Assignment 4

Feiyu Zheng (fz114)

2022/2/18

(1) Create a new function that , given an 'lm' object, returns the top n residuals arranged in descending order according to their largest absolute values (but returns the residuals, not the absolute value of the residuals), where the default value for n is 5. The function should give a clear error message if n is larger than the number of residuals. Demonstrate that your functions works applying it to *mtcars.lm <- lm(mpg ~ disp, data = mtcars)* first with no argument for n, then with n = 6, and then with n = 40 (error message expected)

Create the function

```
top_n_residual <- function(obj, n = 5){
  # get the number of residuals
  rowNum <- obj %>%
    residuals() %>%
    length()

  if(n > rowNum){
    # throws error message if argument n is larger than
    # the number of residuals
    stop(paste("The number of residuals is ",
               rowNum,
               " but argument n is ",
               n,
               " which is larger than the number of residuals.",
               sep = ""))
  }else{
    result <- obj %>%
      residuals() %>%
      abs() %>% # absolute value
      sort(decreasing = T) %>% # sort in descending order
      head(n) %>% # top n
      tibble(name = names(.), residuals = .) # convert result to tibble
  }
  result
}
```

Define the lm object

```
mtcars.lm <- lm(mpg ~ disp, data = mtcars)
```

With no argument for n

```
top_n_residual(mtcars.lm)
```

```
## # A tibble: 5 x 2
##   name             residuals
##   <chr>                <dbl>
## 1 Toyota Corolla        7.23
## 2 Pontiac Firebird      6.09
## 3 Fiat 128              6.04
## 4 Merc 280C             4.89
## 5 Lotus Europa          4.72
```

With n = 6

```
top_n_residual(mtcars.lm, 6)
```

```
## # A tibble: 6 x 2
##   name              residuals
##   <chr>                 <dbl>
## 1 Toyota Corolla         7.23
## 2 Pontiac Firebird       6.09
## 3 Fiat 128               6.04
## 4 Merc 280C              4.89
## 5 Lotus Europa           4.72
## 6 Hornet Sportabout      3.94
```

With n = 40

```
top_n_residual(mtcars.lm, 40)
```

```
## Error in top_n_residual(mtcars.lm, 40): The number of residuals is 32 but argument n is 40 which is l
```

(2) Split the gapminder by country and use map() to calculate, by country, the R-squared for the linear model lifeExp ~ log10(gdpPercap). Using ggplot2, make a set of boxplots of R-squared by continent.

```
library(gapminder) # import gapminder package

gapminder %>%
  split(.$country) %>% # split by country
  # linear model
  map(~lm(lifeExp ~ log10(gdpPercap), data = .)) %>%
  map(summary) %>% # get summary
  map_dbl("r.squared") %>% # get R-squared from summary
  tibble(country = names(.), rsquared = .) %>% # convert to tibble
  # add continent column corresponding to country
  inner_join(distinct(select(gapminder, country, continent)), by = "country") %>%
  ggplot(aes(continent, rsquared)) +
  geom_boxplot() + # boxplot
  labs(
    x = "Continent",
    y = "R-squared",
```

```
    title = "Boxplots of R-squared by continent") +
  theme(plot.title = element_text(hjust = 0.5))
```