# HW 05 Soultions

*Elijah Willie*

*November 08, 2018*

## Introduction

In this assignment, I will attempt to do the following tasks:

- Writing functions that are useful to analyze the gapminder dataset
- Work with a nested data frame

### Load in required libraries

First we will load in all the libraries we will be using for this assignment.

```
library(gapminder)
suppressMessages(library("tidyverse"))
library(knitr)
suppressMessages(library(reshape2))
suppressMessages(library(Hmisc))
suppressMessages(library(gridExtra))
library(knitr)
suppressMessages(library(MASS))
suppressMessages(library(broom))
```

### Part 1: Writing Functions

#### For this section, I will do the following:

- Fit a model model linear model that predicts `life expectancy` for `year`, `pop` or `gdpPercap` for a given country.
- Create a function to plot a linear fit

#### Create a function that fits a linear model

```
fit_model <- function(Country, variable, data){
#extract the data for the specific country
data_country <-  as.data.frame(data) %>%
  filter(country == Country)

#fit the model
if(variable == "pop"){
  model_fit <- lm(lifeExp ~ pop, data_country)
}
if(variable == "gdpPercap"){
  model_fit  <- lm(lifeExp ~ gdpPercap, data_country)
}
if(variable == "year"){
  model_fit <- lm(lifeExp ~ year, data_country)
```

```
}

return(model_fit)
}
```

### Create a function that plots the a given regression fit

```
plotReg <- function (fit) {

require(ggplot2)

p <- ggplot(fit$model, aes_string(x = names(fit$model)[2], y = names(fit$model)[1])) +
  geom_point() +
  stat_smooth(method = "lm", col = "red") +
  labs(title = paste("Intercept =",signif(fit$coef[[1]],5 ),
                     " Slope =",signif(fit$coef[[2]], 5),
                     " P =",signif(summary(fit)$coef[2,4], 5)))
return(p)
}
```

### Test out the functions

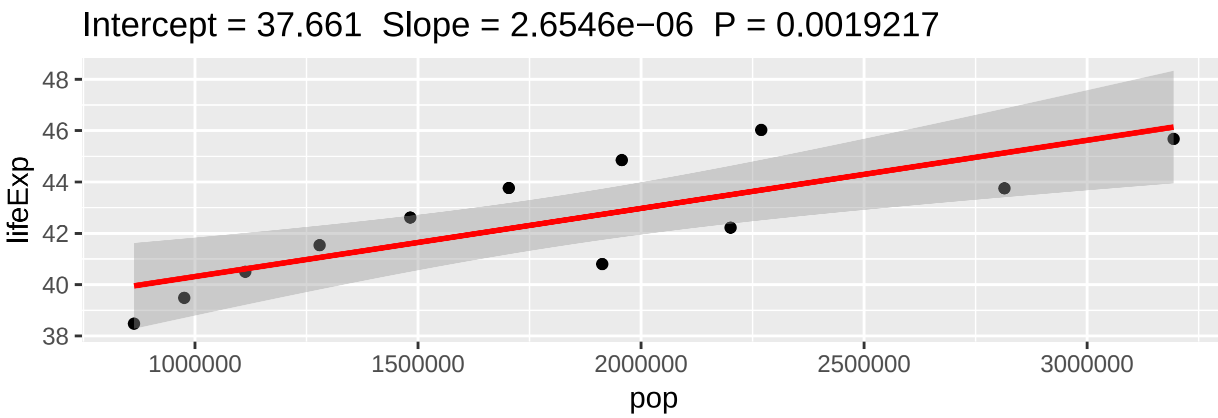
Predict life expectancy for `Liberia` and `Ghana` using population.

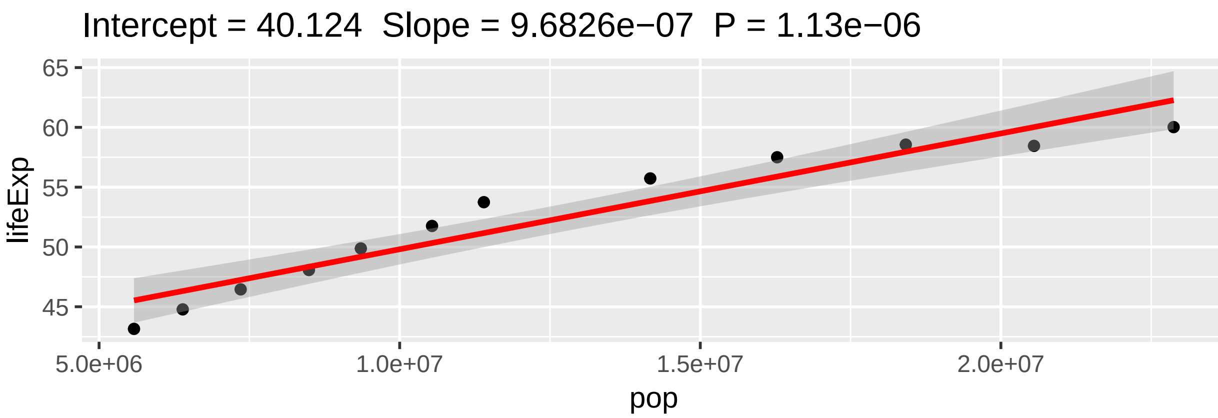
```
#create a set of countries
countries <- c("Liberia", "Ghana")

#fit a linear model using population
res <- lapply(countries, fit_model, variable = "pop", data = gapminder)

#generate the plots
plots <- lapply(res, plotReg)

#put them on a grid
grid.arrange(plots[[1]], plots[[2]])
```



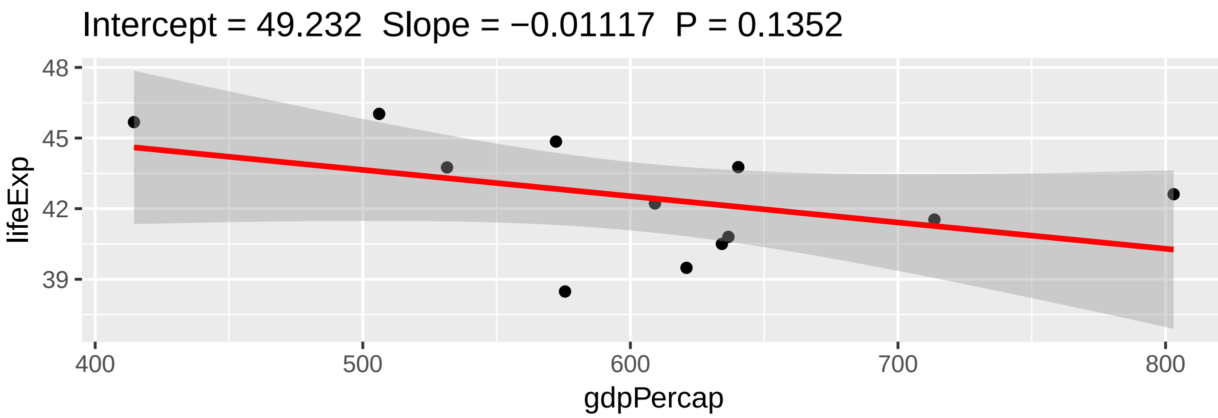

Looking at the plots above, we see that population does a much better job at predicting life expectancy when for `Ghana` When compared with `Liberia`.

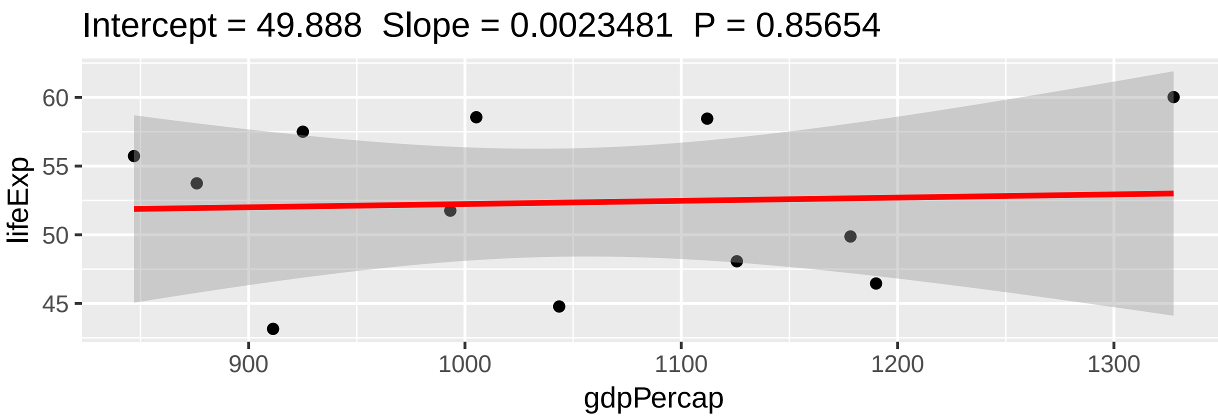
Now predict again, but this time use `gdpPercap`

```
#fit a linear model using gdpPercap
res <- lapply(countries, fit_model, variable = "gdpPercap", data = gapminder)

#generate the plots
plots <- lapply(res, plotReg)

#put them on a grid
grid.arrange(plots[[1]], plots[[2]])
```



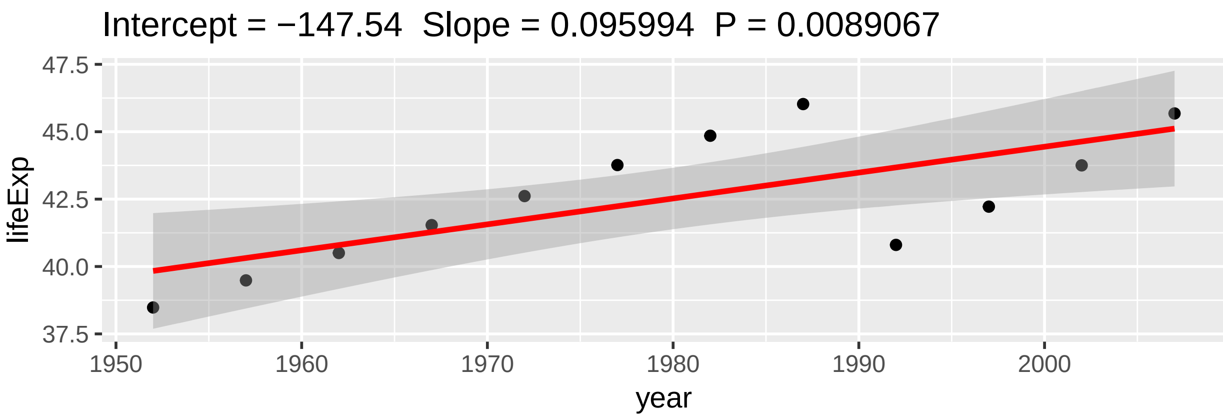



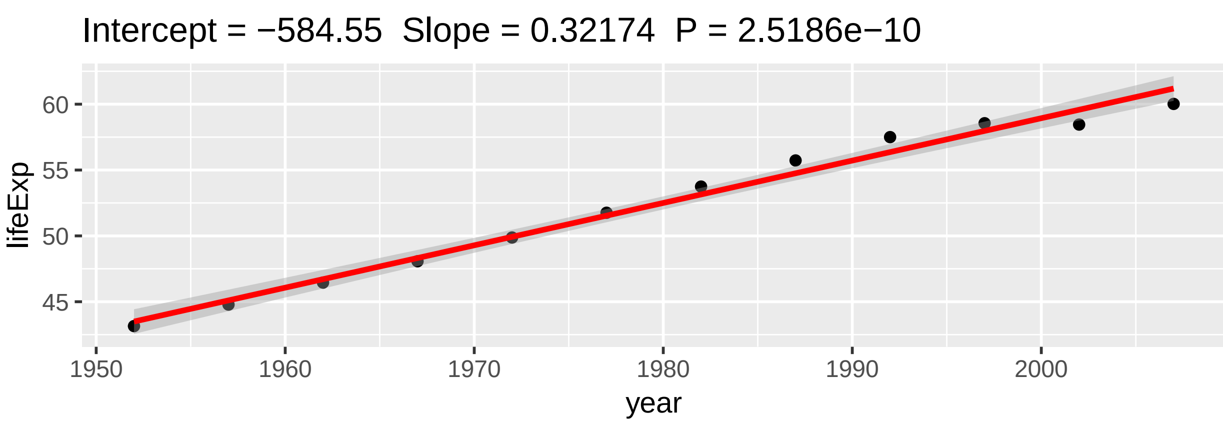

Here we see that `gdpPercap` does a pretty terrible job at predicting life xpectancy. Maybe using year would help.

```
#fit a linear model using population
res <- lapply(countries, fit_model, variable = "year", data = gapminder)

#generate the plots
plots <- lapply(res, plotReg)

#put them on a grid
grid.arrange(plots[[1]], plots[[2]])
```




That is much better indeed. We still see that it performs much better for `Ghana` than `Liberia`.

## Part 2 Work with a nested data frame

In the part, I will attempt to do the following

- Nest the data by country (and continent).
- Fit a model of life expectancy against year. Possibly quadratic, possibly robust.
- Use functions for working with fitted models or the `broom` package to get information out of your linear models.
- Use the usual dplyr, tidyr, and ggplot2 workflows to explore, e.g., the estimated cofficients.
- I will be following the tutorial here closely, but with my own twists and turns.

```
#nest the gapminder dataset by country and continent
gap_nested <- gapminder %>%
  group_by(continent, country) %>%
  nest()

#lets get some information on the new dataset
gap_nested
```

```
## # A tibble: 142 x 3
##    continent country     data
##    <fct>     <fct>       <list>
##  1 Asia      Afghanistan <tibble [12 x 4]>
##  2 Europe    Albania     <tibble [12 x 4]>
##  3 Africa    Algeria     <tibble [12 x 4]>
```

```
##   4 Africa    Angola      <tibble [12 x 4]>
##   5 Americas  Argentina   <tibble [12 x 4]>
##   6 Oceania   Australia   <tibble [12 x 4]>
##   7 Europe    Austria     <tibble [12 x 4]>
##   8 Asia      Bahrain     <tibble [12 x 4]>
##   9 Asia      Bangladesh  <tibble [12 x 4]>
##  10 Europe    Belgium     <tibble [12 x 4]>
## # ... with 132 more rows
```

```r
nrow(gap_nested)
```

```
## [1] 142
```

```r
ncol(gap_nested)
```

```
## [1] 3
```

We see that the new dataframe has 142 rows and 3 columns

Now lets take a look at some of the elements of the new nested dataframe

```r
#we can use indices to get information about the data
kable(gap_nested[[1, "data"]])
```

| year | lifeExp | pop | gdpPercap |
|---|---|---|---|
| 1952 | 28.801 | 8425333 | 779.4453 |
| 1957 | 30.332 | 9240934 | 820.8530 |
| 1962 | 31.997 | 10267083 | 853.1007 |
| 1967 | 34.020 | 11537966 | 836.1971 |
| 1972 | 36.088 | 13079460 | 739.9811 |
| 1977 | 38.438 | 14880372 | 786.1134 |
| 1982 | 39.854 | 12881816 | 978.0114 |
| 1987 | 40.822 | 13867957 | 852.3959 |
| 1992 | 41.674 | 16317921 | 649.3414 |
| 1997 | 41.763 | 22227415 | 635.3414 |
| 2002 | 42.129 | 25268405 | 726.7341 |
| 2007 | 43.828 | 31889923 | 974.5803 |

```r
#let see if we can get the country
gap_nested$country[[1]]
```

```
## [1] Afghanistan
## 142 Levels: Afghanistan Albania Algeria Angola Argentina ... Zimbabwe
```

```r
#we can even get data for a specific country
gap_nested[gap_nested$country == "Liberia",][["data"]]
```

```
## [[1]]
## # A tibble: 12 x 4
##     year lifeExp     pop gdpPercap
##    <int>   <dbl>   <int>     <dbl>
##  1  1952    38.5  863308      576.
##  2  1957    39.5  975950      621.
##  3  1962    40.5 1112796      634.
##  4  1967    41.5 1279406      714.
##  5  1972    42.6 1482628      803.
##  6  1977    43.8 1703617      640.
```

```
##  7  1982    44.9 1956875      572.
##  8  1987    46.0 2269414      506.
##  9  1992    40.8 1912974      637.
## 10  1997    42.2 2200725      609.
## 11  2002    43.8 2814651      531.
## 12  2007    45.7 3193942      415.
```

```
#lets try it for another country
gap_nested[gap_nested$country == "Ghana",][["data"]]
```

```
## [[1]]
## # A tibble: 12 x 4
##     year lifeExp      pop gdpPercap
##    <int>   <dbl>    <int>     <dbl>
##  1  1952    43.1  5581001      911.
##  2  1957    44.8  6391288     1044.
##  3  1962    46.5  7355248     1190.
##  4  1967    48.1  8490213     1126.
##  5  1972    49.9  9354120     1178.
##  6  1977    51.8 10538093      993.
##  7  1982    53.7 11400338      876.
##  8  1987    55.7 14168101      847.
##  9  1992    57.5 16278738      925.
## 10  1997    58.6 18418288     1005.
## 11  2002    58.5 20550751     1112.
## 12  2007    60.0 22873338     1328.
```

Now lets fit some models to the data and see how they behave

```
#fit a linear model
lin_fit <- lm(lifeExp ~ year, data = gap_nested[[1, "data"]])
paste("linear fit summary: ")
```

```
## [1] "linear fit summary: "
```

```
summary(lin_fit)
```

```
##
## Call:
## lm(formula = lifeExp ~ year, data = gap_nested[[1, "data"]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5447 -0.9905 -0.2757  0.8847  1.6868
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -507.53427   40.48416  -12.54 1.93e-07 ***
## year           0.27533    0.02045   13.46 9.84e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.223 on 10 degrees of freedom
## Multiple R-squared:  0.9477, Adjusted R-squared:  0.9425
## F-statistic: 181.2 on 1 and 10 DF,  p-value: 9.835e-08
```

```r
#compute the AIC and BIC for this model
paste("linear fit AIC and BIC are: ", AIC(lin_fit), BIC(lin_fit))
```

```
## [1] "linear fit AIC and BIC are:  42.6938696916089 44.1485896409729"
```

```r
#fit a quadratic
quad_fit <- lm(lifeExp ~ poly(year, 2), data = gap_nested[[1, "data"]])
paste("Quad fit summary: ")
```

```
## [1] "Quad fit summary: "
```

```r
summary(quad_fit)
```

```
##
## Call:
## lm(formula = lifeExp ~ poly(year, 2), data = gap_nested[[1, "data"]])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75900 -0.51487  0.02653  0.51654  0.62231
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     37.4788     0.1691 221.693  < 2e-16 ***
## poly(year, 2)1  16.4623     0.5856  28.110 4.44e-10 ***
## poly(year, 2)2  -3.4446     0.5856  -5.882 0.000234 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5856 on 9 degrees of freedom
## Multiple R-squared:  0.9892, Adjusted R-squared:  0.9868
## F-statistic: 412.4 on 2 and 9 DF,  p-value: 1.41e-09
```

```r
#compute the AIC and BIC for this model
paste("Quad fit AIC and BIC are: ", AIC(lin_fit), BIC(lin_fit))
```

```
## [1] "Quad fit AIC and BIC are:  42.6938696916089 44.1485896409729"
```

```r
#fit a robust regression
#note that we will need to use the MASS library to do this
robust_fit <- rlm(lifeExp ~ year, data = gap_nested[[1, "data"]])
paste("Robust fit summary: ")
```

```
## [1] "Robust fit summary: "
```

```r
summary(robust_fit)
```

```
##
## Call: rlm(formula = lifeExp ~ year, data = gap_nested[[1, "data"]])
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5447 -0.9905 -0.2757  0.8847  1.6868
##
## Coefficients:
##             Value     Std. Error t value
## (Intercept) -507.5343   40.4842   -12.5366
## year           0.2753    0.0205    13.4629
##
```

```
## Residual standard error: 1.526 on 10 degrees of freedom
```

```r
#compute the AIC and BIC for this model
paste("Robust fit AIC and BIC are: ",AIC(robust_fit), BIC(robust_fit))
```

```
## [1] "Robust fit AIC and BIC are:  42.6938696916086 44.1485896409726"
```

We see that for `lifeEXp`, there is no difference in using a `linear`, `quadratic`, or `robust` model. They all return the same `AIC` and `BIC` values.

Now all this seems cumbesome, so lets create another function to fit the linear and the robust for us. :relieved:

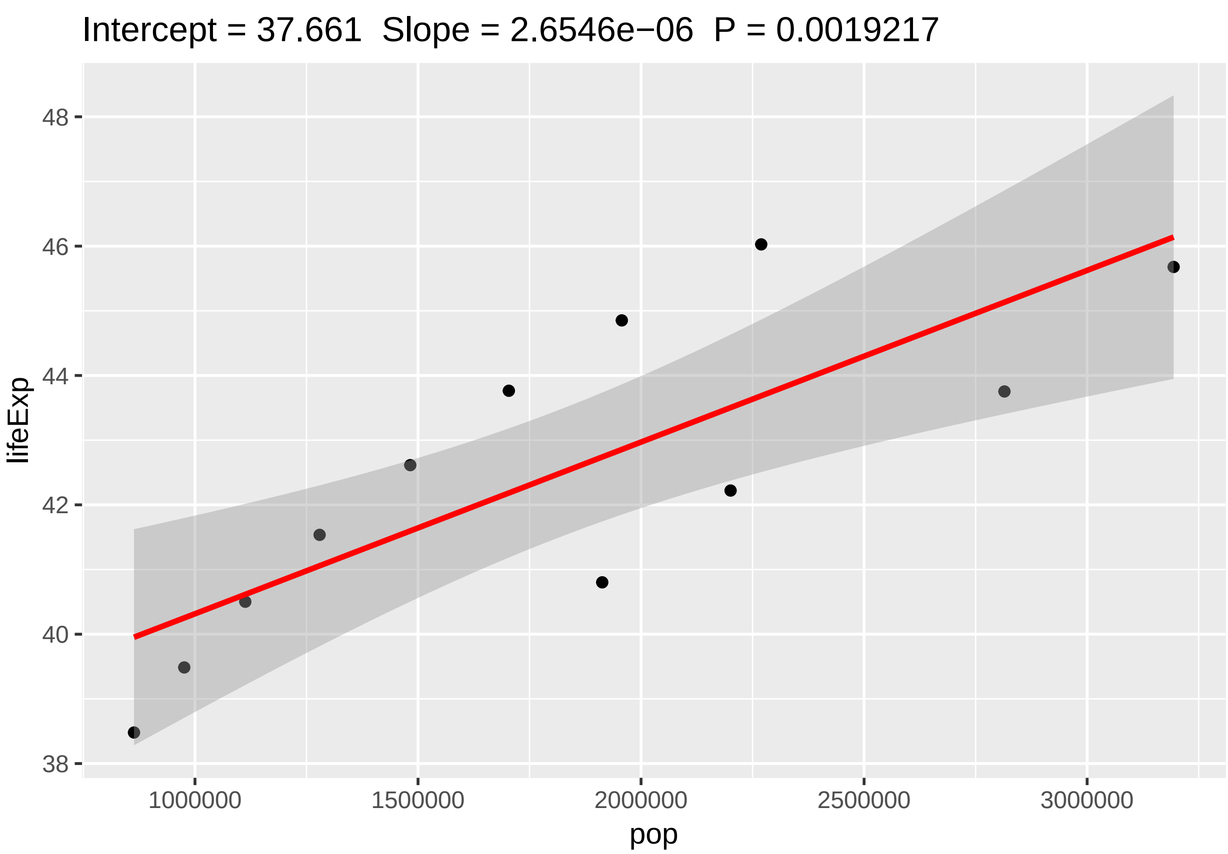
```r
nested_fit <- function(Country, type = "linear", variable = "pop"){
  country_dat <- as.data.frame(gap_nested[gap_nested$country == Country,][["data"]])
  if(type == "robust"){
    if(variable == "pop") fit <- rlm(lifeExp ~ pop, data = country_dat)
    if(variable == "year") fit <- rlm(lifeExp ~ year, data = country_dat)
    if(variable == "gdpPercap") fit <- rlm(lifeExp ~ gdpPercap, data = country_dat)
  }
  if(type == "linear"){
    if(variable == "pop") fit <- lm(lifeExp ~ pop, data = country_dat)
    if(variable == "year") fit <- lm(lifeExp ~ year, data = country_dat)
    if(variable == "gdpPercap") fit <- lm(lifeExp ~ gdpPercap, data = country_dat)
  }
  return(fit)
}
```

Lets test this function out. :smirk:

```r
#compute a fit for liberia using a regular linear model
res <- nested_fit("Liberia", variable = "pop")
#summarize the model
summary(res)
```

```
##
## Call:
## lm(formula = lifeExp ~ pop, data = country_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9374 -1.3067 -0.2875  1.1579  2.3414
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.766e+01  1.237e+00  30.447 3.42e-11 ***
## pop         2.655e-06  6.368e-07   4.169  0.00192 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.533 on 10 degrees of freedom
## Multiple R-squared:  0.6348, Adjusted R-squared:  0.5982
## F-statistic: 17.38 on 1 and 10 DF,  p-value: 0.001922
```

```r
#compute AIC and BIC
#using the function defined earlier, plot this fit
plotReg(res)
```




```r
#now fit a robust model
res_rob <- nested_fit("Liberia", type = "robust", variable = "pop")

#show some summary
summary(res_rob)
```

```
##
## Call: rlm(formula = lifeExp ~ pop, data = country_dat)
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9374 -1.3067 -0.2875  1.1579  2.3414
##
## Coefficients:
##             Value   Std. Error t value
## (Intercept) 37.6612  1.2369    30.4470
## pop          0.0000  0.0000     4.1688
##
## Residual standard error: 1.973 on 10 degrees of freedom
```

That was fun, now lets use the `map` function to do this for a few countries from part 1. :wink:

```r
#use the map function to fit to two countries using year as predictor
fits <- map(countries,nested_fit)
fits
```

```
## [[1]]
##
## Call:
## lm(formula = lifeExp ~ pop, data = country_dat)
```

```
##
## Coefficients:
## (Intercept)          pop
##   3.766e+01    2.655e-06
##
##
## [[2]]
##
## Call:
## lm(formula = lifeExp ~ pop, data = country_dat)
##
## Coefficients:
## (Intercept)          pop
##   4.012e+01    9.683e-07
```

We can finally do this for all countries in our dataset. :muscle:

```
#apply this function to all countries in the dataset
gap_nested <- gap_nested %>%
   mutate(fit = map(country, nested_fit))

#show the result
gap_nested
```

```
## # A tibble: 142 x 4
##    continent country     data              fit
##    <fct>     <fct>       <list>            <list>
##  1 Asia      Afghanistan <tibble [12 x 4]> <S3: lm>
##  2 Europe    Albania     <tibble [12 x 4]> <S3: lm>
##  3 Africa    Algeria     <tibble [12 x 4]> <S3: lm>
##  4 Africa    Angola      <tibble [12 x 4]> <S3: lm>
##  5 Americas  Argentina   <tibble [12 x 4]> <S3: lm>
##  6 Oceania   Australia   <tibble [12 x 4]> <S3: lm>
##  7 Europe    Austria     <tibble [12 x 4]> <S3: lm>
##  8 Asia      Bahrain     <tibble [12 x 4]> <S3: lm>
##  9 Asia      Bangladesh  <tibble [12 x 4]> <S3: lm>
## 10 Europe    Belgium     <tibble [12 x 4]> <S3: lm>
## # ... with 132 more rows
```

Finally, we will use the `broom` library to tidy up the results

```
#apply broom to each country
gap_nested <- gap_nested %>%
  mutate(tidy = map(fit, tidy))

#have a look at the result
gap_nested
```

```
## # A tibble: 142 x 5
##    continent country     data              fit      tidy
##    <fct>     <fct>       <list>            <list>   <list>
##  1 Asia      Afghanistan <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  2 Europe    Albania     <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  3 Africa    Algeria     <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  4 Africa    Angola      <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  5 Americas  Argentina   <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  6 Oceania   Australia   <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
```

```
##  7 Europe    Austria     <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  8 Asia      Bahrain     <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
##  9 Asia      Bangladesh  <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
## 10 Europe    Belgium     <tibble [12 x 4]> <S3: lm> <tibble [2 x 5]>
## # ... with 132 more rows
```

```
#lets have a look at the tidy table for one of the countries
gap_nested[gap_nested$country == "Liberia",][["tidy"]]
```

```
## [[1]]
## # A tibble: 2 x 5
##   term           estimate   std.error statistic  p.value
##   <chr>             <dbl>       <dbl>     <dbl>    <dbl>
## 1 (Intercept) 37.7        1.24             30.4  3.42e-11
## 2 pop          0.00000265 0.000000637      4.17 1.92e- 3
```

```
#we can finally return to the original data type for which we started using all the information we have
gap_coefs <- gap_nested %>%
   dplyr::select(continent, country, tidy) %>%
   unnest(tidy)
#show the final table
gap_coefs
```

```
## # A tibble: 284 x 7
##    continent country    term        estimate  std.error statistic  p.value
##    <fct>     <fct>      <chr>          <dbl>      <dbl>     <dbl>    <dbl>
##  1 Asia      Afghanist~ (Interce~    2.83e+1    2.31e+0     12.2  2.41e- 7
##  2 Asia      Afghanist~ pop          5.77e-7    1.34e-7      4.30 1.57e- 3
##  3 Europe    Albania    (Interce~    4.96e+1    1.94e+0     25.6  1.87e-10
##  4 Europe    Albania    pop          7.29e-6    7.17e-7     10.2  1.37e- 6
##  5 Africa    Algeria    (Interce~    3.57e+1    1.63e+0     21.8  9.09e-10
##  6 Africa    Algeria    pop          1.18e-6    7.59e-8     15.5  2.55e- 8
##  7 Africa    Angola     (Interce~    2.86e+1    1.92e+0     14.9  3.84e- 8
##  8 Africa    Angola     pop          1.28e-6    2.48e-7      5.14 4.35e- 4
##  9 Americas  Argentina  (Interce~    5.32e+1    3.78e-1    141.   8.10e-18
## 10 Americas  Argentina  pop          5.53e-7    1.28e-8     43.1  1.08e-12
## # ... with 274 more rows
```

Now isn't that beautiful!! :smiley: