Assignment 3 Answer Sketch

432 Staff

Due 2017-02-09. Sketch developed 2018-02-10

Contents


```
library(gapminder)
library(skimr)
library(broom)
library(leaps)
library(modelr)
library(tidyverse)

skim_with(numeric = list(hist = NULL),
          integer = list(hist = NULL)) # drop histograms
```

1 Question 1

Consider the `hbp330` data used in Homeworks 1 and 2. In question 4 of HW2, you built a model for the prediction of body-mass index, considering the following 14 predictors: `practice`, `age`, `race`, `eth_hisp`, `sex`, `insurance`, `income`, `hsgrad`, `tobacco`, `depdiag`, `sbp`, `dbp`, `statin` and `bpmed`. Your task now is to fit a Spearman rho-squared plot to identify the candidate variables on which you might most reasonably try to address non-linearity in a model predicting body-mass index, now making use of as much of the data set that missing data allow. Show the plot, and provide a written explanation of your conclusions about it, and specify the variables that are most appealing for non-linear augmentations, all in complete sentences. Which variables are most appealing candidates to add non-linear evaluations to a linear fit to the complete set of 14 predictors, and why?

1.1 Data Load

```r
hbp330 <- read.csv("data/hbp330.csv") %>% tbl_df
```

1.2 Create `bmi` variable

```r
hbp330 <- hbp330 %>%
  mutate( bmi = weight / (height*height))
```

1.3 Full `hbp330` data (including Missing Values)

```r
hbp330_full <- hbp330 %>%
    select(subject, bmi, practice, age, race, eth_hisp, sex,
                    insurance, income, hsgrad, tobacco,
                    depdiag, sbp, dbp, statin, bpmed)

hbp330_full %>% skim(-subject)
```

```
Skim summary statistics
 n obs: 330
 n variables: 16

Variable type: factor
  variable missing complete   n n_unique
   depdiag       0      330 330        2
  eth_hisp       5      325 330        2
 insurance       0      330 330        4
  practice       0      330 330        2
      race       2      328 330        4
       sex       0      330 330        2
   tobacco       0      330 330        3
                           top_counts ordered
             No: 214, Yes: 116, NA: 0   FALSE
              No: 261, Yes: 64, NA: 5   FALSE
 Med: 134, Med: 130, Com: 53, Uni: 13   FALSE
                A: 180, B: 150, NA: 0   FALSE
  Bla: 180, Whi: 131, Asi: 10, Mul: 7   FALSE
                F: 203, M: 127, NA: 0   FALSE
   nev: 140, for: 117, cur: 73, NA: 0   FALSE

Variable type: integer
 variable missing complete   n     mean       sd  p0   p25  median   p75
      age       0      330 330    55.35    11.53  23    48     57     65
    bpmed       0      330 330     0.66     0.48   0     0      1      1
      dbp       0      330 330    74.75    10.2   41    68     74     82
   hsgrad       0      330 330    81.51    10.66  -2    75     81     89
   income       0      330 330 35243.33 16056.44 100 25600 30600  42475
      sbp       0      330 330   128.37    17.3   84   116    128.5  138
   statin       0      330 330     0.71     0.46   0     0      1      1
   p100
```

```
     77
      1
    106
    100
 147400
    194
      1

Variable type: numeric
 variable missing complete   n  mean   sd    p0   p25 median   p75  p100
      bmi       0      330 330 34.83 8.03 16.73 29.73  33.92 39.18 64.04
```

There are 5 missing `eth_hisp` values and 2 missing `race` values.

1.4 Complete Cases: `hbp330` data after deleting cases with NAs

```r
hbp330_noNA <- hbp330 %>%
    select(subject, bmi, practice, age, race, eth_hisp, sex,
                    insurance, income, hsgrad, tobacco,
                    depdiag, sbp, dbp, statin, bpmed) %>%
    drop_na

dim(hbp330_noNA)
```

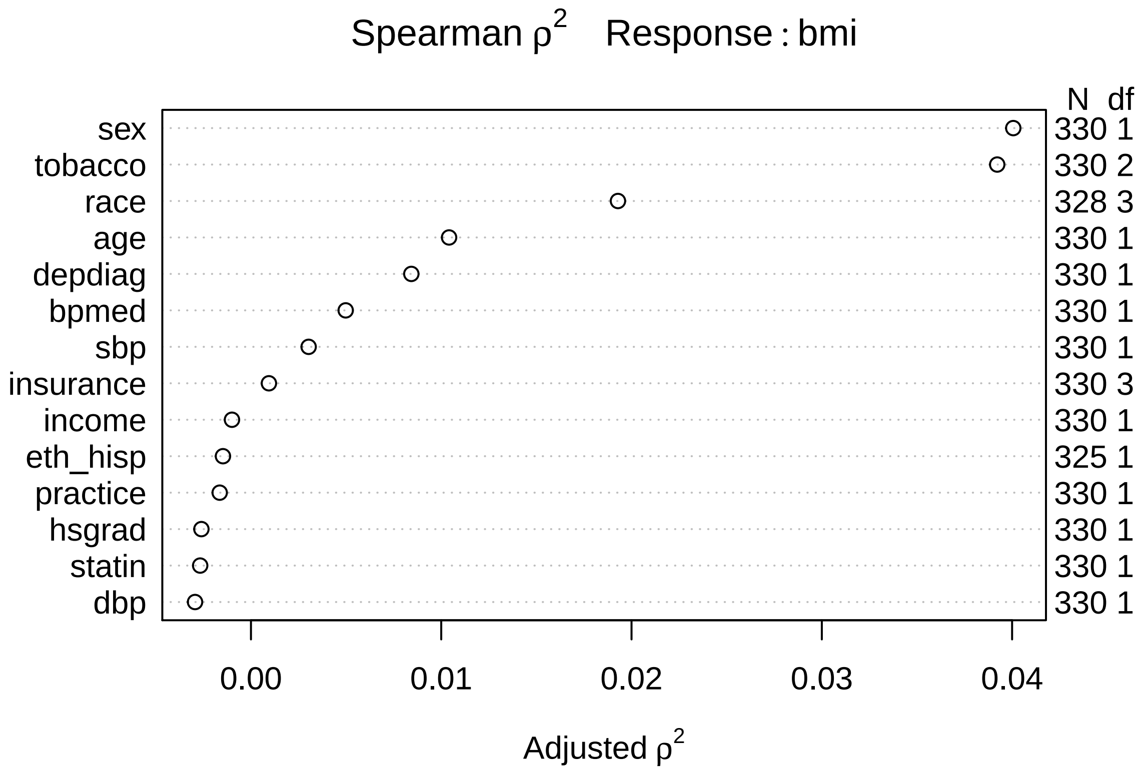
```
[1] 325  16
```

We lose a total of five observations (dropping from 330 to 325 subjects) by dropping missing values.

1.5 Spearman rho-squared plot (applied to full data)

You might have chosen to include all observations, and simply allow the Spearman ρ^2 plot to reduce the sample size for the specific variables (`race` and `eth_hisp`) that had missing values. That produces this result.

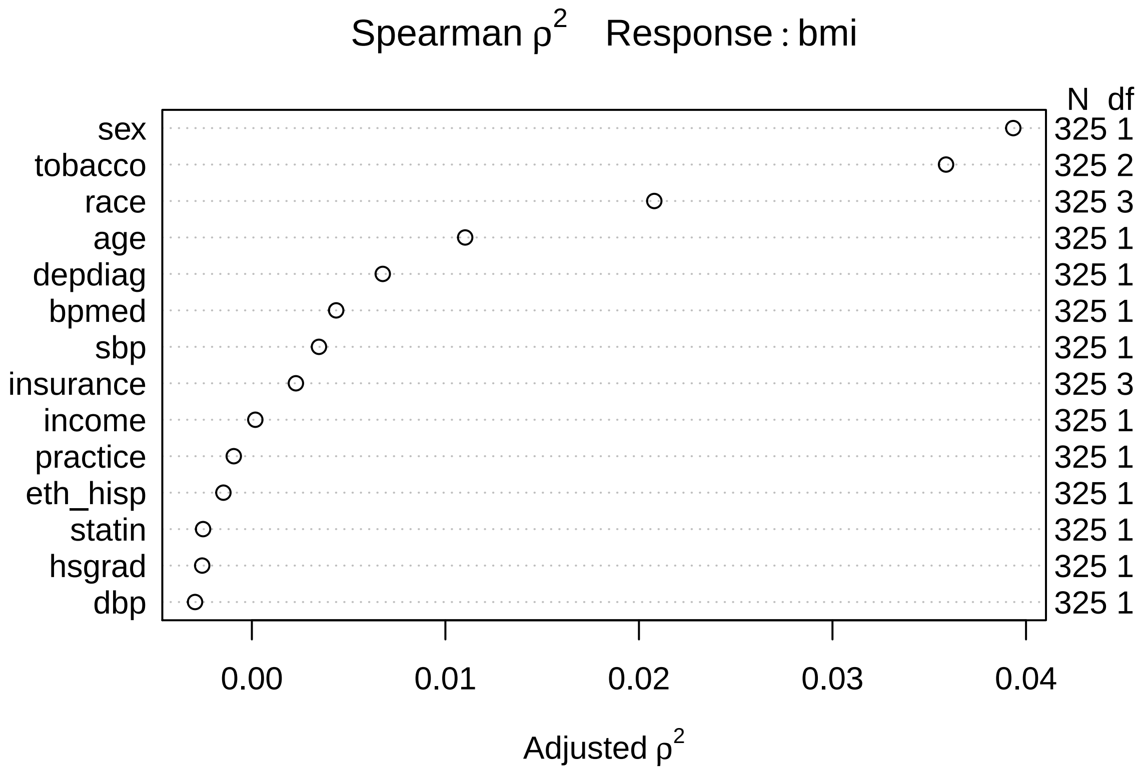
```r
plot(Hmisc::spearman2(bmi ~ practice + age + race +
                          eth_hisp + sex + insurance +
                          income + hsgrad + tobacco +
                          depdiag + sbp + dbp + statin +
                          bpmed, data = hbp330_full))
```



1.6 Spearman rho-squared plot (applied to 325 complete cases)

Or, you might have chosen to include only the 325 complete cases, and so the Spearman ρ^2 plot would address only those subjects without missing `eth_hisp` or `race`. That produces this result.

```
plot(Hmisc::spearman2(bmi ~ practice + age + race +
                          eth_hisp + sex + insurance +
                          income + hsgrad + tobacco +
                          depdiag + sbp + dbp + statin +
                          bpmed, data = hbp330_noNA))
```



In either case, the variables which pack the largest "potential predictive punch" in this setting are, in order, (1) `sex` and (2) `tobacco`. Certainly, those are the most appealing variables for which we should consider non-linear augmentations. Since these are categorical variables, the inclusion of interaction terms seems appealing. We might, for instance, include the `sex-tobacco` interaction or an interaction of `sex` or `tobacco` or both with the next two highest variables on the list: `race` and `age`.

2 Question 2

2.1 The gapminder data

> This assignment uses the `gapminder` library and the `gapminder` data set within. The data set is the work of Jenny Bryan[1], and this assignment is also closely related to assignments she has used in her work. I am deeply indebted to her for this.

2.2 Task 1

> Bring the `gapminder` data in by loading the `gapminder` package. Characterize the data. For instance, how many rows and variables are in the data frame? What do the rows describe?

First, let's glimpse the data to see what we've got:

```
skim(gapminder)
```

[1] Jenny is a great follow on Twitter `@JennyBryan`.

```
Skim summary statistics
 n obs: 1704
 n variables: 6

Variable type: factor
  variable missing complete    n n_unique
 continent       0     1704 1704        5
   country       0     1704 1704      142
                             top_counts ordered
 Afr: 624, Asi: 396, Eur: 360, Ame: 300   FALSE
     Afg: 12, Alb: 12, Alg: 12, Ang: 12   FALSE

Variable type: integer
 variable missing complete    n    mean       sd    p0        p25  median
      pop       0     1704 1704 3e+07    1.1e+08 60011 2793664    7e+06
     year       0     1704 1704  1979.5 17.27     1952    1965.75  1979.5
      p75       p100
 2e+07      1.3e+09
  1993.25 2007

Variable type: numeric
  variable missing complete    n    mean      sd     p0     p25  median
 gdpPercap       0     1704 1704 7215.33 9857.45 241.17 1202.06 3531.85
   lifeExp       0     1704 1704   59.47   12.92  23.6    48.2    60.71
     p75      p100
 9325.46 113523.13
   70.85     82.6
```

Other useful descriptions can be obtained through, for example,

- `gapminder`
- `str(gapminder)`
- `summary(gapminder)`
- `glimpse(gapminder)`
- `Hmisc::describe(gapminder)`

Since the `gapminder` data set is part of an R package, it likely has a help file associated with it, which you can see with the command `?gapminder`

We could take this in any of many directions – now is a good time to be creative. Some key thoughts:

- Each of the 1704 rows represents the characteristics of a single country in a single year.
- Each row contains a set of 6 variables, which specify:
  - `country` = the country's name (142 countries are included)
  - `continent` = the continent in which the country is found (5 continents are included)
  - `year` = the year in which the last three columns of data are relevant (available years range from 1952 to 2007 in increments of five years)
  - `lifeExp` = the life expectancy at birth, in years
  - `pop` = the population of the country
  - `gdpPercap` = gross domestic product per capita

2.3 Task 2

> Pick at least one quantitative and one categorical (factor) variable to explore. What are the possible values? What is a typical value? What is the spread? What is the shape of the distribution, for quantitative variables? What are the levels, for factors?

2.3.1 A "Simple" Approach

Suppose we select the `lifeExp` variable as our quantitative variable, and the `continent` as our factor. We might start with some numeric summaries.

```
gapminder %>%
    select(lifeExp, continent) %>%
    Hmisc::describe()
```

```
.

 2  Variables      1704  Observations
---------------------------------------------------------------------------
lifeExp
       n  missing distinct     Info     Mean      Gmd      .05      .10
    1704        0     1626        1    59.47    14.82    38.49    41.51
     .25      .50      .75      .90      .95
   48.20    60.71    70.85    75.10    77.44

lowest : 23.599 28.801 30.000 30.015 30.331, highest: 81.701 81.757 82.000 82.208 82.603
---------------------------------------------------------------------------
continent
       n  missing distinct
    1704        0        5

Value        Africa Americas     Asia   Europe  Oceania
Frequency       624      300      396      360       24
Proportion    0.366    0.176    0.232    0.211    0.014
---------------------------------------------------------------------------
```

The `describe` function from the `Hmisc` package provides a useful numeric summary for either quantitative or categorical variables.

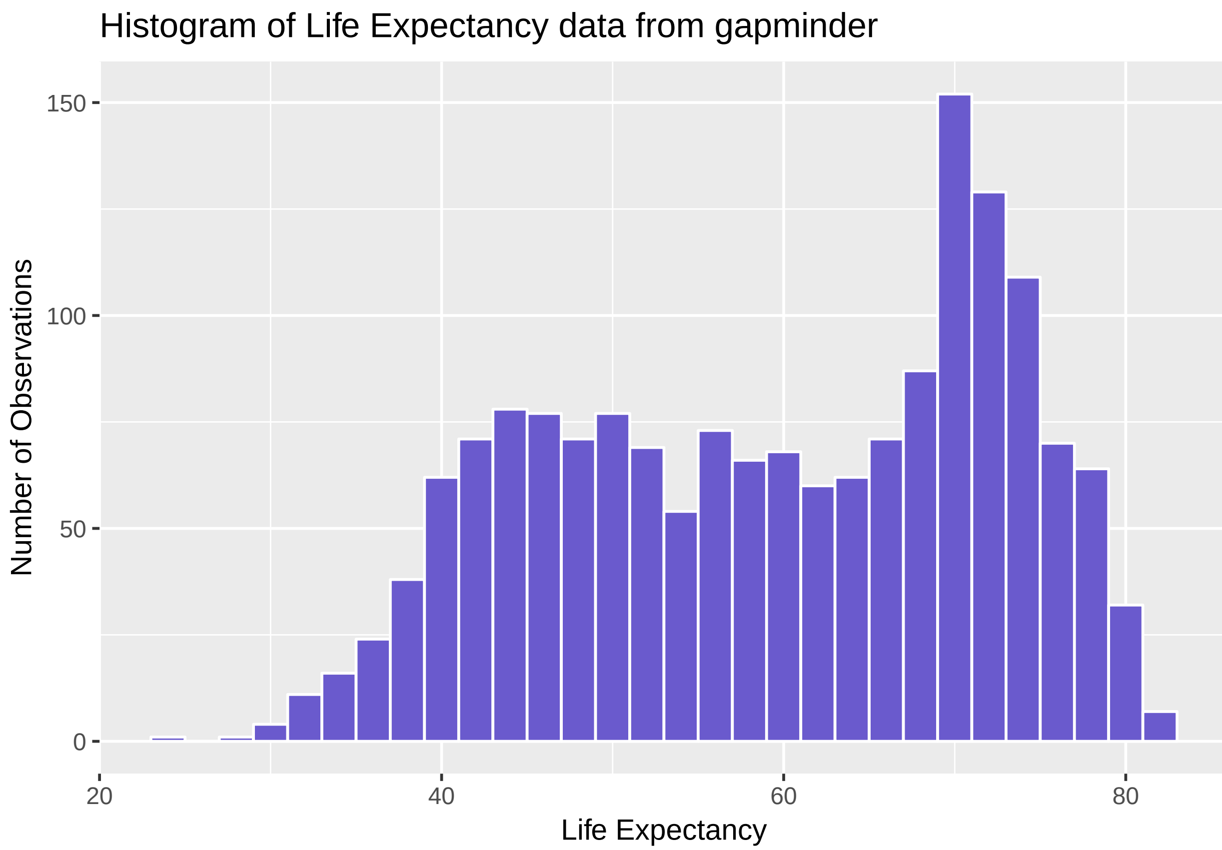
- For `lifeExp`, a quantitative variable, we learn:
    - `n`, the number of observations, as well as the number of `missing` observations, and the number of unique (`distinct`) values.
    - The `Mean` of the data, which is 59.47 years
    - Several quantiles, including the median (from `.50`) and the lower and upper quartiles (from `.25` and `.75`, respectively.) For example, half of the life expectancy values fall below 60.71.
    - The smallest five and large five observations, which is a very useful tool for checking variables.
    - The `Info`, which is related to how continuous the variable is. Values close to 1 indicate that the variable is very continuous, with minimal ties. For more details, see `?Hmisc::describe`.
    - The `Gmd`, which is called the *Gini mean difference*, which is a robust measure of dispersion that is the mean absolute difference between any pairs of observations.
- For a multi-categorical variable (in this case, a factor) like `continent`, `Hmisc::describe` shows:
    - The number of observations, missing observations and distinct values.
    - (if there are fewer than 20 distinct values) a frequency table with counts and proportions.

We might also want to get the standard deviation of `lifeExp`, which turns out to be 12.92 years.

2.3.2 Distribution of your quantitative variable, via `ggplot2`

I also asked about the shape of the distribution for the quantitative variable (in our case, `lifeExp`). Let's draw a histogram.

```r
ggplot(gapminder, aes(x = lifeExp)) +
    geom_histogram(binwidth = 2, fill = "slateblue", color = "white") +
    labs(x = "Life Expectancy", y = "Number of Observations",
         title = "Histogram of Life Expectancy data from gapminder")
```



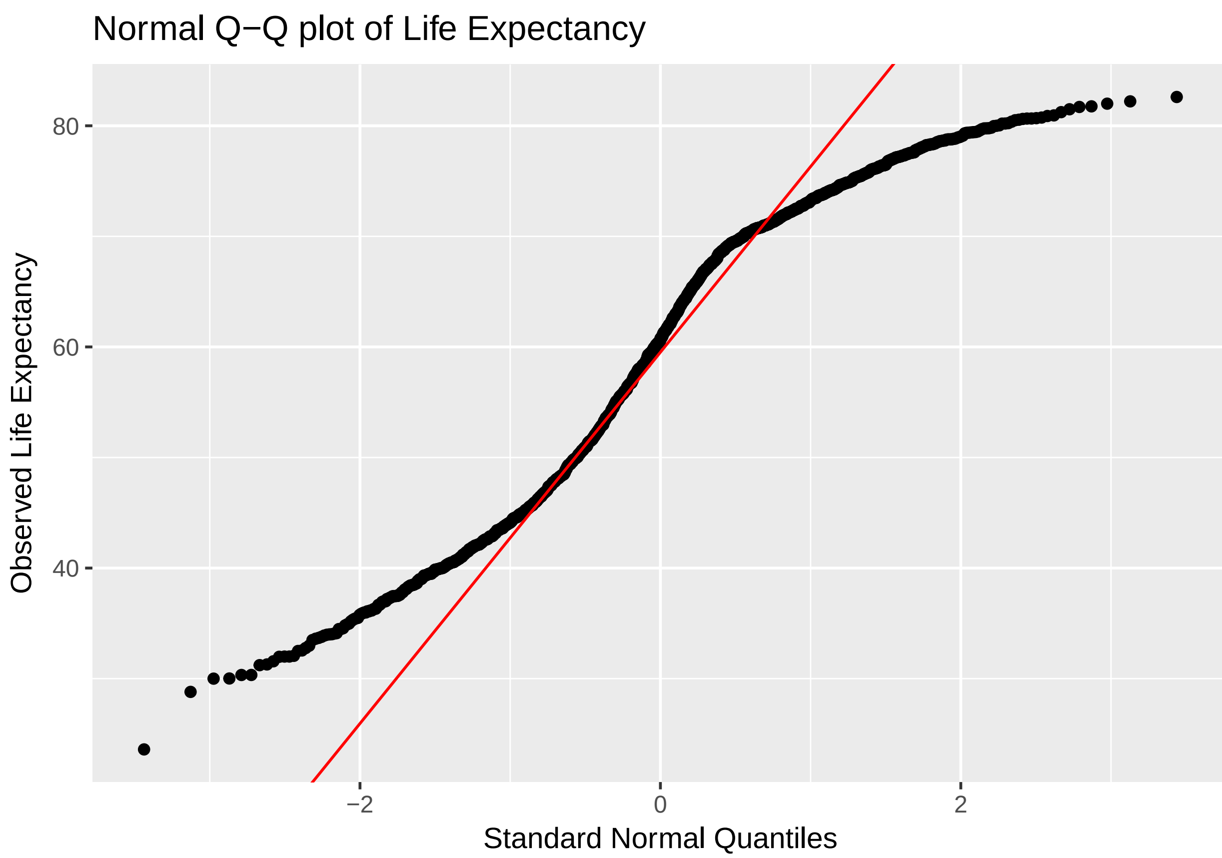
We seem to have a mode near 70, but a largely uniform distribution outside that range, with similar counts for most values between 40 and 80. The choice of `bandwidth = 2` here simply groups the data into bins of approximately 2 years wide. A different choice would reveal a slightly different picture.

The data certainly aren't perfectly approximated by a Normal distribution, but it's hard to say whether the issue is better described as modest skew or by the more detailed description I gave in the previous paragraph. Here's a Normal Q-Q plot to help us thik about this further, built using the approach modeled in our Slides for Class 6.

```r
temp.slo <- diff( quantile(gapminder$lifeExp, c(0.25, 0.75)) ) /
    diff( qnorm(c(0.25, 0.75)) )

temp.int <- quantile(gapminder$lifeExp, c(0.25, 0.75))[1L] -
    temp.slo * qnorm(c(0.25, 0.75))[1L]

ggplot(gapminder, aes(sample = lifeExp)) +
    geom_qq() +
    geom_abline(intercept = temp.int, slope = temp.slo, col = "red") +
    labs(y = "Observed Life Expectancy",
         x = "Standard Normal Quantiles",
         title = "Normal Q-Q plot of Life Expectancy")
```



The S shape in the Normal Q-Q plot suggests that the data are substantially lighter tailed compared to a Normal distribution.

2.4 Task 3

> Build a plot of a quantitative variable, and another plot of a quantitative and a categorical variable, for one or more subsets of the data that interest you. Use `dplyr` functions to create a subset or two you want to plot. Explore more than one plot type and try to use more than one `geom` in your work.

We'll start by looking at Life Expectancy (a quantitative variable) across all nations in 1977.

```
gapminder_1977 <-
    gapminder %>%
    filter(year == 1977)

gapminder_1977
```

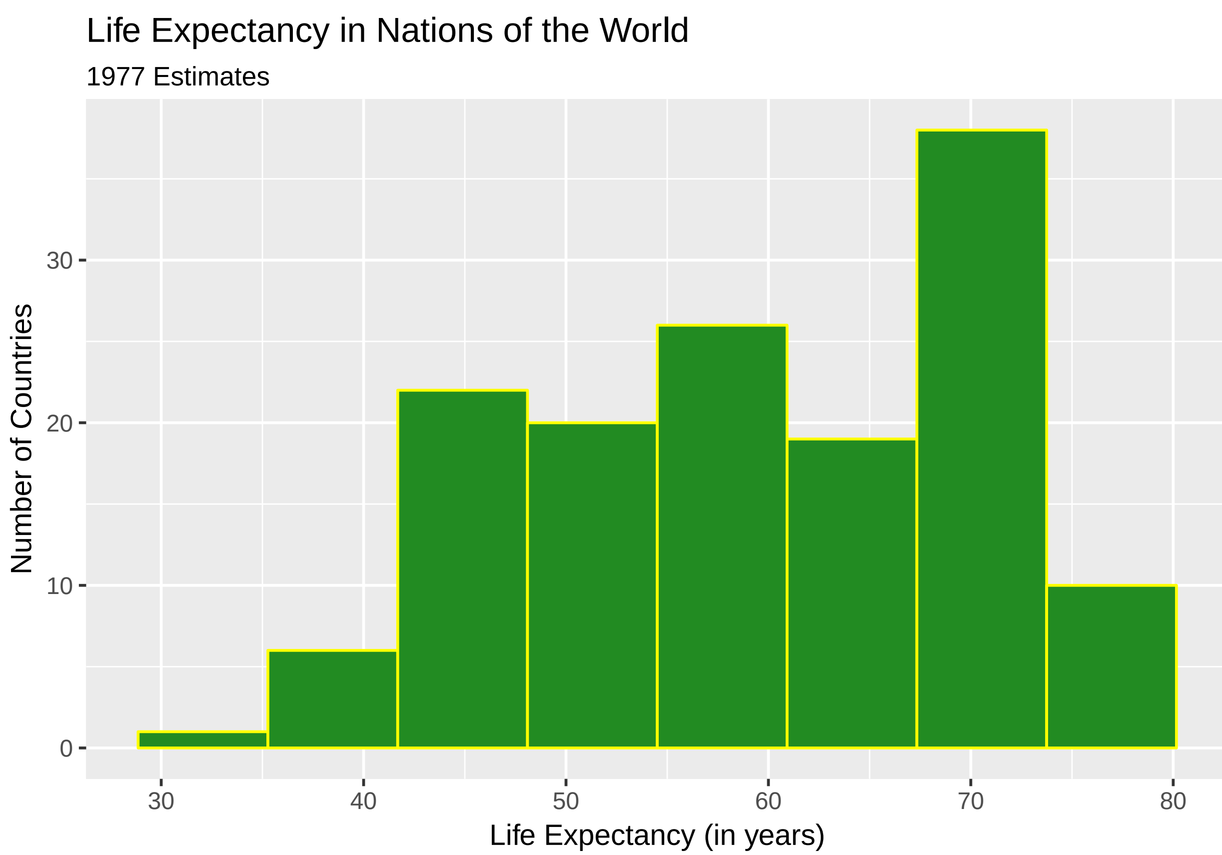
```
# A tibble: 142 x 6
   country     continent  year lifeExp      pop gdpPercap
   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
 1 Afghanistan Asia       1977    38.4 14880372       786
 2 Albania     Europe     1977    68.9  2509048      3533
 3 Algeria     Africa     1977    58.0 17152804      4910
 4 Angola      Africa     1977    39.5  6162675      3009
 5 Argentina   Americas   1977    68.5 26983828     10079
 6 Australia   Oceania    1977    73.5 14074100     18334
```

```
 7 Austria     Europe     1977     72.2  7568430      19749
 8 Bahrain     Asia       1977     65.6   297410      19340
 9 Bangladesh  Asia       1977     46.9 80428306        660
10 Belgium     Europe     1977     72.8  9821800      19118
# ... with 132 more rows
```

2.4.1 Plotting a Quantitative Variable in a Subgroup of Interest

And now, we'll plot the histogram of these life expectancy estimates. Clearly the default choice of 30 bins is too many in this setting, since the resulting plot with 30 bins isn't smooth at all. The plot below uses just 8 bins.

```r
ggplot(gapminder_1977, aes(x = lifeExp)) +
    geom_histogram(bins = 8, col = "yellow", fill = "forestgreen") +
    labs(x = "Life Expectancy (in years)",
         y = "Number of Countries",
         title = "Life Expectancy in Nations of the World",
         subtitle = "1977 Estimates")
```



2.4.2 Plotting a Quantitative Variable and a Categorical Variable in a Subgroup of Interest

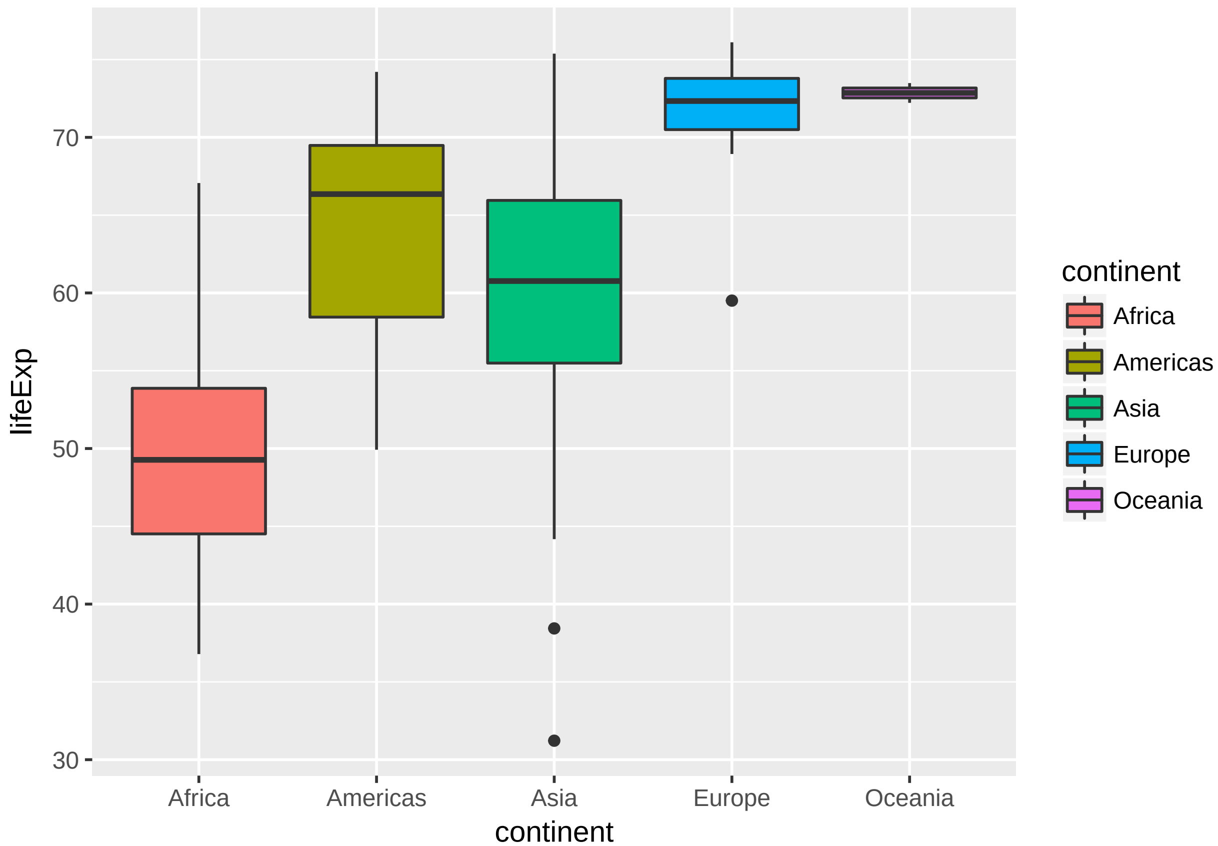
Let's build a plot of `lifeExp` (quantitative) stratified by `continent` (categorical) using, again, the 1977 data. First, build the data we'll need, using `dplyr` operations:

```
gapminder_1977_a <- gapminder %>%
    filter(year == 1977) %>%
    select(year, country, continent, lifeExp)
```

And now, we plot the results in two ways...

2.4.3 A boxplot

```
ggplot(gapminder_1977_a, aes(x = continent, y = lifeExp,
                             fill = continent)) +
    geom_boxplot()
```



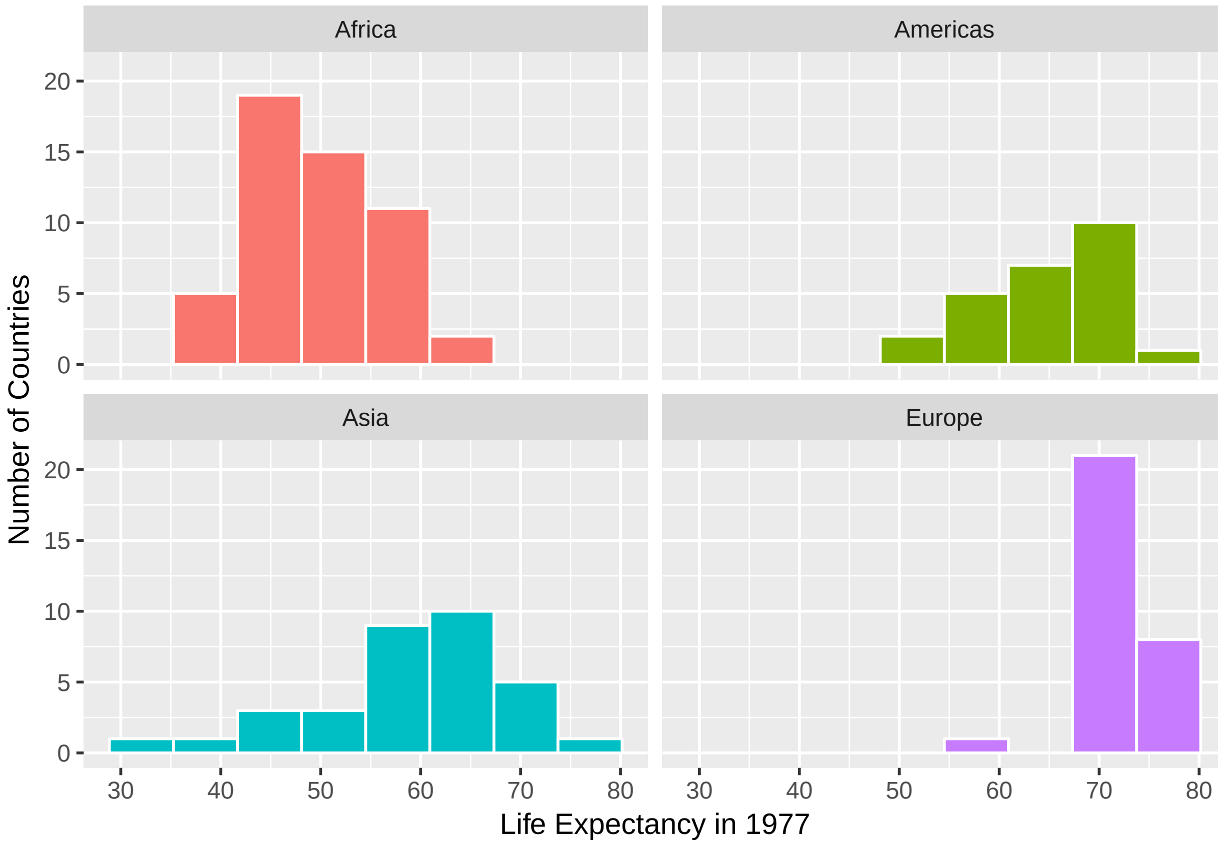
2.4.4 Using Facets to build Multiple Histograms in One Plot

We'll drop Oceania here, since it's only a couple of countries, and so a histogram (like the boxplot above) seems silly. We'll also drop the legend, since it's repetitive.

```
gapminder_1977_b <- filter(gapminder_1977_a, continent != "Oceania")

ggplot(gapminder_1977_b, aes(x = lifeExp, fill = continent)) +
    geom_histogram(bins = 8, color = "white") +
    guides(fill = FALSE) + ## leaves out the legend
    facet_wrap(~ continent, nrow = 2) +
```

```
    labs(x = "Life Expectancy in 1977",
         y = "Number of Countries")
```



2.5 Task 4

> Make a scatterplot which shows the relationship between two quantitative variables, either for a clearly specified subset of interest, or for the data as a whole. Include a regression line in the plot. Make an active choice as to which variable should be the predictor and which the outcome.

We'll use per-capita gross domestic product (`gdpPercap`) as a predictor of an outcome, `lifeExp,` here, for all countries in the continents of Africa and the Americas, in the most recent available data period (2007).

First, I'll create the data set with the pieces I need.

```
gap_4 <- gapminder %>%
    filter(year == 2007) %>%
    filter(continent %in% c("Africa", "Americas")) %>%
    select(country, year, continent, gdpPercap, lifeExp)
gap_4
```

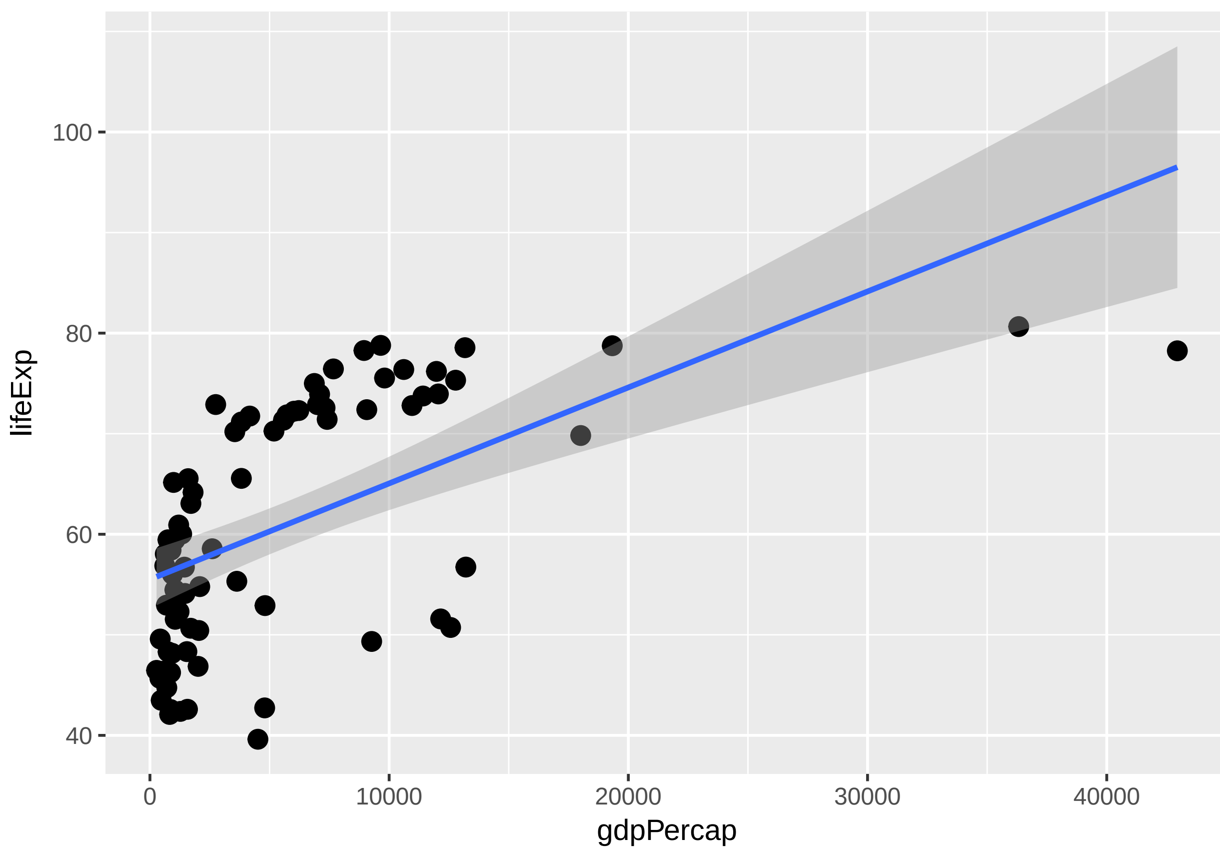
```
# A tibble: 77 x 5
   country       year continent gdpPercap lifeExp
   <fct>        <int> <fct>         <dbl>   <dbl>
 1 Algeria       2007 Africa         6223    72.3
 2 Angola        2007 Africa         4797    42.7
 3 Argentina     2007 Americas      12779    75.3
```

```
 4 Benin         2007 Africa         1441    56.7
 5 Bolivia       2007 Americas       3822    65.6
 6 Botswana      2007 Africa        12570    50.7
 7 Brazil        2007 Americas       9066    72.4
 8 Burkina Faso  2007 Africa         1217    52.3
 9 Burundi       2007 Africa          430    49.6
10 Cameroon      2007 Africa         2042    50.4
# ... with 67 more rows
```

2.5.1 A scatterplot ignoring the continent information

Now, I'll build a plot using the `gap_4` data, and ignoring the continent information.

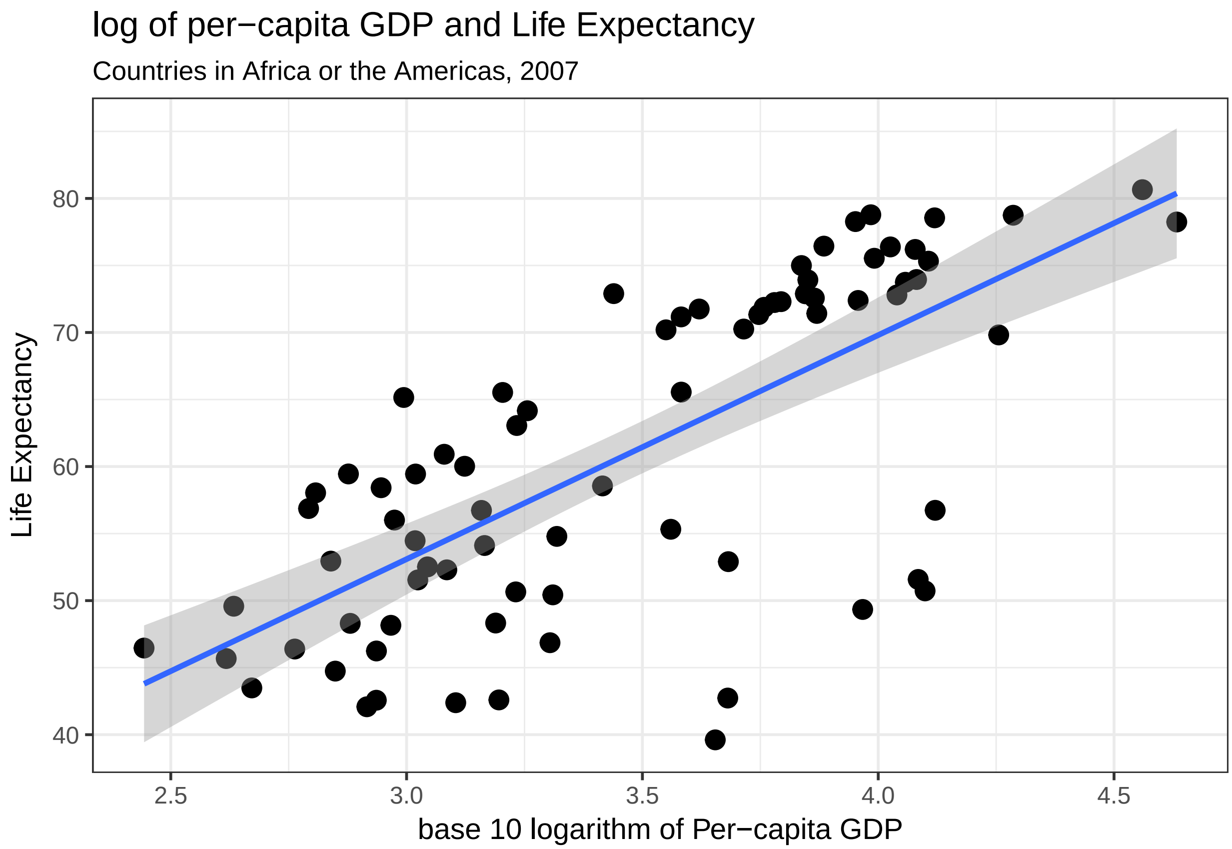
```
ggplot(gap_4, aes(x = gdpPercap, y = lifeExp)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm")
```



Typically, `gdpPercap` is represented in Rosling's gapminder presentations on the log scale. Let's try a base 10 logarithm here for our predictor, and add some titles, while switching themes.

```
ggplot(gap_4, aes(x = log10(gdpPercap), y = lifeExp)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm") +
    theme_bw() +
    labs(x = "base 10 logarithm of Per-capita GDP",
         y = "Life Expectancy",
```

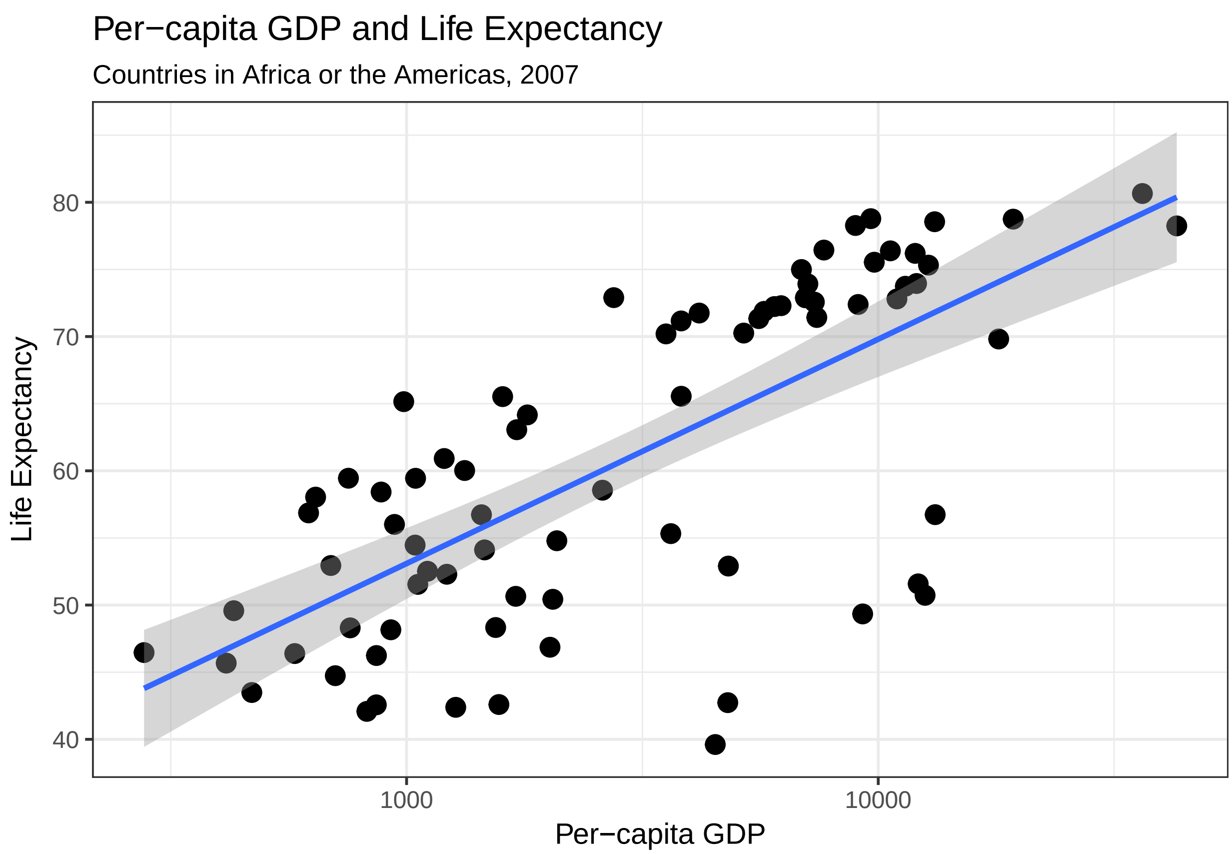
```
        title = "log of per-capita GDP and Life Expectancy",
        subtitle = "Countries in Africa or the Americas, 2007")
```



2.5.2 Rescaling the axis to the log

Or, instead of plotting the log, we could do something better still, and just plot the data on a log (base 10) scale, using the `scale_x_log10` function, which results in a plot that is far easier to interpret. Let's do that for the x axis here.

```
ggplot(gap_4, aes(x = gdpPercap, y = lifeExp)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm") +
    scale_x_log10() +
    theme_bw() +
    labs(x = "Per-capita GDP",
        y = "Life Expectancy",
        title = "Per-capita GDP and Life Expectancy",
        subtitle = "Countries in Africa or the Americas, 2007")
```



2.6 Task 5

> Specify the regression equation fitted for Task 4, and evaluate it using summary statistics like R^2 or the residual standard deviation, as well as through assessments of the direction and size of the coefficient estimates.

The equation for Task 4 involves one predictor (per-capita GDP, or actually its base 10 logarithm looks better) and one outcome (life expectancy). The model is:

```
mod4 <- lm(lifeExp ~ log10(gdpPercap), data = gap_4)
summary(mod4)
```

```

Call:
lm(formula = lifeExp ~ log10(gdpPercap), data = gap_4)

Residuals:
    Min      1Q  Median      3Q     Max 
-24.418  -4.254   2.730   6.117  12.466 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)         2.963      6.694   0.443    0.659    
log10(gdpPercap)   16.710      1.910   8.750 4.45e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 8.586 on 75 degrees of freedom
Multiple R-squared:  0.5052,    Adjusted R-squared:  0.4986
F-statistic: 76.57 on 1 and 75 DF,  p-value: 4.447e-13
```

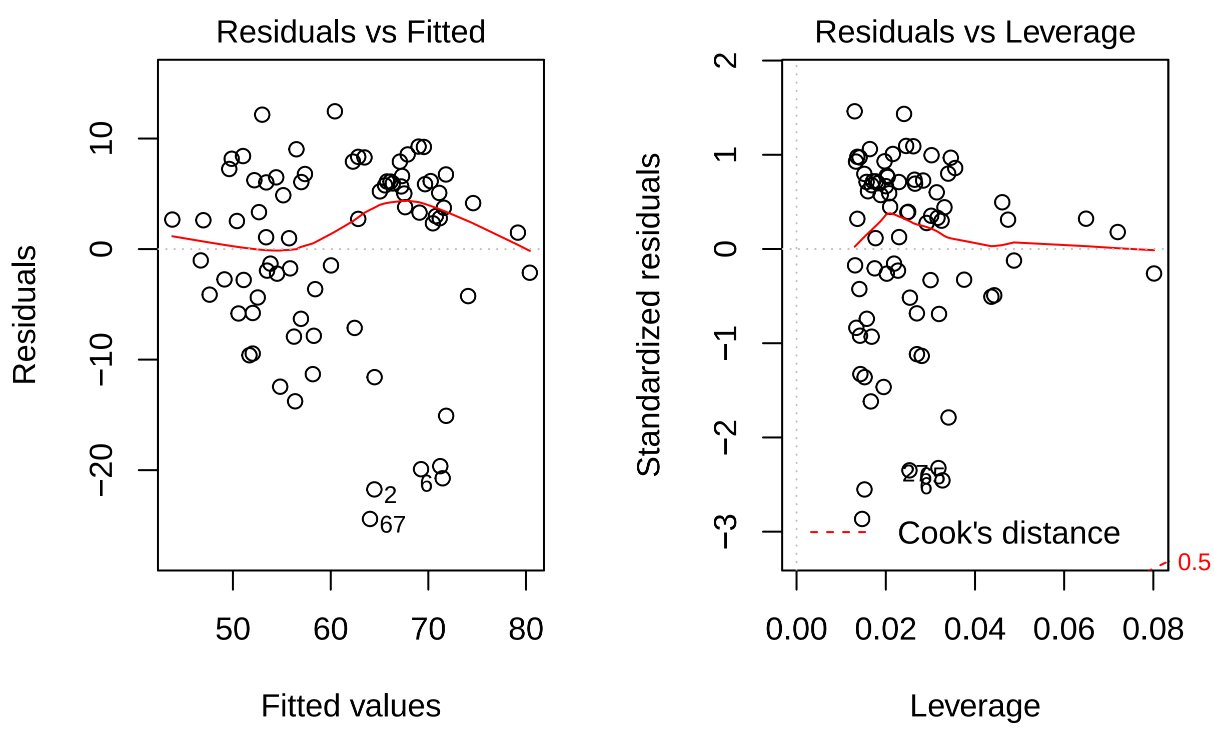
If you ran the model on `gdpPercap` without the logarithm, you would have to increase the number of digits in our model's `display` to get anything useful in the `gdpPercap` variables.

A strong assessment here would say something about:

- the direction of the relationship (an increasing relationship between log of `gdpPercap` and `lifeExp`,)
- the fact that both predictors in this model are statistically significantly different from 0 at the 5% level,
- the summary statistics R^2 which is 0.51 which is at best moderately strong, and the residual SD, $\hat{\sigma}$, which is 8.6, so that a typical error in predicting life Expectancy should be less than $2(8.6) = 17.2$ years, which doesn't actually sound all that great.

You could look at residual plots here. That wasn't something we saw as vital. The log scale/transformation for `gdpPercap` definitely seems to help produce a more linear relationship, reigning in some outliers.

```
par(mfrow = c(1,2))
plot(mod4, which = c(1,5))
```




```
par(mfrow = c(1,1))
```

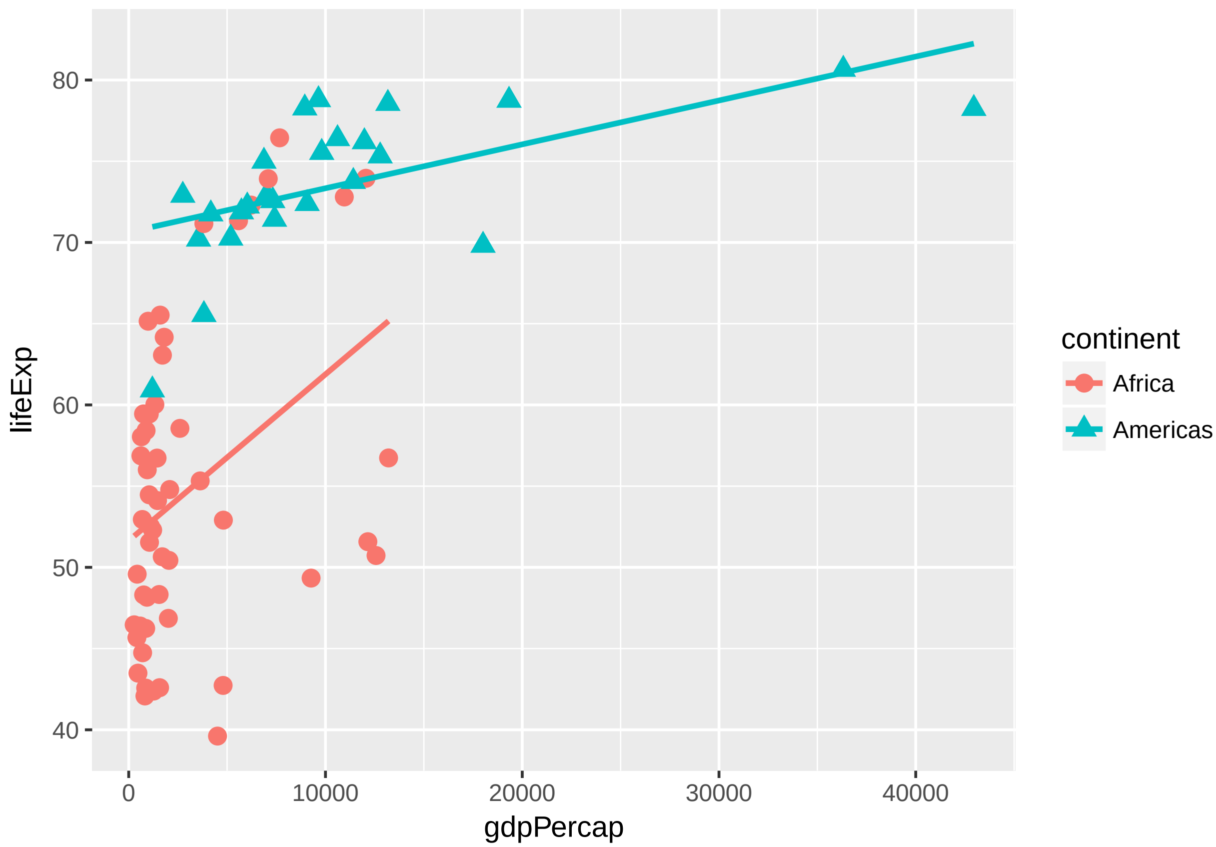
2.7 Task 6

Augment your scatterplot from Task 4 by incorporating a categorical variable as well as a predictor. Show the new predictor in a useful way as part of the plot, and show the regression model incorporating the predictor as part of the plot.

We'll use per-capita gross domestic product (`gdpPercap`) as a predictor of an outcome, `lifeExp`, here, for all countries in the continents of Africa and the Americas, in the most recent available data period (2007), and use the `continent` as the other predictor.

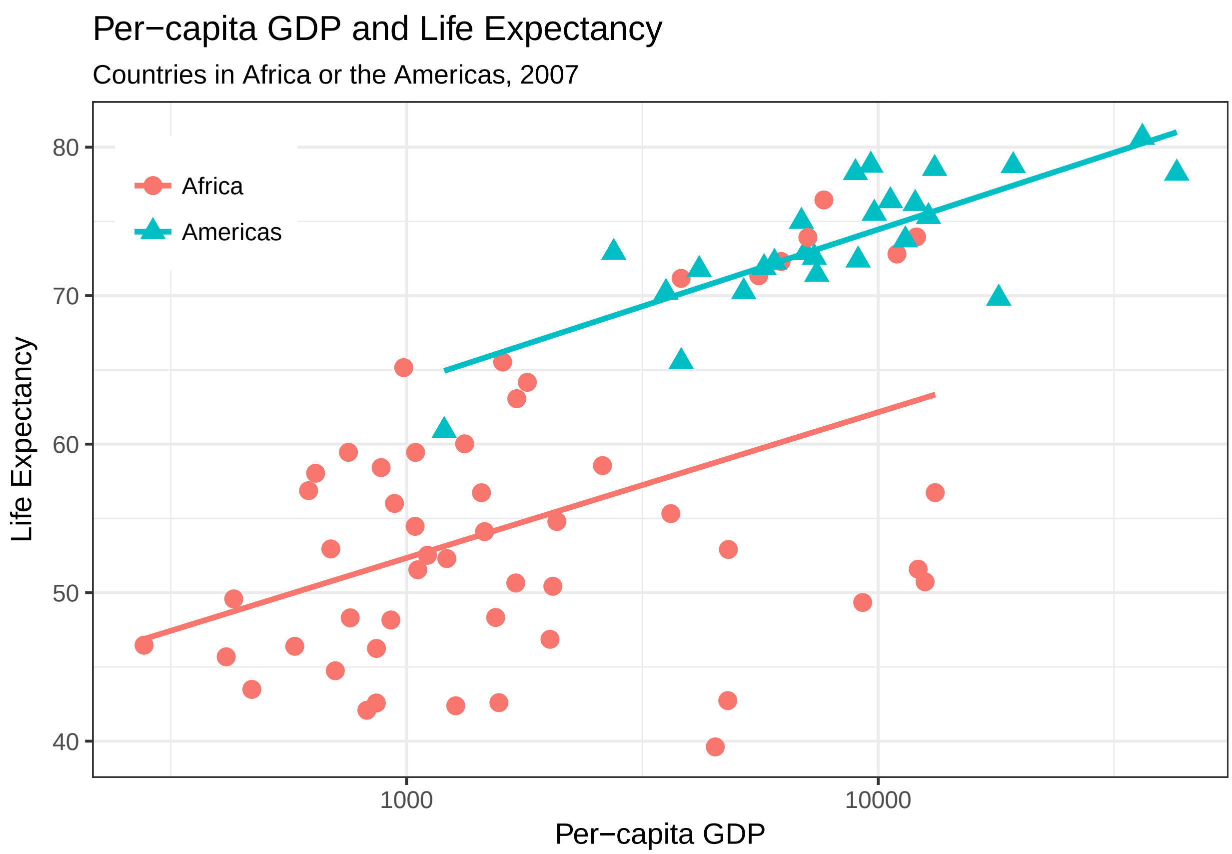
I'll use the same data set I created in Task 4, called `gap_4`. Here's my first attempt at a plot.

```
ggplot(gap_4, aes(x = gdpPercap, y = lifeExp, color = continent, shape = continent)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm", se = FALSE)
```



As we've seen, `gdpPercap` is represented in Rosling's gapminder presentations on the log scale. Here, we plot the data on a log (base 10) scale, using the `scale_x_log10` function, which results in a plot that is far easier to interpret. Let's do that for the x axis here, and also move the legend to the top left of the plot area, while dropping its title (continent).

```
ggplot(gap_4, aes(x = gdpPercap, y = lifeExp, color = continent, shape = continent)) +
    geom_point(size = 3) +
    geom_smooth(method = "lm", se = FALSE) +
    scale_x_log10() +
    theme_bw() +
    theme(legend.position = c(0.1, 0.85)) +
    guides(color = guide_legend(title = NULL), shape = guide_legend(title = NULL)) +
    labs(x = "Per-capita GDP",
         y = "Life Expectancy",
         title = "Per-capita GDP and Life Expectancy",
         subtitle = "Countries in Africa or the Americas, 2007")
```



2.8 Task 7

> Specify the regression equation from Task 6, and describe the nature of the relationship between the predictors, and their impact on your outcome of interest through useful summaries.

The equation for Task 6 involves two predictors (continent, and base-10 log of per-capita GDP) and one outcome (life expectancy). The model is:

```
mod6 <- lm(lifeExp ~ log10(gdpPercap) * continent, data = gap_4)
summary(mod6)
```

```
Call:
lm(formula = lifeExp ~ log10(gdpPercap) * continent, data = gap_4)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.1483  -4.0185  -0.0751   4.2449  15.4213 

Coefficients:
                                   Estimate Std. Error t value Pr(>|t|)
(Intercept)                         22.9068     7.6253   3.004  0.00365
log10(gdpPercap)                     9.8110     2.3242   4.221  6.9e-05
continentAmericas                   10.1583    19.2143   0.529  0.59863
log10(gdpPercap):continentAmericas   0.5369     5.0521   0.106  0.91565
```

```
(Intercept)                        **
log10(gdpPercap)                   ***
continentAmericas
log10(gdpPercap):continentAmericas
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.357 on 73 degrees of freedom
Multiple R-squared:  0.6463,    Adjusted R-squared:  0.6318
F-statistic: 44.47 on 3 and 73 DF,  p-value: < 2.2e-16
```

So the models are:

`Continent`	Model for `lifeExp`
Africa	22.91 + 9.81 log_10_(`gdpPercap`)
Americas	(22.91 + 10.16) + (9.81 + 0.54) $\log_{10}$(`gdpPercap`), which is 33.07 + 10.35 $\log_{10}$(`gdpPercap`)

A strong assessment here would say something about:

- the shift between the two continents, with the Americas living an additional 10 years (comparing the intercepts, since the slopes are similar) as compared to African residents with the same value of Per-Capita GDP, on average,
- the direction of the relationships in the two continents (both show an increasing relationship between `gdpPercap` and `lifeExp`, with a slightly larger slope in the Americas,)
- the fact that the two main effects are statistically significant at the 5% level, but not the interaction term,
- the summary statistics: R^2 which is 0.61 and seems fairly strong, and the residual SD, $\hat{\sigma}$, which is 7.7, so that a typical error in predicting life Expectancy should be less than 15.4 years.

2.9 Task 8

> Now, report your process. Write a brief essay (at least 75 words but probably not much more than 150) reflecting on what was hard/easy, problems you solved, helpful tutorials you read, etc. What things were hard, even though you saw them in class? What was easy (or sort of easy) even though we haven't done it in class?*

A good essay (and **this is the part we'll be reviewing most closely**) will address the questions well. We don't write answer sketches for essay questions.