticks.x = element_line(),
        panel.grid.major.y = element_line(color= 'grey30', size = .2) ) +
  gghighlight(max(last_rank) <= n_cities,
              use_direct_label = FALSE,
              label_key = City,
              unhighlighted_colour = "grey20")
```

```
p
```

```r
library(ggTimeSeries)
p <- to_plot %>% filter(City %in% labels) %>%
  ggplot(aes(x = Year, y = Population, group = City, fill = City)) +
  scale_y_continuous("", labels = scales::comma_format(big.mark = " "))+
  stat_steamgraph() +
  theme_elegant_dark() +
  scale_fill_viridis_d() +
  theme(plot.margin = unit(c(1,3,1,1), "lines"),
        axis.line.y = element_blank(),
        axis.line.x = element_blank(),
        axis.ticks.x = element_line(),
        panel.grid.major.y = element_line(color= 'grey30', size = .2) )
```

```
p
```

Barchart race

Load data

```
data("us_city_populations")

n_cities = 10

top_cities <-us_city_populations %>% filter(Rank <= n_cities) %>%
  select(City, State, Region) %>% distinct()
```

Create all missing dates

```
# create a data frame with all the years between min and max Year
all_years <- data.frame(Year = seq(min(us_city_populations$Year),
                                   max(us_city_populations$Year), 1))

# combine top_cities and all_years
all_combos <- merge(top_cities, all_years, all = T)

# combine all_combos with the original dataset
res_interp <- merge(us_city_populations, all_combos, all.y = T)
```

Interpolate the Populations when missing (linear interpolation here)

```
res_interp  <- res_interp  %>%
  group_by(City) %>%
  mutate(Population=approx(Year,Population,Year)$y)
```

Filter data

```
to_plot <- res_interp %>%
  group_by(Year) %>%
  arrange(-Population) %>%
  mutate(Rank=row_number()) %>%
  filter(Rank<=n_cities)
```

Ease transitions

```
to_plot_trans <- to_plot %>%
  group_by(City) %>%
  arrange(Year) %>%
  mutate(lag_rank = lag(Rank, 1),
         change = ifelse(Rank > lag(Rank, 1), 1, 0),
         change = ifelse(Rank < lag(Rank, 1), -1, 0)) %>%
  mutate(transition = ifelse(lead(change, 1) == -1, -.9, 0),
         transition = ifelse(lead(change,2) == -1, -.5, transition),
         transition = ifelse(lead(change,3) == -1, -.3, transition),
         transition = ifelse(lead(change, 1) == 1, .9, transition),
         transition = ifelse(lead(change,2) == 1, .5, transition),
         transition = ifelse(lead(change,3) == 1, .3, transition)) %>
  mutate(trans_rank = Rank + transition)
```

Make the plot

```
p <- to_plot_trans %>%
  ggplot(aes(x = -trans_rank,y = Population, group =City)) +
   geom_tile(aes(y = Population / 2, height = Population, fill = Region),
             width = 0.9) +
  geom_text(aes(label = City),
            hjust = "right", colour = "white",
            fontface="bold", nudge_y = -100000) +
  geom_text(aes(label = scales::comma(Population,big.mark = ' ')),
            hjust = "left", nudge_y = 100000, colour = "grey90") +
  coord_flip(clip="off") +
  hrbrthemes::scale_fill_ipsum() +
  scale_x_discrete("") +
  scale_y_continuous("",labels=scales::comma_format(big.mark = " ")) +
  theme_elegant_dark(base_size = 20) +
  theme(
    panel.grid.minor.x=element_blank(),
    axis.line = element_blank(),
    panel.grid.major= element_line(color='lightgrey', size=.2),
    legend.position = c(0.6, 0.2),
    plot.margin = margin(1,1,1,2,"cm"),
    plot.title = element_text(hjust = 0),
    axis.text.y=element_blank(),
    legend.text = element_text(size = 15),
    legend.background = element_blank()) +
  # gganimate code to transition by year:
  transition_time(Year) +
  ease_aes('cubic-in-out') +
  labs(title='Evolution des plus grandes villes US',
       subtitle='Population en {round(frame_time,0)}')
```

```
animate(p, nframes = 400, fps = 25, end_pause = 30, width = 1200)
anim_save("bar_race.gif", animation = last_animation())
```