Simulating Population Growth

```
# load packages
library(tidyverse)
library(deSolve)
```

Consider the data set `us-historical-population-uscensus.csv` - provided with this mini-assignment. This data set was from the Historical Population Change Data (1910-2020) of the US Census Bureau. The data set contain population estimates by decade of US states and regions. The variables includes:

- *Name* - The name of the US state or region
- *GeographyType* - A label whether the row represents a "state" or "region"
- *Year* - The census year/decade
- *ResidentPopulation* - The estimated population of the US state or region

```
pop_us <- read_csv("us-historical-population-uscensus.csv")
```

1. **What is the population percent change in time?** Consider the temporal population data of three states. The code below filters three states from the data and it computes the percent change of each chosen state.. Choose different combination of states other than what's listed below.

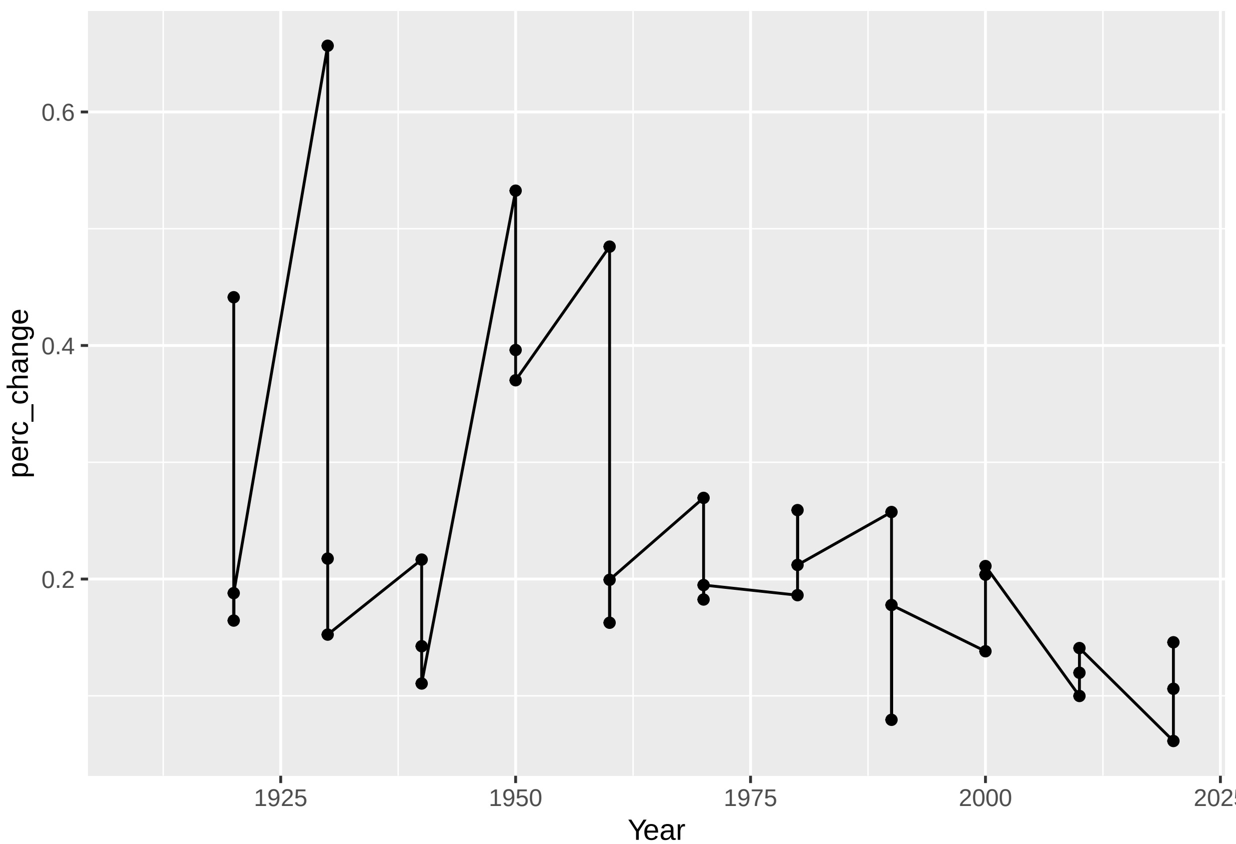
```r
# define state name
### [BEGIN] - EDIT STATES HERE
state_name <- c("California","Oregon","Washington")
### [END] - EDIT STATES HERE

# filter population data of a state
state_df <- pop_us %>%
  filter(Name %in% state_name) %>%
  arrange(Year) # sort population by year

# function to compute the percent change given population data
percent_change <- function(pop_vect){
  result <- c(NA)
  for (i in 2:length(pop_vect)){
    change <- ((pop_vect[i]-pop_vect[i-1])/pop_vect[i-1])
    result <- c(result,change)
  }
  return(result)
}

# apply the function to each state
state_df_new <- state_df %>%
  group_by(Name) %>%
  mutate(perc_change = percent_change(ResidentPopulation))
```

The code below plots - *improperly* - the yearly percent change. Your task is to modify the `ggplot` pipeline to properly plot the population change for each state, and the plot should have the following.

a. A dashed horizontal line on $y = 0$.
b. Properly labeled axes and titles, and the x-axis needs to be discrete showing all tick mark labels.
c. Adjust the scales and axes limits accordingly.

Provide A short paragraph of your observations, and explain - in terms of birth and death rates - why do think the percent change is linearly/non-linearly decreasing or increasing.

```r
### [BEGIN] - MODIFY GGPLOT PIPELINE HERE
p1 <- ggplot(data = state_df_new, aes(x = Year, y = perc_change)) +
  geom_line() +
  geom_point()
### [END] - MODIFY GGPLOT PIPELINE HERE
p1
```



2. **Which models can best describe the data?** In this problem, you will attempt to fit two simple population models using the data. The population models we consider here are the exponential growth model and the logistic growth model.

Exponential Model. The differential equation that governs the exponential *change in population over time* is given below.

$$\frac{dN(t)}{dt} = rN(t)$$

where r is the rate at which the population changes in time depending on the population abundance N. Below is a code that defines the exponential differential equation.

```
# Differential equation for the exponential growth model
exp_growth <- function(t, N, p) {
  with (as.list(p), {
    dNdt <- rate*N
    return(list(dNdt))
  })
}
```

Logistic Model. Another way to define an equation that governs the *change in population over time* is using a logistic equation where we have two parameters. The differential logistic equation is defined below.

$$\frac{dN(t)}{dt} = rN(t)\left(1 - \frac{N(t)}{K}\right)$$

where r is the rate and K is the limiting capacity. Below is a code that defines the logistic differential equation.

```
# Differential equation for the logisitc growth model
logistic_growth <- function(t, N, p) {
  with (as.list(p), {
    dNdt <- rate*N*(1-N/K)
    return(list(dNdt))
  })
}
```

Here, we are solving these differential equations numerically - meaning we solve these equations using our computer instead of using Calculus. Note that you can solve these equations analytically using Calculus, but some systems of non-linear differential equations actually can't be solved analytically.

```r
### [BEGIN] - CHOOSE STATE FROM STATES CHOSEN FROM THE PREVIOUS PROBLEM
single_state_data <-  state_df_new %>%
  filter(Name == "California")
### [END] - CHOOSE STATE FROM STATES CHOSEN FROM THE PREVIOUS PROBLEM

# time vector in years by decade
T_init = 1910
T_lim = 2020
t = seq(T_init,T_lim,10)

# initial population (from actual data - one state)
N_init <- c(N = single_state_data$ResidentPopulation[1])
```

a. **Solving the Exponential Model.** The goal here is to decide what value of r is a good fit. Using the data, your task is to estimate r (graphically - see part c) and use that estimate to simulate the population using the exponential differential equation.

```
### [BEGIN] - ESTIMATE R HERE
# notice that this variable is used in the next code block
estimatedRate <- 0.035 # update this value
### [END] - ESTIMATE R HERE
```

```
# parameters of the model
pars_exp <- c (
   rate = estimatedRate) # rate at which the population changes

# solve ode and them produce simulated data
N_t <- ode(y = N_init, times = t, parms = pars_exp, func = exp_growth)

# convert simulated data into tibble
df_sim_exp <- as_tibble(N_t) %>%
  mutate(Year = as.numeric(time),
         N_exp = as.numeric(N)) %>%
  select(-time,-N)
```

b. **Solving the Logistic Model.** The goal here is to decide what value of r and K is a good fit. Using the data, your task is to estimate r and K, and use those estimates to simulate the population using the logistic differential equation.

Using the graphical approach (see part c), manually adjust values below so that the data fits with the model. You can use the estimate rate r from the previous problem as a starting point.

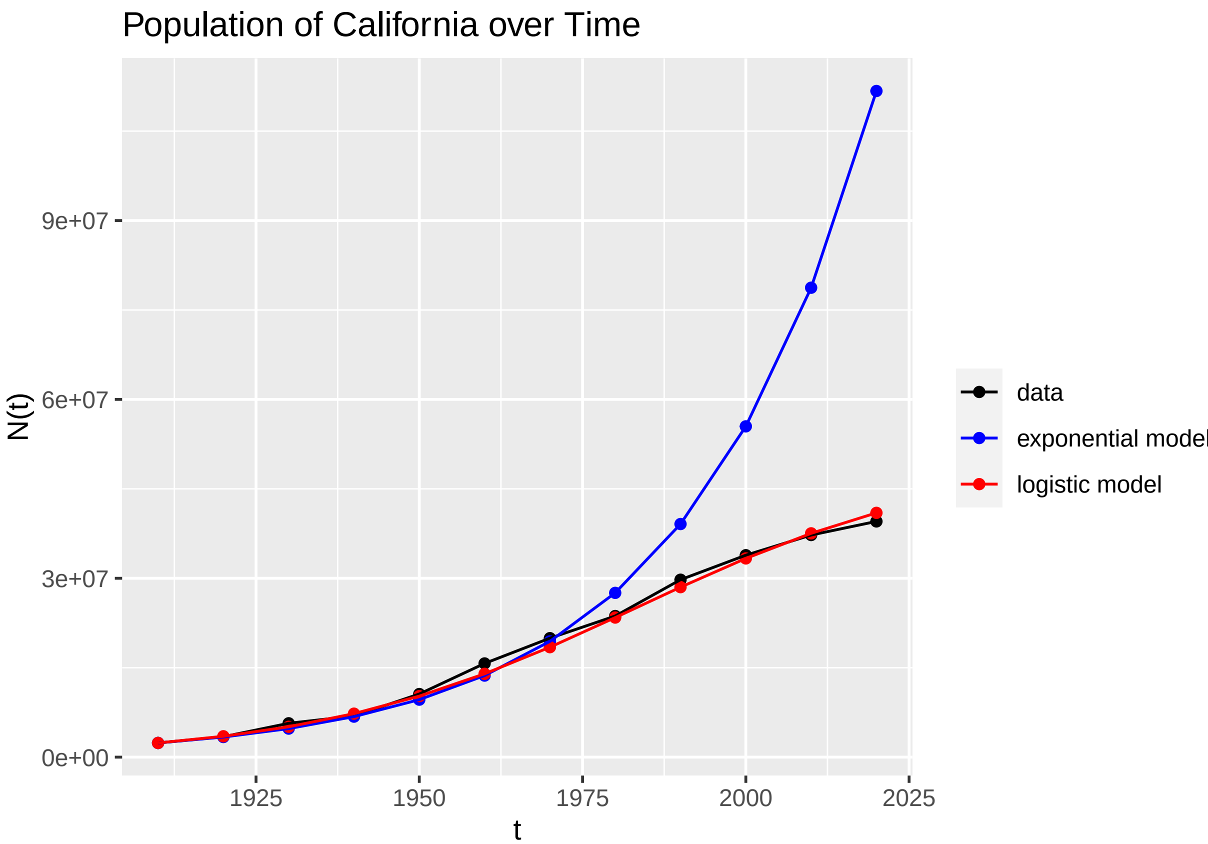
```
### [BEGIN] - ESTIMATE R HERE
# notice that these variables are used in the next code block
estimatedRate <- 0.041 # update this value
estimatedK <- 50000000 # update this value
### [END] - ESTIMATE R HERE
```

```
# parameters of the model
pars_logistic <- c (
   rate = estimatedRate, # rate at which the population changes
   K = estimatedK ) # limiting capacity

# solve ode and them produce simulated data
N_t <- ode(y = N_init, times = t, parms = pars_logistic, func = logistic_growth)

# convert simulated data into tibble
df_sim_logistic <- as_tibble(N_t) %>%
  mutate(Year = as.numeric(time),
         N_logistic = as.numeric(N)) %>%
  select(-time,-N)
```

c. **Plotting the models with the data.** The goal here is to plot the models with the data. Compare the models and provide a paragraph describing your observations.

```
# join the data and the simulated data using the models
df_all <- single_state_data %>%
  left_join(df_sim_exp, by = c("Year" = "Year")) %>%
  left_join(df_sim_logistic, by = c("Year" = "Year"))

# transform the data for plotting
df_all_pivot <- df_all %>%
  pivot_longer(
    cols = c("ResidentPopulation","N_exp","N_logistic"),
    names_to = "type",
    values_to = "value") %>%
  mutate(type_factor = recode_factor(type,
                `ResidentPopulation` = "data",
                `N_exp` = "exponential model",
                `N_logistic` = "logistic model"))
```

```
# plot data and models
ggplot(data = df_all_pivot, aes(x = Year, y = value, color = type_factor)) +
  geom_point() +
  geom_line() +
  scale_color_manual(values = c("data" = "black",
                              "exponential model" = "blue",
                              "logistic model" = "red")) +
  labs(x = "t",
       y = "N(t)",
       color = "",
       title = "Population of California over Time")
```


Population of California over Time
9e+07
6e+07
3e+07
0e+00
N(t)
1925
1950
1975
2000
2025
t
data
exponential model
logistic model