cm014 Worksheet: The Model-Fitting Paradigm in R

```r
suppressPackageStartupMessages(library(tidyverse))
library(gapminder)
library(broom)
```

So you want to fit a model to your data. How can you achieve this with R?

Topics:

1. What *is* model-fitting?
2. How do we fit a model in R?
3. How can we obtain tidy results from the model output?

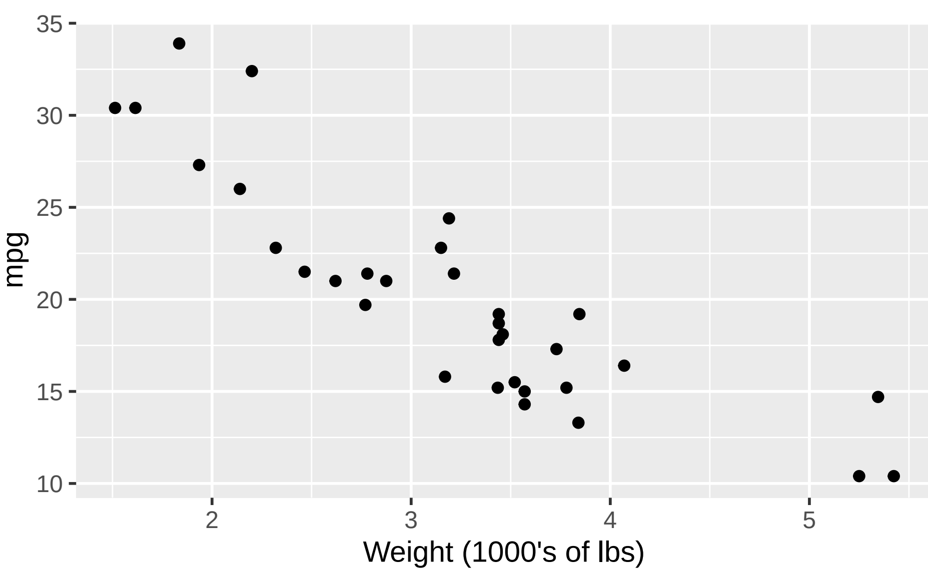
What is Model-Fitting?

When variables are not independent, then we can gain information about one variable if we know something about the other.

Examples: Use the scatterplot below:

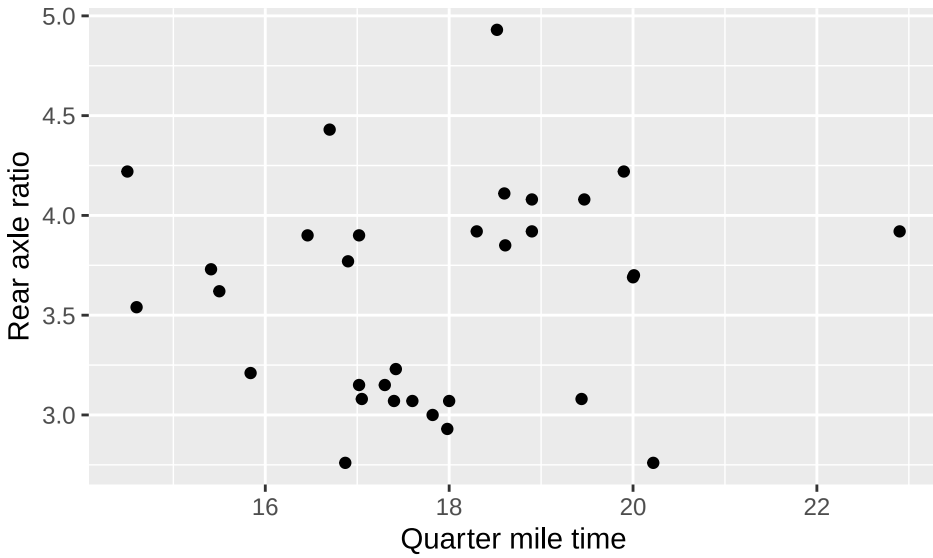
1. A car weighs 4000 lbs. What can we say about its mpg?
2. A car weights less than 3000 lbs. What can we say about its mpg?

```r
library(tidyverse)
ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  labs(x = "Weight (1000's of lbs)")
```



Example: What can we say about rear axle ratio if we know something about quarter mile time?

```r
ggplot(mtcars, aes(qsec, drat)) +
  geom_point() +
  labs(x = "Quarter mile time",
       y = "Rear axle ratio")
```



If EDA isn't enough, we can answer these questions by fitting a model: a curve that predicts Y given X. Aka, a **regression curve** or a **machine learning model**.

(There are more comprehensive models too, such as modelling entire distributions, but that's not what we're doing here)

There are typically two goals of fitting a model:

1. Make predictions.
2. Interpret variable relationships.

Fitting a model in R

Model fitting methods tend to use a common format in R:

```
method(formula, data, options)
```

They also tend to have a common output: a special *list*.

Method:

A function such as:

- Linear Regression: `lm`
- Generalized Linear Regression: `glm`
- Local regression: `loess`
- Quantile regression: `quantreg::rq`
- ...

Formula:

In R, takes the form `y ~ x1 + x2 + ... + xp` (use column names in your data frame).

Data: The data frame.

Options: Specific to the method.

Exercise:

1. Fit a linear regression model to life expectancy ("Y") from year ("X") by filling in the formula. Notice what appears as the output.
2. On a new line, use the `unclass` function to uncover the object's true nature: a list. Note: it might be easier to use the `names` function to see what components are included in the list.

First, create a subset of the `gapminder` dataset containing only the country of 'France

```
gapminder_France <- gapminder%>%
  filter(country=="France")

gapminder_France
```

```
## # A tibble: 12 x 6
##    country continent  year lifeExp      pop gdpPercap
##    <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
##  1 France  Europe     1952    67.4 42459667     7030.
##  2 France  Europe     1957    68.9 44310863     8663.
##  3 France  Europe     1962    70.5 47124000    10560.
##  4 France  Europe     1967    71.6 49569000    13000.
##  5 France  Europe     1972    72.4 51732000    16107.
##  6 France  Europe     1977    73.8 53165019    18293.
##  7 France  Europe     1982    74.9 54433565    20294.
##  8 France  Europe     1987    76.3 55630100    22066.
##  9 France  Europe     1992    77.5 57374179    24704.
## 10 France  Europe     1997    78.6 58623428    25890.
## 11 France  Europe     2002    79.6 59925035    28926.
## 12 France  Europe     2007    80.7 61083916    30470.
```

Now, using the `lm()` function we will create the linear model

```
(my_lm <- lm(lifeExp~year, data=gapminder_France))
```

```
##
## Call:
## lm(formula = lifeExp ~ year, data = gapminder_France)
##
## Coefficients:
## (Intercept)         year
##   -397.7646       0.2385
```

Does that mean that the life expectency at "year 0" was equal to -397.7646?! We are interested in the modeling results around the modeling period which starts at year 1952. To get a meaningful "interpretable" intercept we can use the `I()` function.

```r
(my_lm <- lm(lifeExp~I(year-min(year)), data=gapminder_France))
```

```
##
## Call:
## lm(formula = lifeExp ~ I(year - min(year)), data = gapminder_France)
##
## Coefficients:
##         (Intercept)  I(year - min(year))
##             67.7901               0.2385
```

Use the `unclass()` function to take a look at how the `lm()` object actually looks like.

```r
unclass(my_lm)
```

```
## $coefficients
##         (Intercept) I(year - min(year))
##          67.7901282           0.2385014
##
## $residuals
##           1           2           3           4           5           6
## -0.38012821 -0.05263520  0.33485781  0.18235082 -0.18015618  0.07733683
##           7           8           9          10          11          12
## -0.05517016  0.20232284  0.12981585  0.11730886 -0.12519814 -0.25070513
##
## $effects
##         (Intercept) I(year - min(year))
##       -257.55220231         14.26030956          0.41516662
##
##          0.26479522         -0.09557618          0.16405242
##
##          0.03368103          0.29330963          0.22293823
##
##          0.21256684         -0.02780456         -0.15117596
##
## $rank
## [1] 2
##
## $fitted.values
##        1        2        3        4        5        6        7        8
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##        9       10       11       12
## 77.33018 78.52269 79.71520 80.90771
##
## $assign
## [1] 0 1
##
## $qr
## $qr
##    (Intercept) I(year - min(year))
## 1   -3.4641016        -95.26279442
## 2    0.2886751         59.79130372
## 3    0.2886751          0.18965544
```

```
## 4    0.2886751          0.10603124
## 5    0.2886751          0.02240704
## 6    0.2886751         -0.06121716
## 7    0.2886751         -0.14484136
## 8    0.2886751         -0.22846557
## 9    0.2886751         -0.31208977
## 10   0.2886751         -0.39571397
## 11   0.2886751         -0.47933817
## 12   0.2886751         -0.56296237
## attr(,"assign")
## [1] 0 1
##
## $qraux
## [1] 1.288675 1.273280
##
## $pivot
## [1] 1 2
##
## $tol
## [1] 1e-07
##
## $rank
## [1] 2
##
## attr(,"class")
## [1] "qr"
##
## $df.residual
## [1] 10
##
## $xlevels
## named list()
##
## $call
## lm(formula = lifeExp ~ I(year - min(year)), data = gapminder_France)
##
## $terms
## lifeExp ~ I(year - min(year))
## attr(,"variables")
## list(lifeExp, I(year - min(year)))
## attr(,"factors")
##                     I(year - min(year))
## lifeExp                               0
## I(year - min(year))                   1
## attr(,"term.labels")
## [1] "I(year - min(year))"
## attr(,"order")
## [1] 1
## attr(,"intercept")
## [1] 1
## attr(,"response")
## [1] 1
## attr(,".Environment")
## <environment: R_GlobalEnv>
```

```
## attr(,"predvars")
## list(lifeExp, I(year - min(year)))
## attr(,"dataClasses")
##             lifeExp I(year - min(year))
##           "numeric"           "numeric"
##
## $model
##    lifeExp I(year - min(year))
## 1   67.410                   0
## 2   68.930                   5
## 3   70.510                  10
## 4   71.550                  15
## 5   72.380                  20
## 6   73.830                  25
## 7   74.890                  30
## 8   76.340                  35
## 9   77.460                  40
## 10  78.640                  45
## 11  79.590                  50
## 12  80.657                  55
```

To complicate things further, some info is stored in *another* list after applying the `summary` function:

```r
summary(my_lm)
```

```
##
## Call:
## lm(formula = lifeExp ~ I(year - min(year)), data = gapminder_France)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38013 -0.13894  0.01235  0.14295  0.33486
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         67.79013    0.11949  567.33  < 2e-16 ***
## I(year - min(year))  0.23850    0.00368   64.81 1.86e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.22 on 10 degrees of freedom
## Multiple R-squared:  0.9976, Adjusted R-squared:  0.9974
## F-statistic:  4200 on 1 and 10 DF,  p-value: 1.863e-14
```

We can use the `predict()` function to make predictions from the model (default is to use fitting/training data). Here are the predictions:

```r
predict(my_lm) %>%
  head()
```

```
##        1        2        3        4        5        6
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266
```

Or we can predict on a new dataset:

```r
years1 = gapminder
predict(my_lm,years1)
```

```
##        1        2        3        4        5        6        7        8 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##        9       10       11       12       13       14       15       16 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##       17       18       19       20       21       22       23       24 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##       25       26       27       28       29       30       31       32 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##       33       34       35       36       37       38       39       40 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##       41       42       43       44       45       46       47       48 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##       49       50       51       52       53       54       55       56 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##       57       58       59       60       61       62       63       64 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##       65       66       67       68       69       70       71       72 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##       73       74       75       76       77       78       79       80 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##       81       82       83       84       85       86       87       88 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##       89       90       91       92       93       94       95       96 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##       97       98       99      100      101      102      103      104 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##      105      106      107      108      109      110      111      112 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##      113      114      115      116      117      118      119      120 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##      121      122      123      124      125      126      127      128 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##      129      130      131      132      133      134      135      136 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##      137      138      139      140      141      142      143      144 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##      145      146      147      148      149      150      151      152 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##      153      154      155      156      157      158      159      160 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##      161      162      163      164      165      166      167      168 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##      169      170      171      172      173      174      175      176 
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768 
##      177      178      179      180      181      182      183      184 
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765 
##      185      186      187      188      189      190      191      192 
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771 
##      193      194      195      196      197      198      199      200 
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      201      202      203      204      205      206      207      208
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      209      210      211      212      213      214      215      216
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      217      218      219      220      221      222      223      224
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      225      226      227      228      229      230      231      232
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      233      234      235      236      237      238      239      240
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      241      242      243      244      245      246      247      248
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      249      250      251      252      253      254      255      256
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      257      258      259      260      261      262      263      264
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      265      266      267      268      269      270      271      272
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      273      274      275      276      277      278      279      280
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      281      282      283      284      285      286      287      288
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      289      290      291      292      293      294      295      296
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      297      298      299      300      301      302      303      304
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      305      306      307      308      309      310      311      312
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      313      314      315      316      317      318      319      320
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      321      322      323      324      325      326      327      328
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      329      330      331      332      333      334      335      336
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      337      338      339      340      341      342      343      344
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      345      346      347      348      349      350      351      352
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      353      354      355      356      357      358      359      360
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      361      362      363      364      365      366      367      368
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      369      370      371      372      373      374      375      376
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      377      378      379      380      381      382      383      384
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      385      386      387      388      389      390      391      392
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      393      394      395      396      397      398      399      400
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      401      402      403      404      405      406      407      408
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      409      410      411      412      413      414      415      416
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      417      418      419      420      421      422      423      424
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      425      426      427      428      429      430      431      432
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      433      434      435      436      437      438      439      440
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      441      442      443      444      445      446      447      448
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      449      450      451      452      453      454      455      456
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      457      458      459      460      461      462      463      464
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      465      466      467      468      469      470      471      472
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      473      474      475      476      477      478      479      480
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      481      482      483      484      485      486      487      488
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      489      490      491      492      493      494      495      496
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      497      498      499      500      501      502      503      504
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      505      506      507      508      509      510      511      512
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      513      514      515      516      517      518      519      520
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      521      522      523      524      525      526      527      528
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      529      530      531      532      533      534      535      536
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      537      538      539      540      541      542      543      544
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      545      546      547      548      549      550      551      552
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      553      554      555      556      557      558      559      560
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      561      562      563      564      565      566      567      568
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      569      570      571      572      573      574      575      576
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      577      578      579      580      581      582      583      584
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      585      586      587      588      589      590      591      592
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      593      594      595      596      597      598      599      600
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      601      602      603      604      605      606      607      608
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      609      610      611      612      613      614      615      616
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      617      618      619      620      621      622      623      624
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      625      626      627      628      629      630      631      632
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      633      634      635      636      637      638      639      640
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      641      642      643      644      645      646      647      648
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      649      650      651      652      653      654      655      656
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      657      658      659      660      661      662      663      664
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      665      666      667      668      669      670      671      672
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      673      674      675      676      677      678      679      680
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      681      682      683      684      685      686      687      688
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      689      690      691      692      693      694      695      696
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      697      698      699      700      701      702      703      704
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      705      706      707      708      709      710      711      712
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      713      714      715      716      717      718      719      720
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      721      722      723      724      725      726      727      728
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      729      730      731      732      733      734      735      736
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      737      738      739      740      741      742      743      744
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      745      746      747      748      749      750      751      752
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      753      754      755      756      757      758      759      760
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      761      762      763      764      765      766      767      768
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      769      770      771      772      773      774      775      776
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      777      778      779      780      781      782      783      784
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      785      786      787      788      789      790      791      792
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      793      794      795      796      797      798      799      800
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      801      802      803      804      805      806      807      808
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      809      810      811      812      813      814      815      816
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      817      818      819      820      821      822      823      824
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      825      826      827      828      829      830      831      832
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      833      834      835      836      837      838      839      840
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      841      842      843      844      845      846      847      848
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      849      850      851      852      853      854      855      856
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      857      858      859      860      861      862      863      864
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      865      866      867      868      869      870      871      872
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      873      874      875      876      877      878      879      880
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      881      882      883      884      885      886      887      888
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      889      890      891      892      893      894      895      896
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      897      898      899      900      901      902      903      904
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      905      906      907      908      909      910      911      912
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      913      914      915      916      917      918      919      920
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      921      922      923      924      925      926      927      928
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      929      930      931      932      933      934      935      936
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      937      938      939      940      941      942      943      944
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      945      946      947      948      949      950      951      952
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      953      954      955      956      957      958      959      960
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      961      962      963      964      965      966      967      968
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      969      970      971      972      973      974      975      976
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##      977      978      979      980      981      982      983      984
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##      985      986      987      988      989      990      991      992
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##      993      994      995      996      997      998      999     1000
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1001     1002     1003     1004     1005     1006     1007     1008
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1009     1010     1011     1012     1013     1014     1015     1016
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1017     1018     1019     1020     1021     1022     1023     1024
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1025     1026     1027     1028     1029     1030     1031     1032
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1033     1034     1035     1036     1037     1038     1039     1040
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1041     1042     1043     1044     1045     1046     1047     1048
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1049     1050     1051     1052     1053     1054     1055     1056
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1057     1058     1059     1060     1061     1062     1063     1064
```

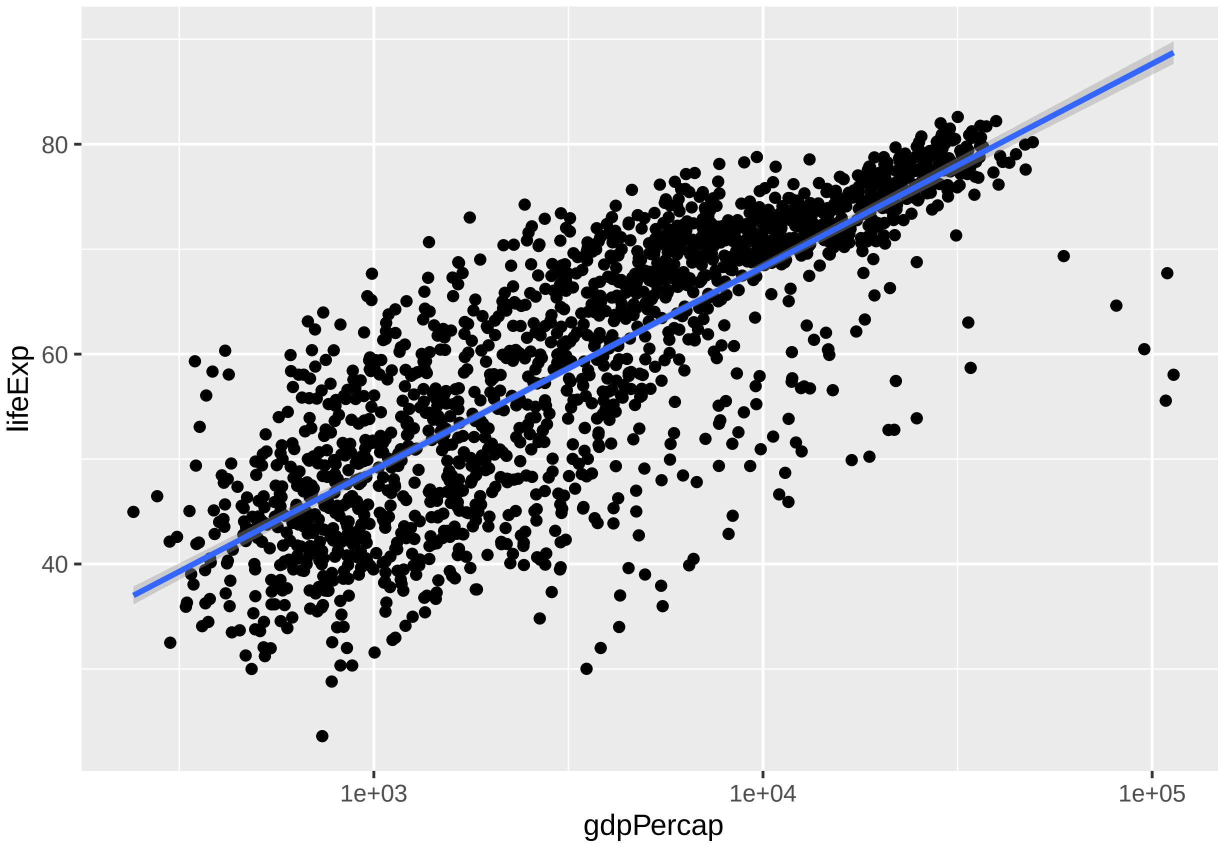
```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1065     1066     1067     1068     1069     1070     1071     1072
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1073     1074     1075     1076     1077     1078     1079     1080
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1081     1082     1083     1084     1085     1086     1087     1088
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1089     1090     1091     1092     1093     1094     1095     1096
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1097     1098     1099     1100     1101     1102     1103     1104
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1105     1106     1107     1108     1109     1110     1111     1112
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1113     1114     1115     1116     1117     1118     1119     1120
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1121     1122     1123     1124     1125     1126     1127     1128
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1129     1130     1131     1132     1133     1134     1135     1136
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1137     1138     1139     1140     1141     1142     1143     1144
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1145     1146     1147     1148     1149     1150     1151     1152
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1153     1154     1155     1156     1157     1158     1159     1160
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1161     1162     1163     1164     1165     1166     1167     1168
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1169     1170     1171     1172     1173     1174     1175     1176
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1177     1178     1179     1180     1181     1182     1183     1184
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1185     1186     1187     1188     1189     1190     1191     1192
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1193     1194     1195     1196     1197     1198     1199     1200
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1201     1202     1203     1204     1205     1206     1207     1208
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1209     1210     1211     1212     1213     1214     1215     1216
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1217     1218     1219     1220     1221     1222     1223     1224
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1225     1226     1227     1228     1229     1230     1231     1232
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1233     1234     1235     1236     1237     1238     1239     1240
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1241     1242     1243     1244     1245     1246     1247     1248
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1249     1250     1251     1252     1253     1254     1255     1256
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1257     1258     1259     1260     1261     1262     1263     1264
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1265     1266     1267     1268     1269     1270     1271     1272
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1273     1274     1275     1276     1277     1278     1279     1280
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1281     1282     1283     1284     1285     1286     1287     1288
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1289     1290     1291     1292     1293     1294     1295     1296
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1297     1298     1299     1300     1301     1302     1303     1304
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1305     1306     1307     1308     1309     1310     1311     1312
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1313     1314     1315     1316     1317     1318     1319     1320
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1321     1322     1323     1324     1325     1326     1327     1328
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1329     1330     1331     1332     1333     1334     1335     1336
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1337     1338     1339     1340     1341     1342     1343     1344
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1345     1346     1347     1348     1349     1350     1351     1352
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1353     1354     1355     1356     1357     1358     1359     1360
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1361     1362     1363     1364     1365     1366     1367     1368
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1369     1370     1371     1372     1373     1374     1375     1376
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1377     1378     1379     1380     1381     1382     1383     1384
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1385     1386     1387     1388     1389     1390     1391     1392
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1393     1394     1395     1396     1397     1398     1399     1400
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1401     1402     1403     1404     1405     1406     1407     1408
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1409     1410     1411     1412     1413     1414     1415     1416
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1417     1418     1419     1420     1421     1422     1423     1424
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1425     1426     1427     1428     1429     1430     1431     1432
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1433     1434     1435     1436     1437     1438     1439     1440
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1441     1442     1443     1444     1445     1446     1447     1448
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1449     1450     1451     1452     1453     1454     1455     1456
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1457     1458     1459     1460     1461     1462     1463     1464
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1465     1466     1467     1468     1469     1470     1471     1472
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1473     1474     1475     1476     1477     1478     1479     1480
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1481     1482     1483     1484     1485     1486     1487     1488
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1489     1490     1491     1492     1493     1494     1495     1496
```

```
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1497     1498     1499     1500     1501     1502     1503     1504
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1505     1506     1507     1508     1509     1510     1511     1512
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1513     1514     1515     1516     1517     1518     1519     1520
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1521     1522     1523     1524     1525     1526     1527     1528
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1529     1530     1531     1532     1533     1534     1535     1536
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1537     1538     1539     1540     1541     1542     1543     1544
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1545     1546     1547     1548     1549     1550     1551     1552
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1553     1554     1555     1556     1557     1558     1559     1560
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1561     1562     1563     1564     1565     1566     1567     1568
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1569     1570     1571     1572     1573     1574     1575     1576
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1577     1578     1579     1580     1581     1582     1583     1584
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1585     1586     1587     1588     1589     1590     1591     1592
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1593     1594     1595     1596     1597     1598     1599     1600
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1601     1602     1603     1604     1605     1606     1607     1608
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1609     1610     1611     1612     1613     1614     1615     1616
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1617     1618     1619     1620     1621     1622     1623     1624
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1625     1626     1627     1628     1629     1630     1631     1632
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1633     1634     1635     1636     1637     1638     1639     1640
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1641     1642     1643     1644     1645     1646     1647     1648
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1649     1650     1651     1652     1653     1654     1655     1656
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1657     1658     1659     1660     1661     1662     1663     1664
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1665     1666     1667     1668     1669     1670     1671     1672
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1673     1674     1675     1676     1677     1678     1679     1680
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
##     1681     1682     1683     1684     1685     1686     1687     1688
## 67.79013 68.98264 70.17514 71.36765 72.56016 73.75266 74.94517 76.13768
##     1689     1690     1691     1692     1693     1694     1695     1696
## 77.33018 78.52269 79.71520 80.90771 67.79013 68.98264 70.17514 71.36765
##     1697     1698     1699     1700     1701     1702     1703     1704
## 72.56016 73.75266 74.94517 76.13768 77.33018 78.52269 79.71520 80.90771
```

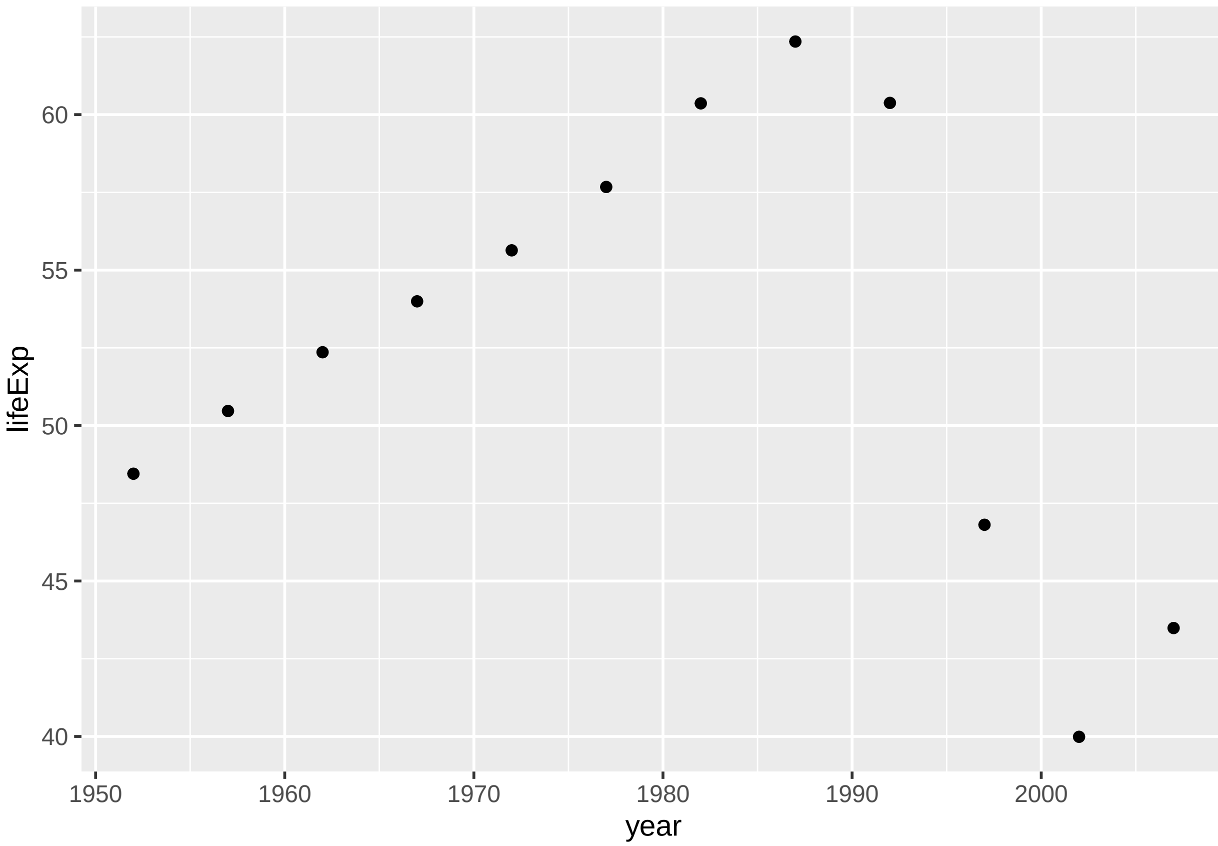
We can plot models (with one predictor/ X variable) using `ggplot2` through the `geom_smooth()` layer. Specifying `method="lm"` gives us the linear regression fit (but only visually!):

```
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
    geom_point() +
    geom_smooth(method="lm") +
    scale_x_log10()
```



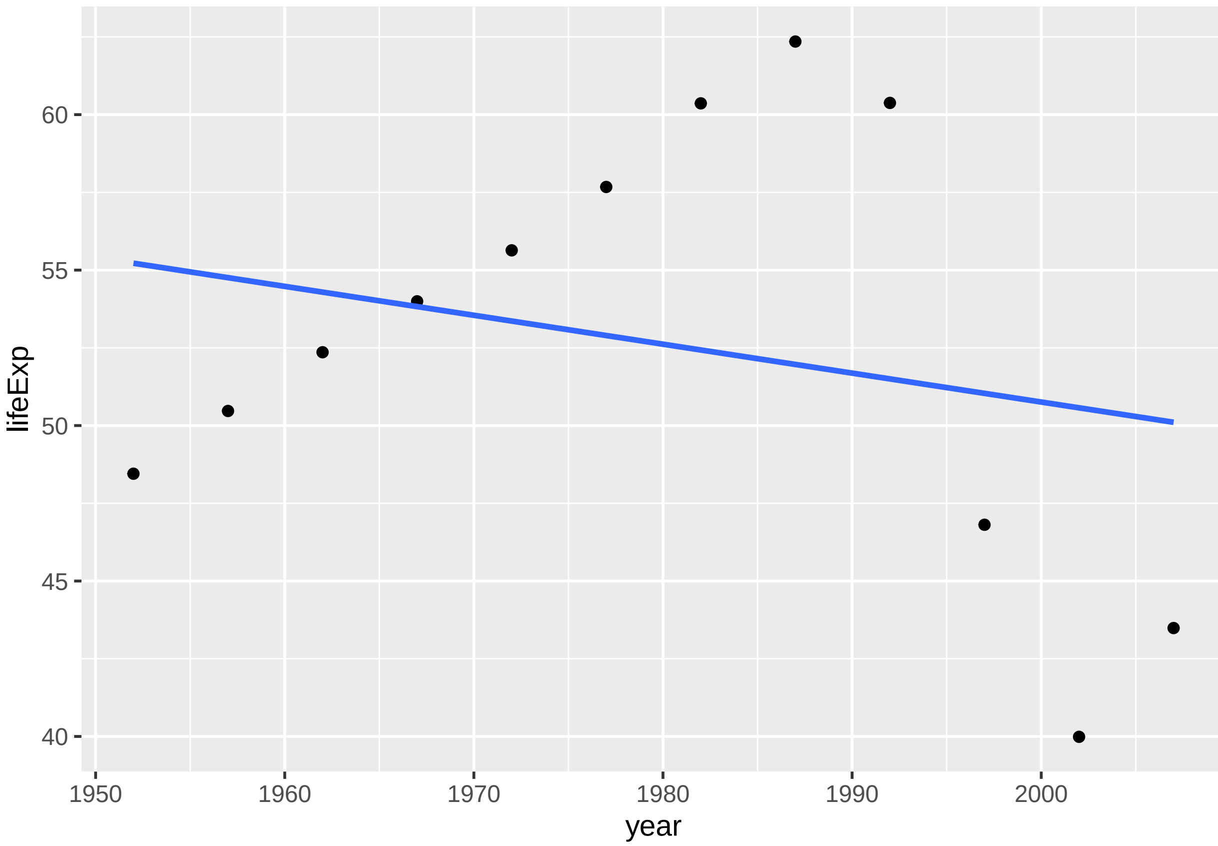
Lets consider another country "Zimbabwe", which has a unique behavior in the `lifeExp` and `year` relationship.

```
gapminder_Zimbabwe <- gapminder%>%
  filter(country=="Zimbabwe")
gapminder_Zimbabwe %>% ggplot(aes(year, lifeExp)) + geom_point()
```



Let's try fitting a linear model to this relationship

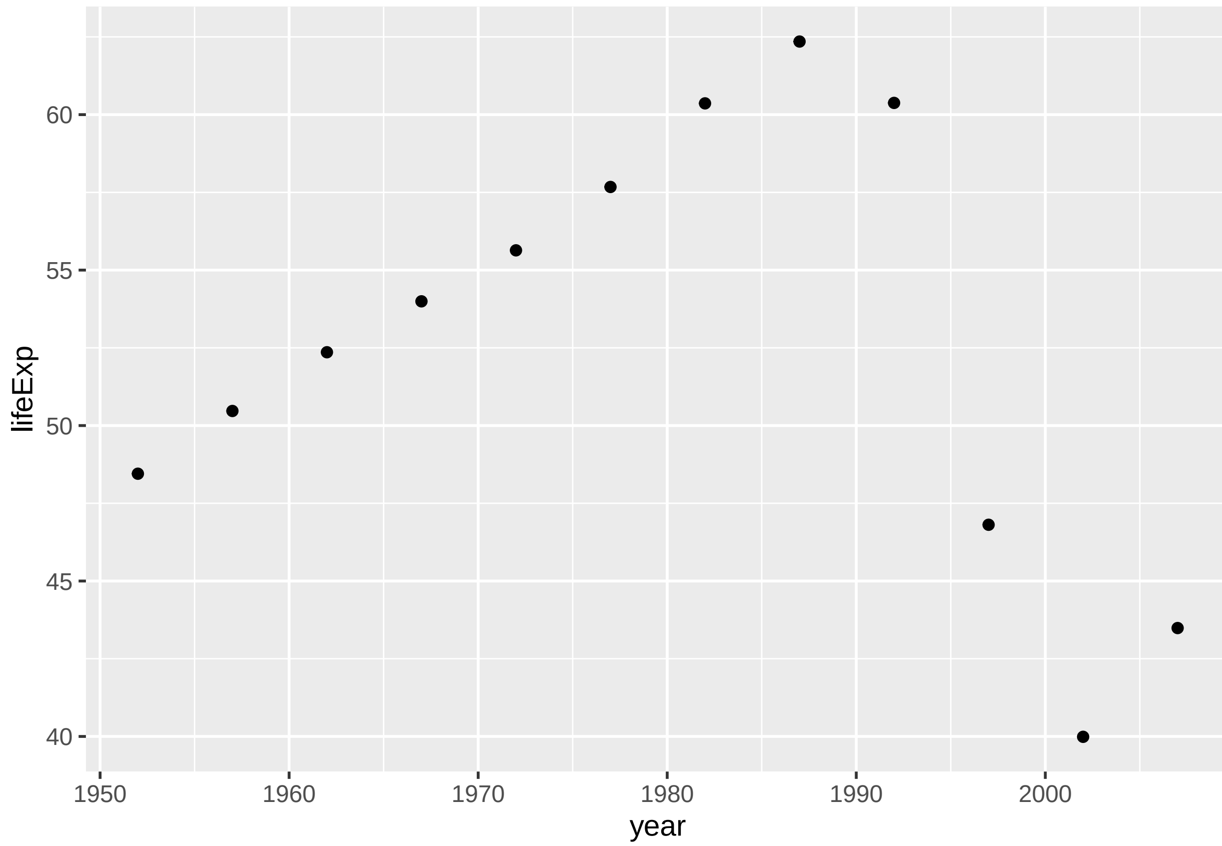
```
ggplot(gapminder_Zimbabwe, aes(year,lifeExp)) + geom_point()+geom_smooth(method = "lm", se = F)
```



Now we will try to fit a second degree polynomial and see what would that look like.

```
ggplot(gapminder_Zimbabwe, aes(year,lifeExp))+
  geom_point()+
  geom_smooth(method = "lm", se = F, formula=lifeExp~poly(year, 2))
```

```
## Warning: Computation failed in `stat_smooth()`:
## object 'lifeExp' not found
```




```r
lm_linear <- lm(data = gapminder,formula = lifeExp~year)
lm_poly <- lm(data = gapminder,formula = lifeExp~poly(year, 2))
```

`anova` lets you compare between different models.

```r
anova(lm_linear,lm_poly)
```

```
## Analysis of Variance Table
##
## Model 1: lifeExp ~ year
## Model 2: lifeExp ~ poly(year, 2)
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1   1702 230229
## 2   1701 228793  1    1436.2 10.678 0.001106 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression with categorical variables

```r
(lm_cat <- lm(gdpPercap ~ I(year - 1952) + continent, data = gapminder))
```

```
##
## Call:
```

```
## lm(formula = gdpPercap ~ I(year - 1952) + continent, data = gapminder)
##
## Coefficients:
##       (Intercept)     I(year - 1952)  continentAmericas
##           -1375.3              129.8             4942.4
##     continentAsia    continentEurope   continentOceania
##            5708.4            12275.7            16427.9
```

How did R know that continent was a categorical variable?

```r
class(gapminder$continent)
```

```
## [1] "factor"
```

```r
levels(gapminder$continent)
```

```
## [1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

```r
contrasts(gapminder$continent)
```

```
##          Americas Asia Europe Oceania
## Africa          0    0      0       0
## Americas        1    0      0       0
## Asia            0    1      0       0
## Europe          0    0      1       0
## Oceania         0    0      0       1
```

How can we change the reference level?

```r
gapminder$continent <- relevel(gapminder$continent, ref = "Oceania")
```

Let's build a new model

```r
lm_cat2 <- lm(gdpPercap ~ I(year - 1952) + continent, data = gapminder)
lm_cat2
```

```
##
## Call:
## lm(formula = gdpPercap ~ I(year - 1952) + continent, data = gapminder)
##
## Coefficients:
##       (Intercept)     I(year - 1952)    continentAfrica
##           15052.5              129.8           -16427.9
## continentAmericas      continentAsia    continentEurope
##          -11485.5           -10719.5            -4152.1
```

Broom

Let's make it easier to extract info, using the `broom` package. There are three crown functions in this package, all of which input a fitted model, and outputs a tidy data frame.

1. `tidy`: extract statistical summaries about each component of the model.
   - Useful for *interpretation* task.
2. `augment`: add columns to the original data frame, giving information corresponding to each row.
   - Useful for *prediction* task.
3. `glance`: extract statistical summaries about the model as a whole (1-row tibble).
   - Useful for checking goodness of fit.

Exercise: apply all three functions to our fitted model, `my_lm`. What do you see?

```
tidy(my_lm) #This gives us a 'tidy' table with the different values from our regression model.
```

```
## # A tibble: 2 x 5
##   term                estimate std.error statistic  p.value
##   <chr>                  <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)           67.8     0.119       567.  7.12e-24
## 2 I(year - min(year))    0.239   0.00368      64.8 1.86e-14
```

```
augment(my_lm) #Augment gives us additional variables such as residuals. It also gives us observation-l
```

```
## # A tibble: 12 x 9
##    lifeExp I.year...min.ye~ .fitted .se.fit  .resid   .hat .sigma .cooksd
##      <dbl>         <I<int>>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>
##  1    67.4                0    67.8  0.119  -0.380  0.295   0.176 0.885
##  2    68.9                5    69.0  0.104  -0.0526 0.225   0.231 0.0107
##  3    70.5               10    70.2  0.0905  0.335  0.169   0.197 0.283
##  4    71.6               15    71.4  0.0784  0.182  0.127   0.223 0.0572
##  5    72.4               20    72.6  0.0693 -0.180  0.0991  0.223 0.0409
##  6    73.8               25    73.8  0.0642  0.0773 0.0851  0.230 0.00628
##  7    74.9               30    74.9  0.0642 -0.0552 0.0851  0.231 0.00319
##  8    76.3               35    76.1  0.0693  0.202  0.0991  0.221 0.0516
##  9    77.5               40    77.3  0.0784  0.130  0.127   0.227 0.0290
## 10    78.6               45    78.5  0.0905  0.117  0.169   0.228 0.0348
## 11    79.6               50    79.7  0.104  -0.125  0.225   0.227 0.0606
## 12    80.7               55    80.9  0.119  -0.251  0.295   0.210 0.385
## # ... with 1 more variable: .std.resid <dbl>
```

```
glance(my_lm) #glance gives us a concise one-row summary of the model including AIC and BIC.
```

```
## # A tibble: 1 x 11
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
## 1     0.998         0.997 0.220     4200. 1.86e-14     2   2.23  1.53  2.99
## # ... with 2 more variables: deviance <dbl>, df.residual <int>
```