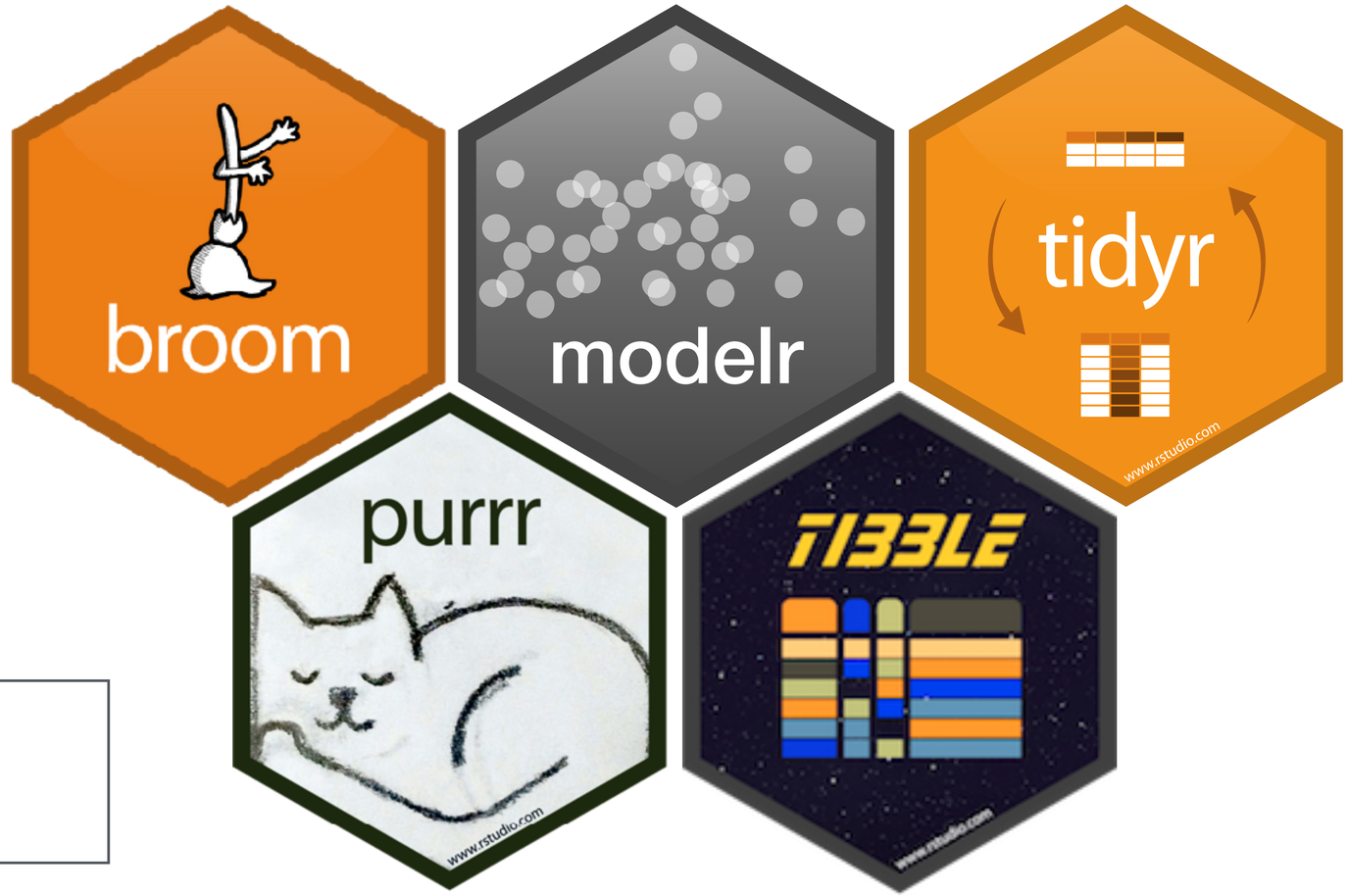
Organize with list columns



In R4DS

Many Models

Your Turn

Open **08-Organize.Rmd**

gapminder

A subset of the data available at Hans Rosling's gapminder.org

```
# install.packages("gapminder")
library(gapminder)
```

```
gapminder
```

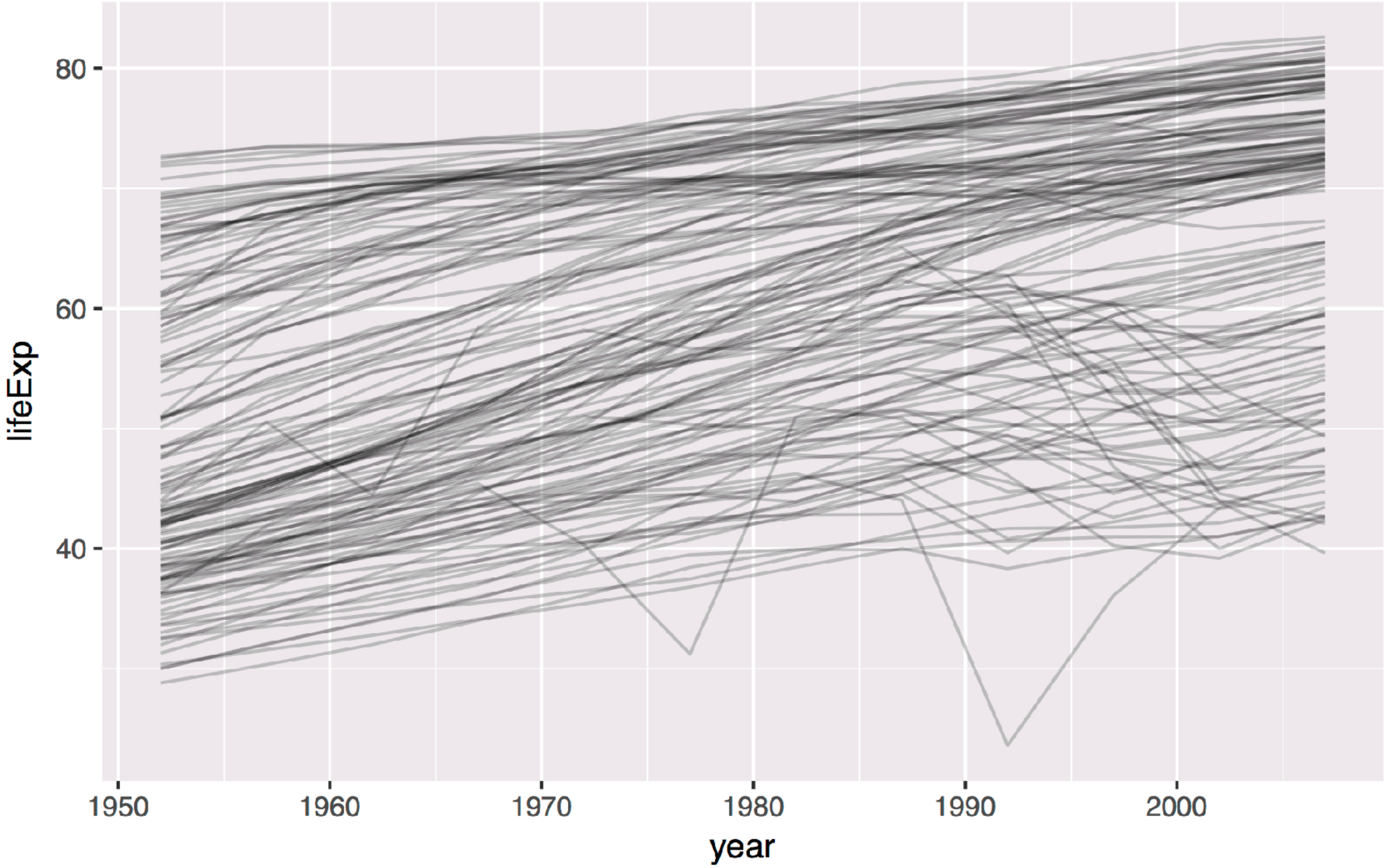
country <fctr>	continent <fctr>	year <int>	lifeExp <dbl>	pop <int>	gdpPercap <dbl>
Afghanistan	Asia	1952	28.80100	8425333	779.4453
Afghanistan	Asia	1957	30.33200	9240934	820.8530
Afghanistan	Asia	1962	31.99700	10267083	853.1007
Afghanistan	Asia	1967	34.02000	11537966	836.1971
Afghanistan	Asia	1972	36.08800	13079460	739.9811
Afghanistan	Asia	1977	38.43800	14880372	786.1134
Afghanistan	Asia	1982	39.85400	12881816	978.0114
Afghanistan	Asia	1987	40.82200	13867957	852.3959
Afghanistan	Asia	1992	41.67400	16317921	649.3414
Afghanistan	Asia	1997	41.76300	22227415	635.3414

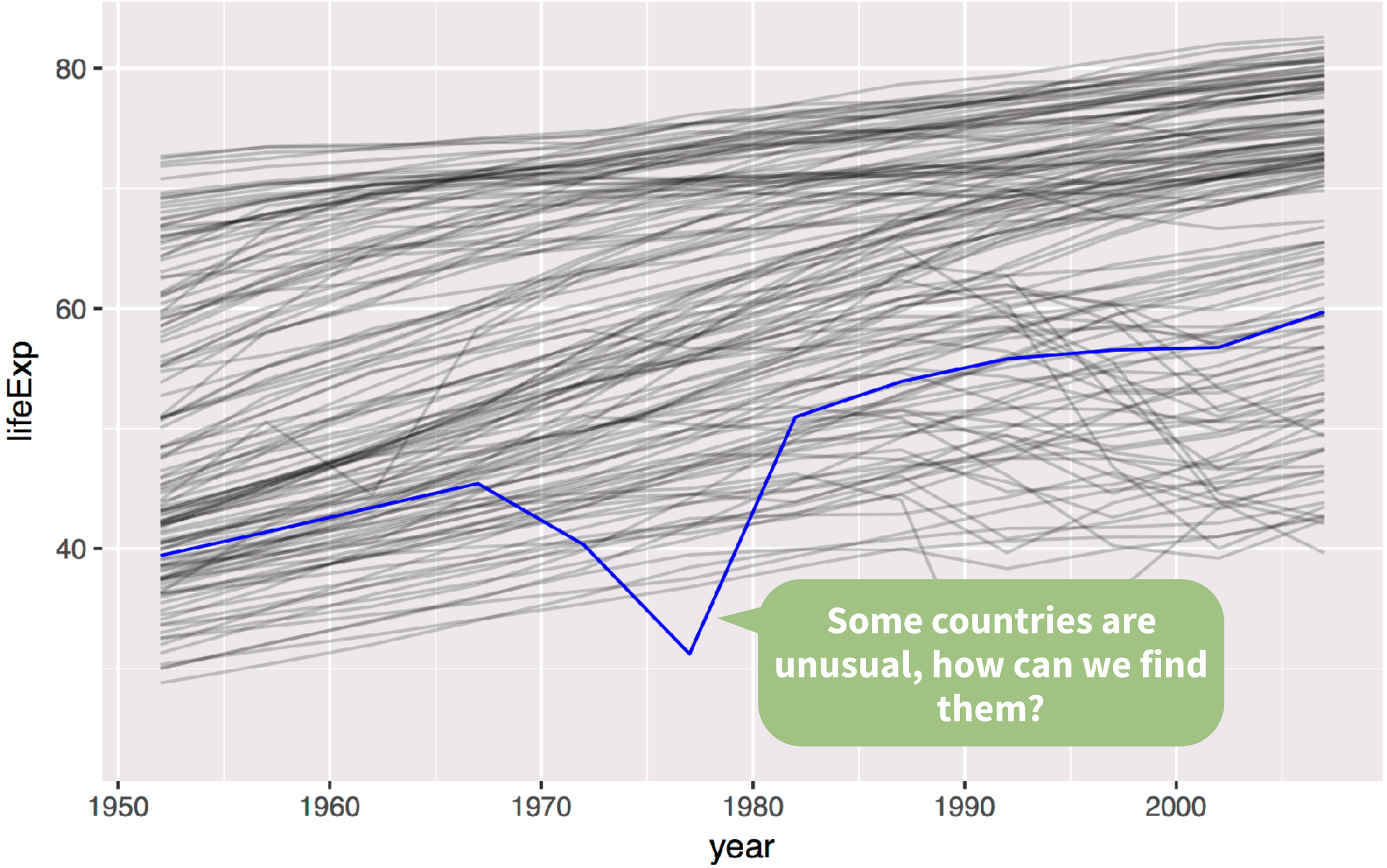
Your Turn 1

How has life expectancy changed over time?

Make a line plot of **lifeExp** vs. **year** grouped by **country**. Set alpha to 0.2, to see the results better.

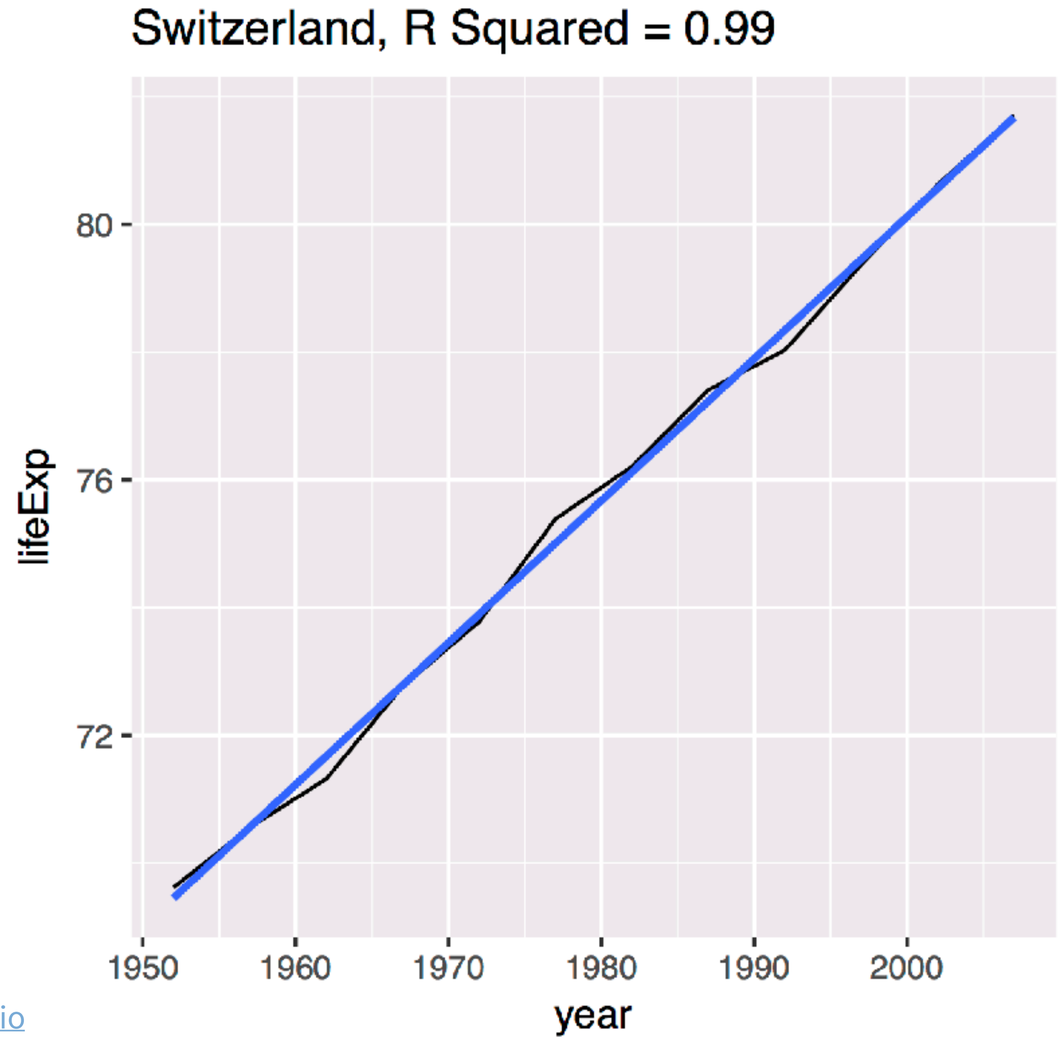
```
gapminder %>%
  ggplot(mapping = aes(x = year, y = lifeExp, group = country)) +
    geom_line(alpha = 0.2)
```

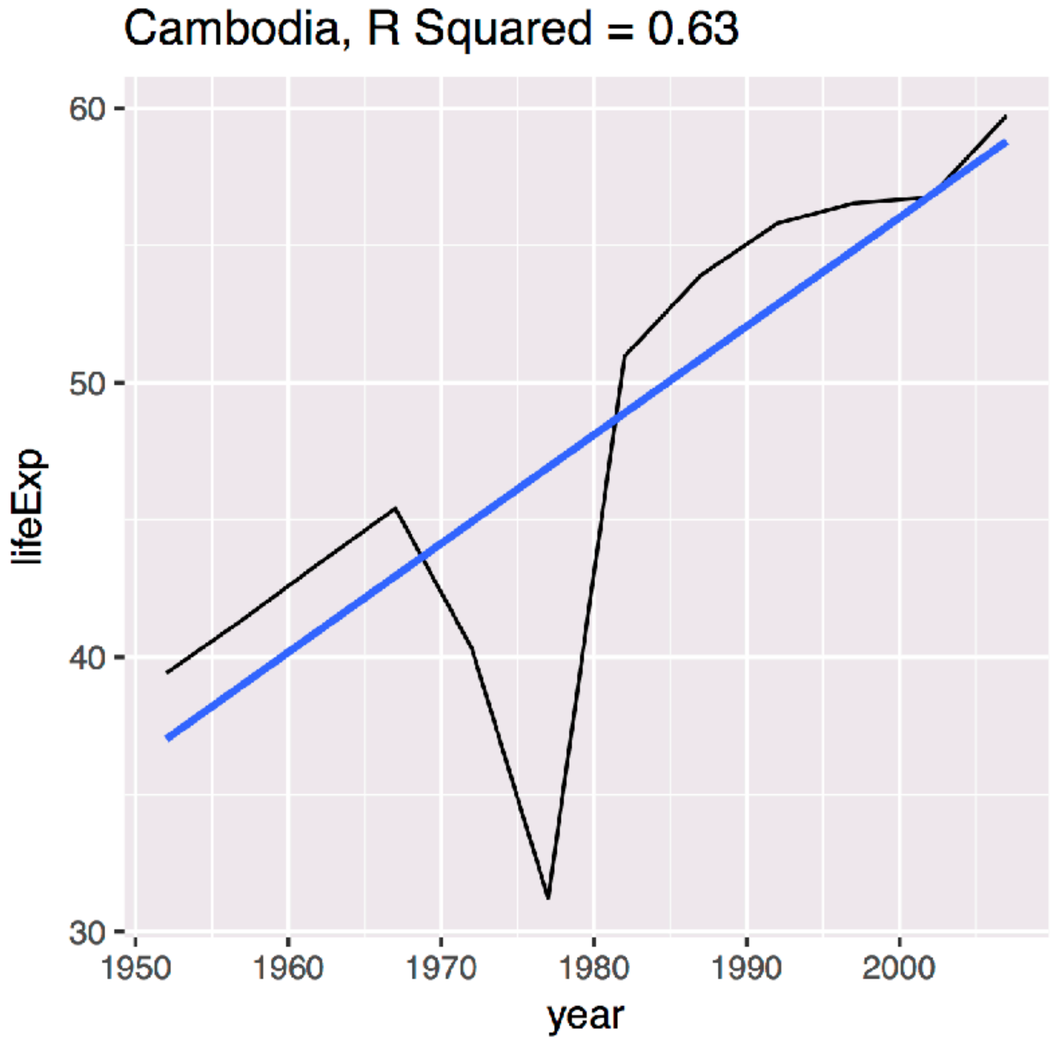

80
60
40
lifeExp
1950
1960
1970
1980
1990
2000
year


80
60
40
lifeExp
1950
1960
1970
1980
1990
2000
year
Some countries are unusual, how can we find them?

Idea 1

To quantify "linearity," fit a linear model, compare **r-squared**.

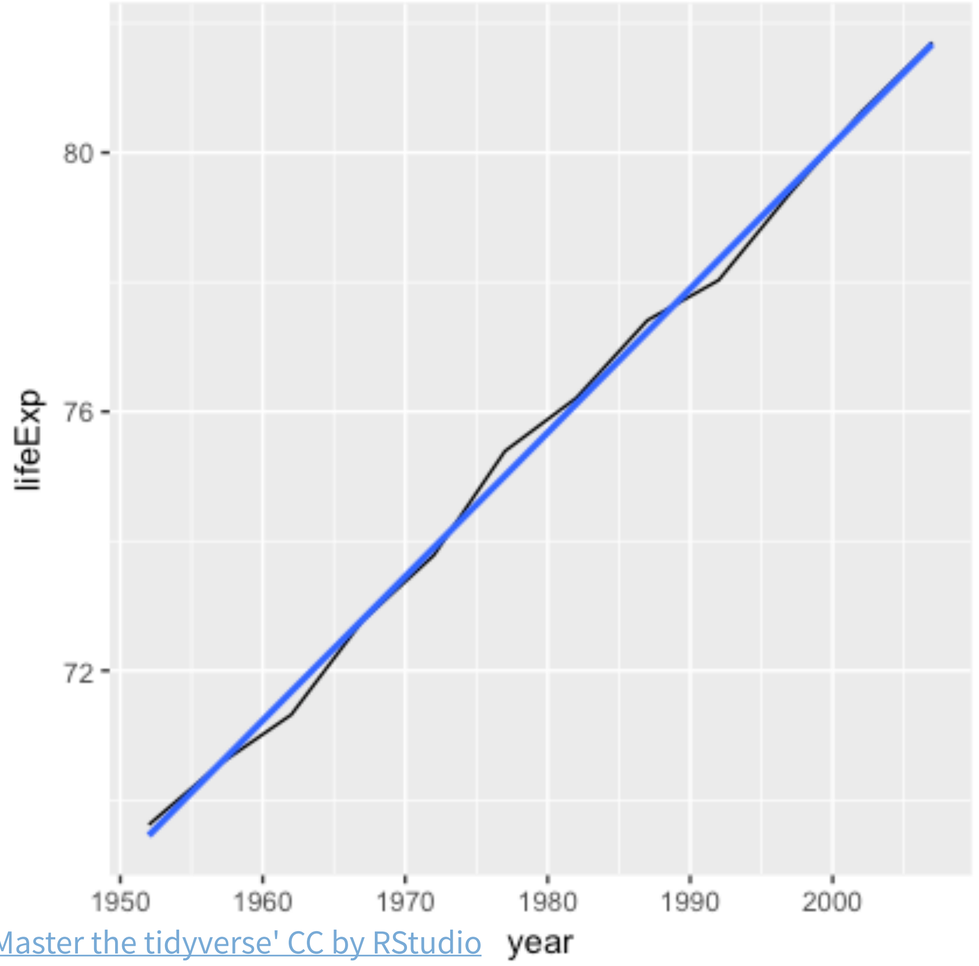




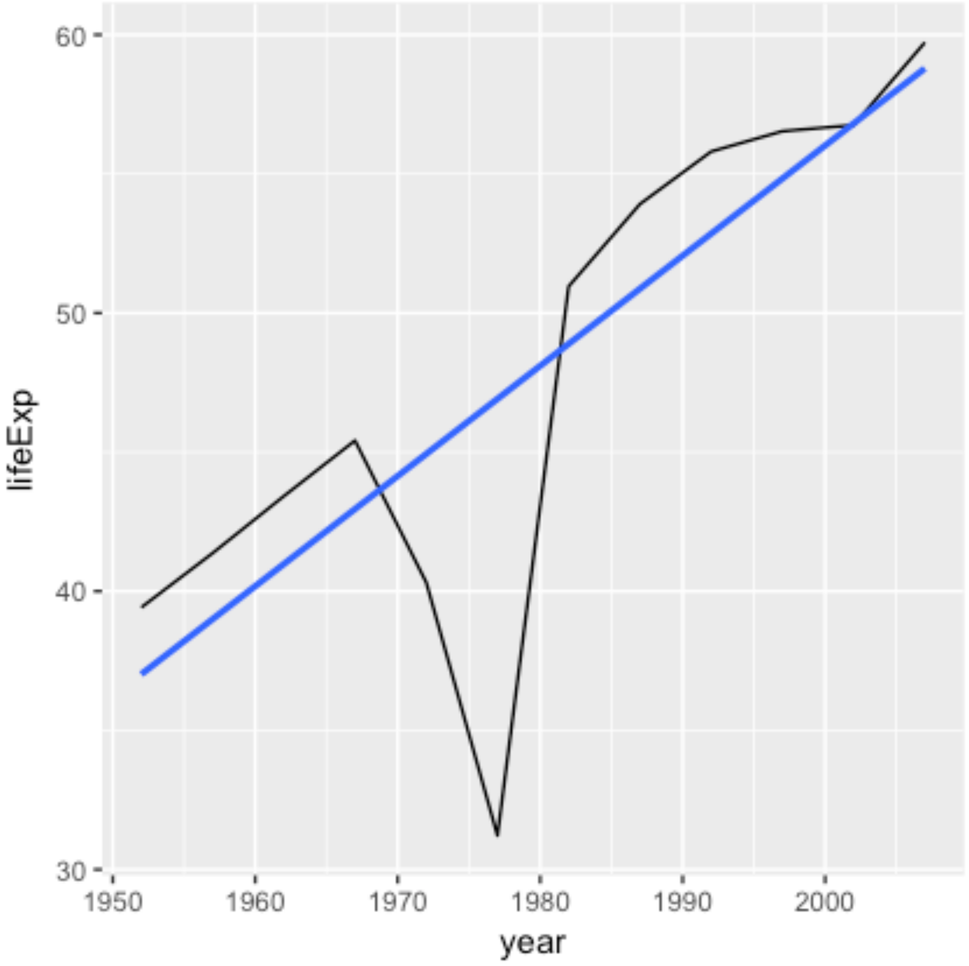
Idea 2

To quantify rate of change fit a linear model, extract **coefficient on year**.

Switzerland = 0.22 years/year



Cambodia = 0.40 years/year



Goal

Fit model, compute r.squared, collect coefficient ***for every country.***

1. **dplyr + tidyr** grouping toolkit
2. **purrr** toolkit and list columns

List columns

Quiz

How is a data frame/tibble similar to a list?

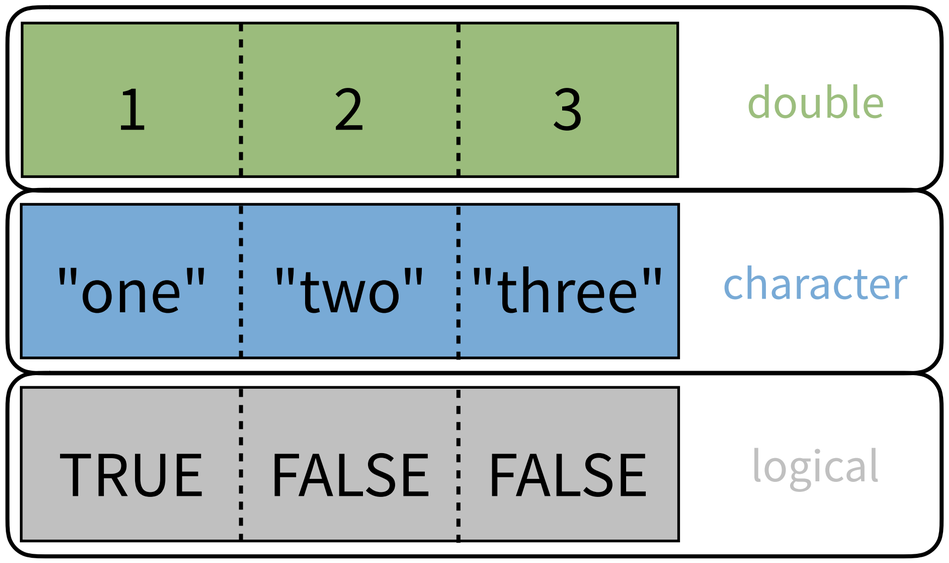
A data frame / tibble is a list

data frame

num	cha	log
1	"one"	TRUE
2	"two"	FALSE
3	"three"	FALSE

=

List



\+ class = "data.frame"

A data frame / tibble is a list

data frame

num	cha	log
1	"one"	TRUE
2	"two"	FALSE
3	"three"	FALSE

`df["num"]`

num
1
2
3

`df[["num"]]`

`df$num`

`c(1, 2, 3)`

A data frame / tibble is a list

data frame

num	cha	log
1	"one"	TRUE
2	"two"	FALSE
3	"three"	FALSE

`df %>% select(num)`

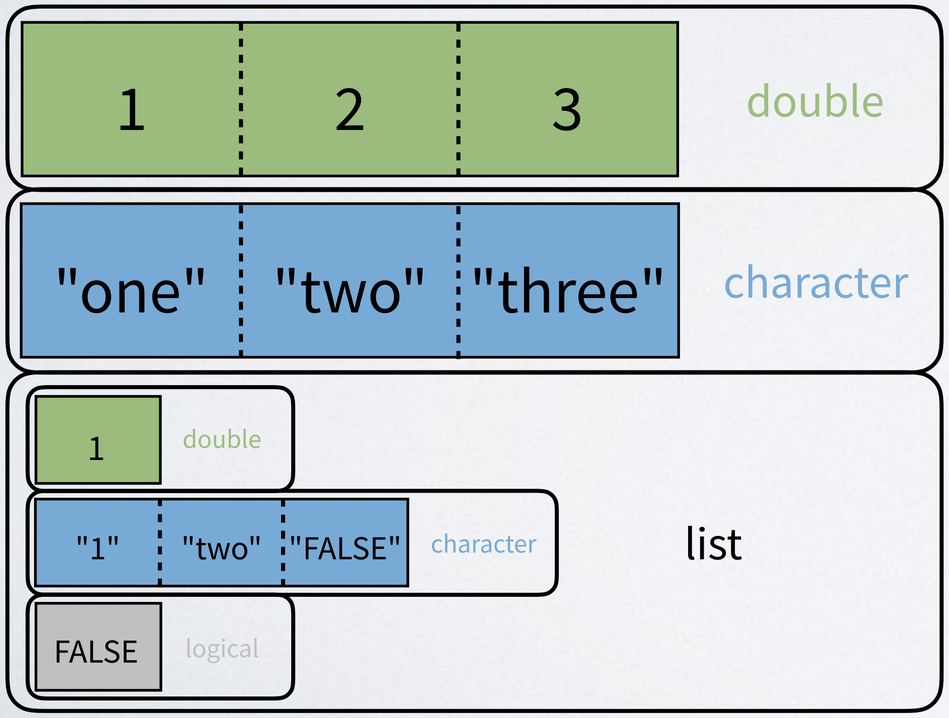
num
1
2
3

Quiz

If one of the elements of a list can be another list,

can one of the columns of a data frame be another list?

List



?

=

data frame

num	cha	listcol
1	"one"	1
2	"two"	c("1", "two", "FALSE")
3	"three"	FALSE

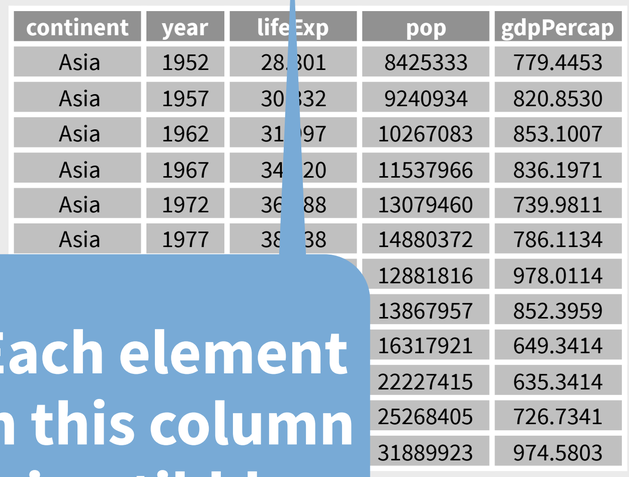
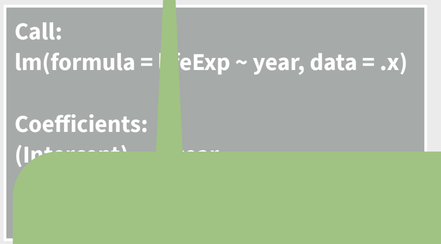
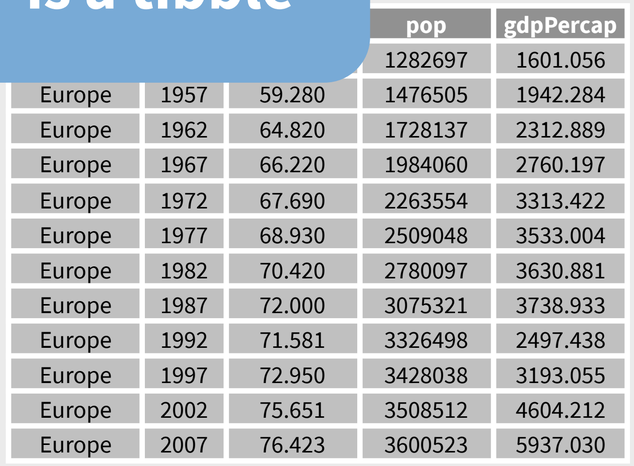
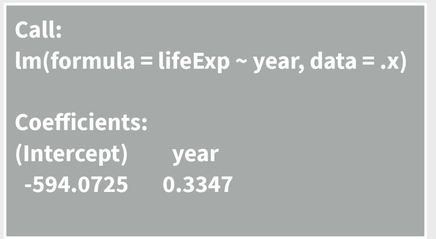
Yes.

```
tibble(
  num = c(1, 2, 3),
  cha = c("one", "two", "three"),
  listcol = list(1, c("1", "two", "FALSE"), FALSE)
)
```

num <dbl>	cha <chr>	listcol <list>
1	one	<dbl [1]>
2	two	<chr [3]>
3	three	<lgl [1]>

3 rows

Goal

country	data	model
Afghanistan		
Albania		
		

Why?

country	data	model	r.squared
Afghanistan	continent year lifeExp pop gdpPercap Asia 1952 28.801 8425333 779.4453 Asia 1957 30.332 9240934 820.8530 Asia 1962 31.997 10267083 853.1007 Asia 1967 34.020 11537966 836.1971 Asia 1972 36.088 13079460 739.9811 Asia 1977 38.438 14880372 786.1134 Asia 1982 39.854 12881816 978.0114 Asia 1987 40.822 13867957 852.3959 Asia 1992 41.674 16317921 649.3414 Asia 1997 41.763 22227415 635.3414 Asia 2002 42.129 25268405 726.7341 Asia 2007 43.828 31889923 974.5803	Call: lm(formula = lifeExp ~ year, data = .x) Coefficients: (Intercept) year -507.5343 0.2753	0.034
Albania	Europe 1982 70.420 2780097 3630.881 Europe 1987 72.000 3075321 3738.933 Europe 1992 71.581 3326498 2497.438 Europe 1997 72.950 3428038 3193.055 Europe 2002 75.651 3508512 4604.212 Europe 2007 76.423 3600523 5937.030	(Intercept) year -594.0725 0.3347	0.493
Algeria	continent year lifeExp pop gdpPercap Africa 1952 43.077 9279525 2449.008	Call:	

Organization.

We keep the things that are related together.

Nesting

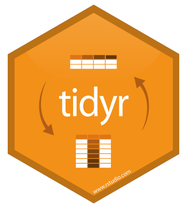
nest()

Nest rows into a list column by group.

```
nest(data,.key = "data")
```

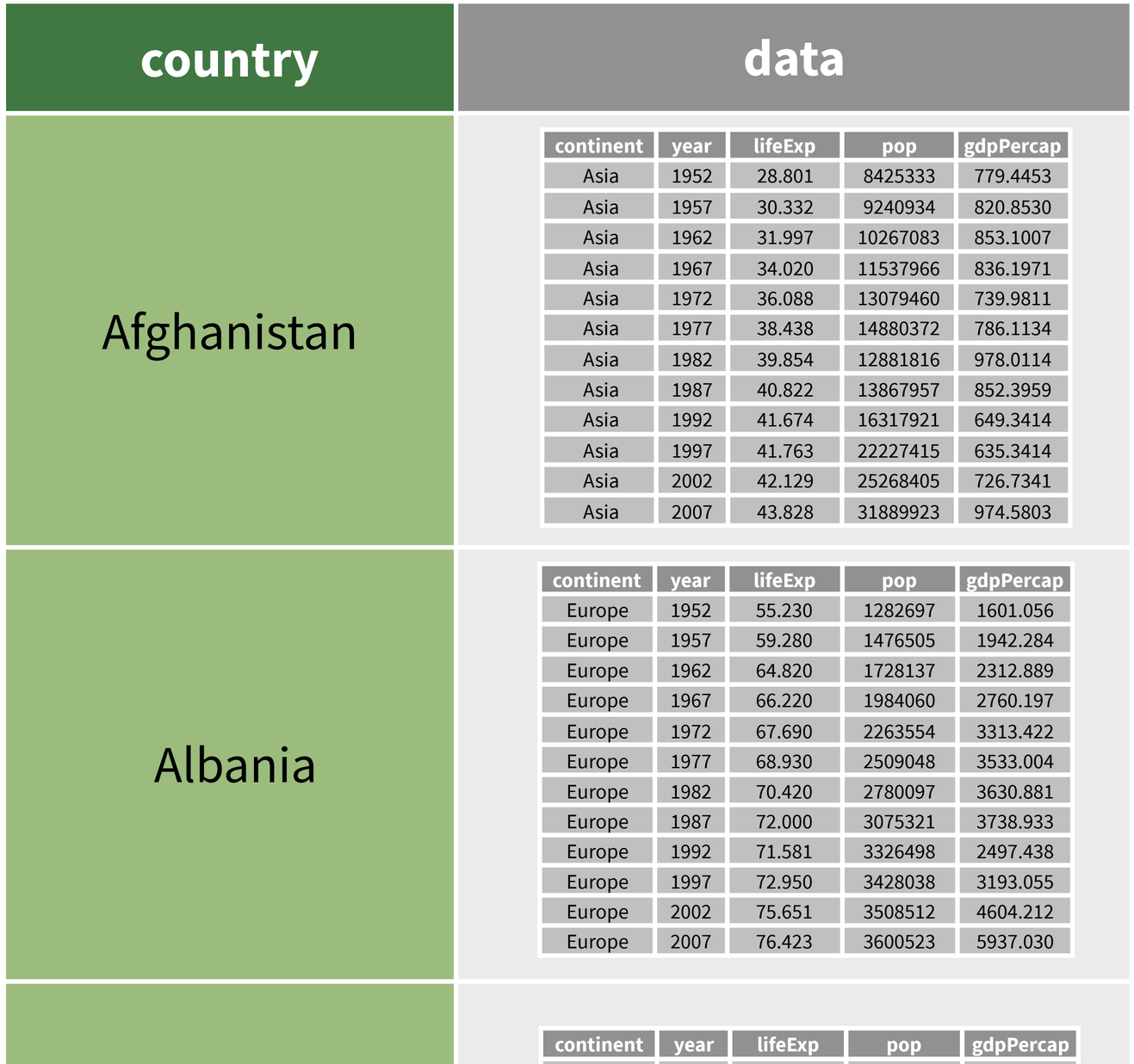
A grouped data frame

name for the new list column



Places grouped cases into a list column.

```
gapminder %>%
  group_by(country) %>%
  nest()
```



country	data
Afghanistan	

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

country	data
Albania	

continent	year	lifeExp	pop	gdpPercap
Europe	1952	55.230	1282697	1601.056
Europe	1957	59.280	1476505	1942.284
Europe	1962	64.820	1728137	2312.889
Europe	1967	66.220	1984060	2760.197
Europe	1972	67.690	2263554	3313.422
Europe	1977	68.930	2509048	3533.004
Europe	1982	70.420	2780097	3630.881
Europe	1987	72.000	3075321	3738.933
Europe	1992	71.581	3326498	2497.438
Europe	1997	72.950	3428038	3193.055
Europe	2002	75.651	3508512	4604.212
Europe	2007	76.423	3600523	5937.030

continent	year	lifeExp	pop	gdpPercap

```
gapminder
```

country <fctr>	continent <fctr>	year <int>	lifeExp <dbl>	pop <int>	gdpPercap <dbl>
Afghanistan	Asia	1952	28.80100	8425333	779.4453
Afghanistan	Asia	1957	30.33200	9240934	820.8530
Afghanistan	Asia	1962	31.99700	10267083	853.1007
Afghanistan	Asia	1967	34.02000	11537966	836.1971
Afghanistan	Asia	1972	36.08800	13079460	739.9811
Afghanistan	Asia	1977	38.43800	14880372	786.1134
Afghanistan	Asia	1982	39.85400	12881816	978.0114
Afghanistan	Asia	1987	40.82200	13867957	852.3959
Afghanistan	Asia	1992	41.67400	16317921	649.3414
Afghanistan	Asia	1997	41.76300	22227415	635.3414

```
gapminder_nested <- gapminder %>%
  group_by(country) %>%
  nest()
```

country <fctr>	data <list>
Afghanistan	<tibble>
Albania	<tibble>
Algeria	<tibble>
Angola	<tibble>
Argentina	<tibble>
Australia	<tibble>
Austria	<tibble>
Bahrain	<tibble>
Bangladesh	<tibble>
Belgium	<tibble>

1–10 of 142 rows

Adapted from

```
gapminder_nested$data[[1]]
```

country <fctr>	data <list>
Afghanistan	<tibble>
Albania	<tibble>
Algeria	<tibble>
Angola	<tibble>
Argentina	<tibble>
Australia	<tibble>
Austria	<tibble>
Bahrain	<tibble>
Bangladesh	<tibble>
Belgium	<tibble>

1–10 of 142 rows

Previous 1 2 3 4 5 6 … 15 Next

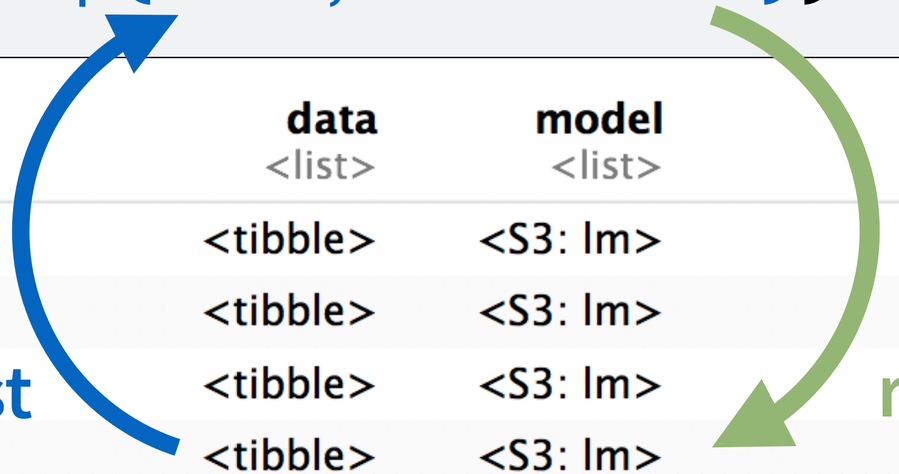
continent <fctr>	year <int>	lifeExp <dbl>	pop <int>	gdpPercap <dbl>
Asia	1952	28.801	8425333	779.4453
Asia	1957	30.332	9240934	820.8530
Asia	1962	31.997	10267083	853.1007
Asia	1967	34.020	11537966	836.1971
Asia	1972	36.088	13079460	739.9811
Asia	1977	38.438	14880372	786.1134
Asia	1982	39.854	12881816	978.0114
Asia	1987	40.822	13867957	852.3959
Asia	1992	41.674	16317921	649.3414
Asia	1997	41.763	22227415	635.3414

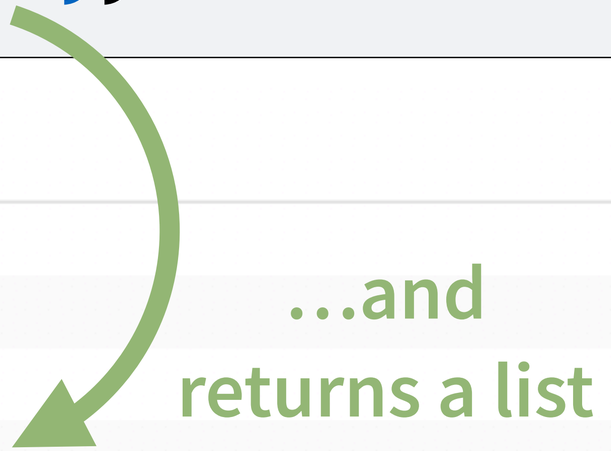
1–10 of 12 rows

Previous 1 2 Next

```
fit_model <- function(df) lm(lifeExp ~ year, data = df)

gapminder_nested <- gapminder_nested %>%
  mutate(model = map(data, fit_model))
```

country <fctr>	data <list>	model <list>
Afghanistan	<tibble>	<S3: lm>
Albania	<tibble>	<S3: lm>
Algeria	<tibble>	<S3: lm>
Angola	<tibble>	<S3: lm>
Argentina	<tibble>	<S3: lm>
Australia	<tibble>	<S3: lm>
Austria	<tibble>	<S3: lm>
Bahrain	<tibble>	<S3: lm>
Bangladesh	<tibble>	<S3: lm>
Belgium	<tibble>	<S3: lm>





```
gapminder_nested$model[[1]]
```

country <fctr>	data <list>	model <list>
Afghanistan	<tibble>	<S3: lm>
Albania	<tibble>	<S3: lm>
Algeria	<tibble>	<S3: lm>
Angola	<tibble>	<S3: lm>
Argentina	<tibble>	<S3: lm>
Australia	<tibble>	<S3: lm>
Austria	<tibble>	<S3: lm>
Bahrain	<tibble>	<S3: lm>
Bangladesh	<tibble>	<S3: lm>
Belgium	<tibble>	<S3: lm>

```
Call:
lm(formula = lifeExp ~ year, data = x)

Coefficients:
(Intercept)         year
  -507.5343       0.2753
```

```
get_rsq <- function(mod) glance(mod)$r.squared

gapminder_nested <- gapminder_nested %>%
  mutate(r.squared = map_dbl(model, get_rsq))
```

country <fctr>	data <list>	model <list>	r.squared <dbl>
Afghanistan	<tibble>	<S3: lm>	0.94771226
Albania	<tibble>	<S3: lm>	0.91057777
Algeria	<tibble>	<S3: lm>	0.98511721
Angola	<tibble>	<S3: lm>	0.88781463
Argentina	<tibble>	<S3: lm>	0.99556810
Australia	<tibble>	<S3: lm>	0.97964774
Austria	<tibble>	<S3: lm>	0.99213401
Bahrain	<tibble>	<S3: lm>	0.96673981
Bangladesh	<tibble>	<S3: lm>	0.98936087
Belgium	<tibble>	<S3: lm>	0.99454056

map_dbl() takes a list

…and returns a number

Your Turn 2

Run the chunk then,

filter `gapminder_nested` to find the countries with `r.squared` less than 0.5.

```
gapminder_nested %>%
  filter(r.squared < 0.5)
```

But how can we plot these?

country <fctr>	data <list>	model <list>	r.squared <dbl>
Botswana	<tibble>	<S3: lm>	0.03402340
Central African Republic	<tibble>	<S3: lm>	0.49324448
Congo, Dem. Rep.	<tibble>	<S3: lm>	0.34820278
Cote d'Ivoire	<tibble>	<S3: lm>	0.28337240
Kenya	<tibble>	<S3: lm>	0.44255729
Lesotho	<tibble>	<S3: lm>	0.08485635
Namibia	<tibble>	<S3: lm>	0.43702163
Rwanda	<tibble>	<S3: lm>	0.01715964
South Africa	<tibble>	<S3: lm>	0.31246865
Swaziland	<tibble>	<S3: lm>	0.06821087

1-10 of 13 rows

unnest()

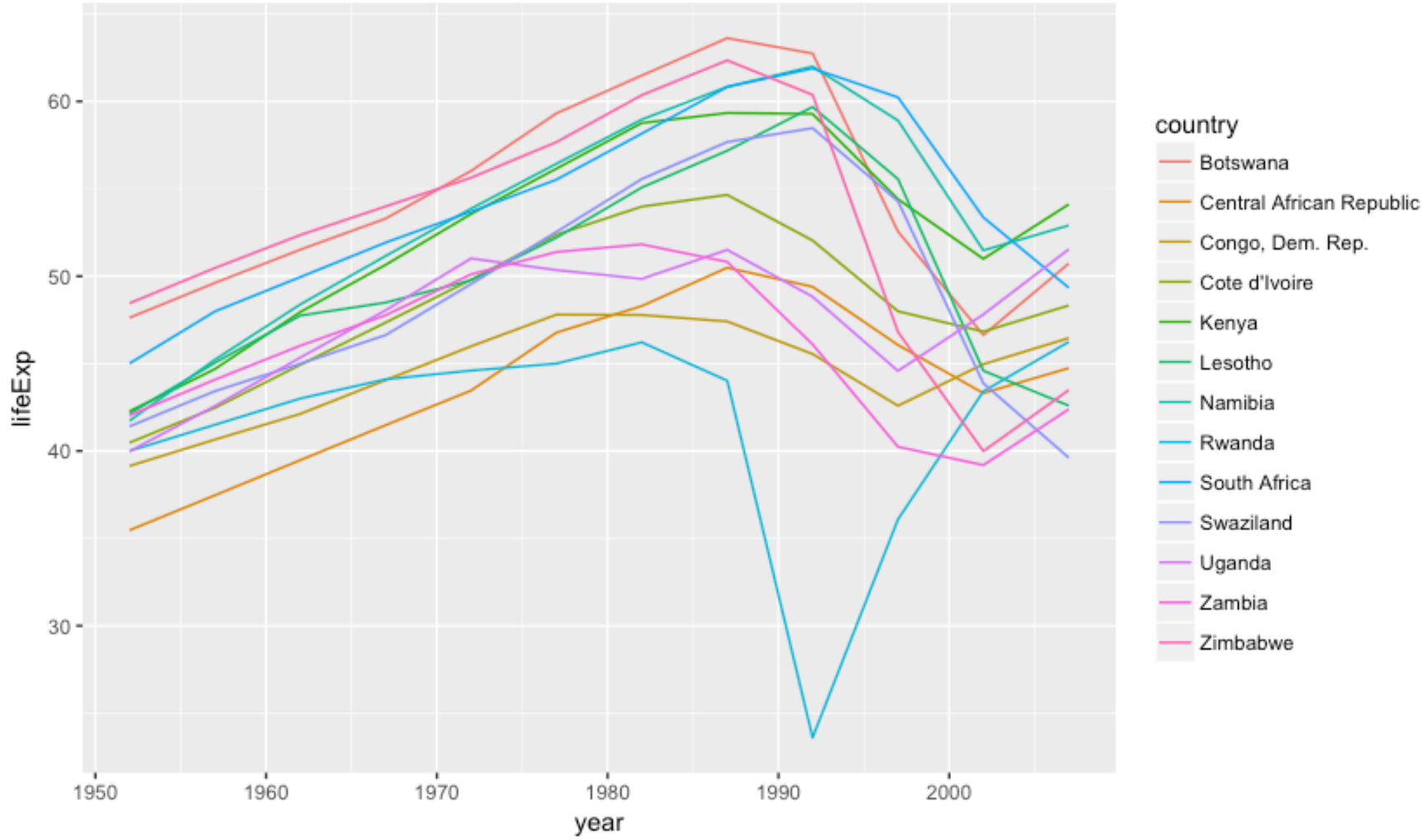
```
poor_fit <- gapminder_nested %>%
  filter(r.squared < 0.5)

gapminder_nested %>% unnest(data)
```

Column to unnest

country <fctr>	r.squared <dbl>	continent <fctr>	year <int>	lifeExp <dbl>	pop <int>
Botswana	0.03402340	Africa	1952	47.622	442308
Botswana	0.03402340	Africa	1957	49.618	474639
Botswana	0.03402340	Africa	1962	51.520	512764
Botswana	0.03402340	Africa	1967	53.298	553541
Botswana	0.03402340	Africa	1972	56.024	619351

Columns from inside data

```
unnest(poor_fit, data) %>%
  ggplot(aes(x = year, y = lifeExp)) +
    geom_line(aes(color = country))
```



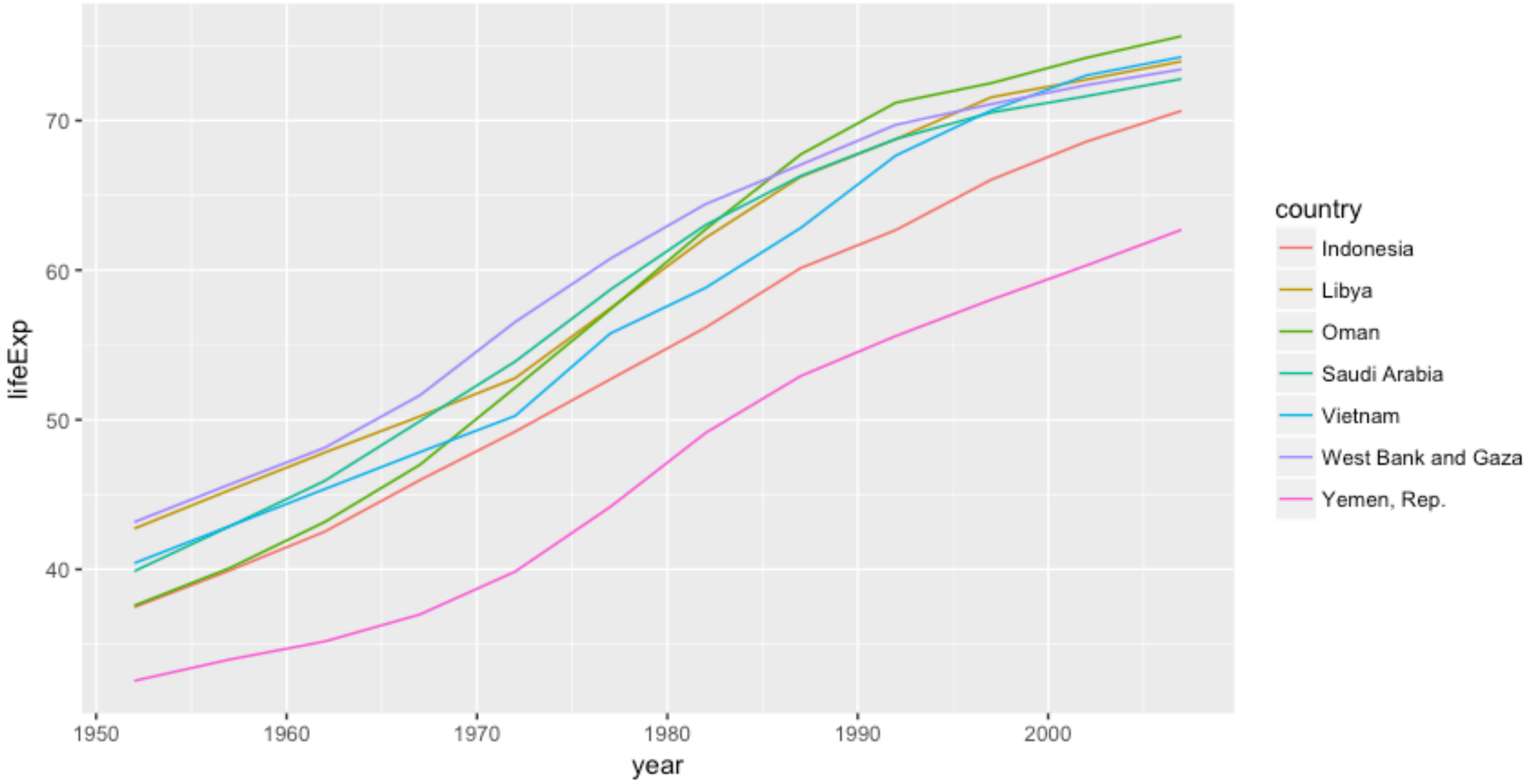
Your Turn 3

Edit the code in the chunk provided to instead find and plot countries with a slope above 0.6 years/year.

I've provided a `get_slope()` function:

```
get_slope <- function(mod) {
  tidy(mod) %>% filter(term == "year") %>% pull(estimate)
}
```

```
gapminder_nested <- gapminder_nested %>%
  mutate(slope = map_dbl(model, get_slope))

big_slope <- gapminder_nested %>%
  filter(slope > 0.6)

unnest(big_slope, data) %>%
  ggplot(aes(x = year, y = lifeExp)) +
    geom_line(aes(color = country))
```



Take Away

A table is …an organizational structure …that you can manipulate.

country	r.squared	data	model
Botswana	0.03	year .resid 1952 -5.3071154 1957 -3.6144580 1962 -2.0158007 1967 -0.5411434 1972 1.8815140 1977 4.8731713 1982 6.7348287 1987 8.5694860 1992 7.3891434 1997 -3.1031993 2002 -9.3285420 2007 -5.5378846	Call: lm(formula = lifeExp ~ year, data = .) Coefficients: (Intercept) year -65.49586 0.06067
Lesotho	0.08	year .resid 1952 -5.2410256 1957 -2.8098543 1962 -0.5876830 1967 -0.3205117 1972 0.4766597 1977 2.4398310 1982 4.8320023 1987 6.4561737 1992 8.4833450 1997 3.8785163 2002 -7.5643124 2007 -10.0431410	Call: lm(formula = lifeExp ~ year, data = .) Coefficients: (Intercept) year -139.16529 0.09557

Benefits

Data and models stay in correspondence across manipulations

```
gapminder_nested %>% filter(str_sub(country, 1, 1) == "P")
```

country <fctr>	data <list>	model <list>	r.squared <dbl>	slope <dbl>
Pakistan	<tibble>	<S3: lm>	0.9972497	0.4057923
Panama	<tibble>	<S3: lm>	0.9511952	0.3542091
Paraguay	<tibble>	<S3: lm>	0.9829865	0.1573545
Peru	<tibble>	<S3: lm>	0.9884740	0.5276979
Philippines	<tibble>	<S3: lm>	0.9914226	0.4204692
Poland	<tibble>	<S3: lm>	0.8396631	0.1962189
Portugal	<tibble>	<S3: lm>	0.9690351	0.3372014
Puerto Rico	<tibble>	<S3: lm>	0.9078191	0.2105748

8 rows

Adapted from

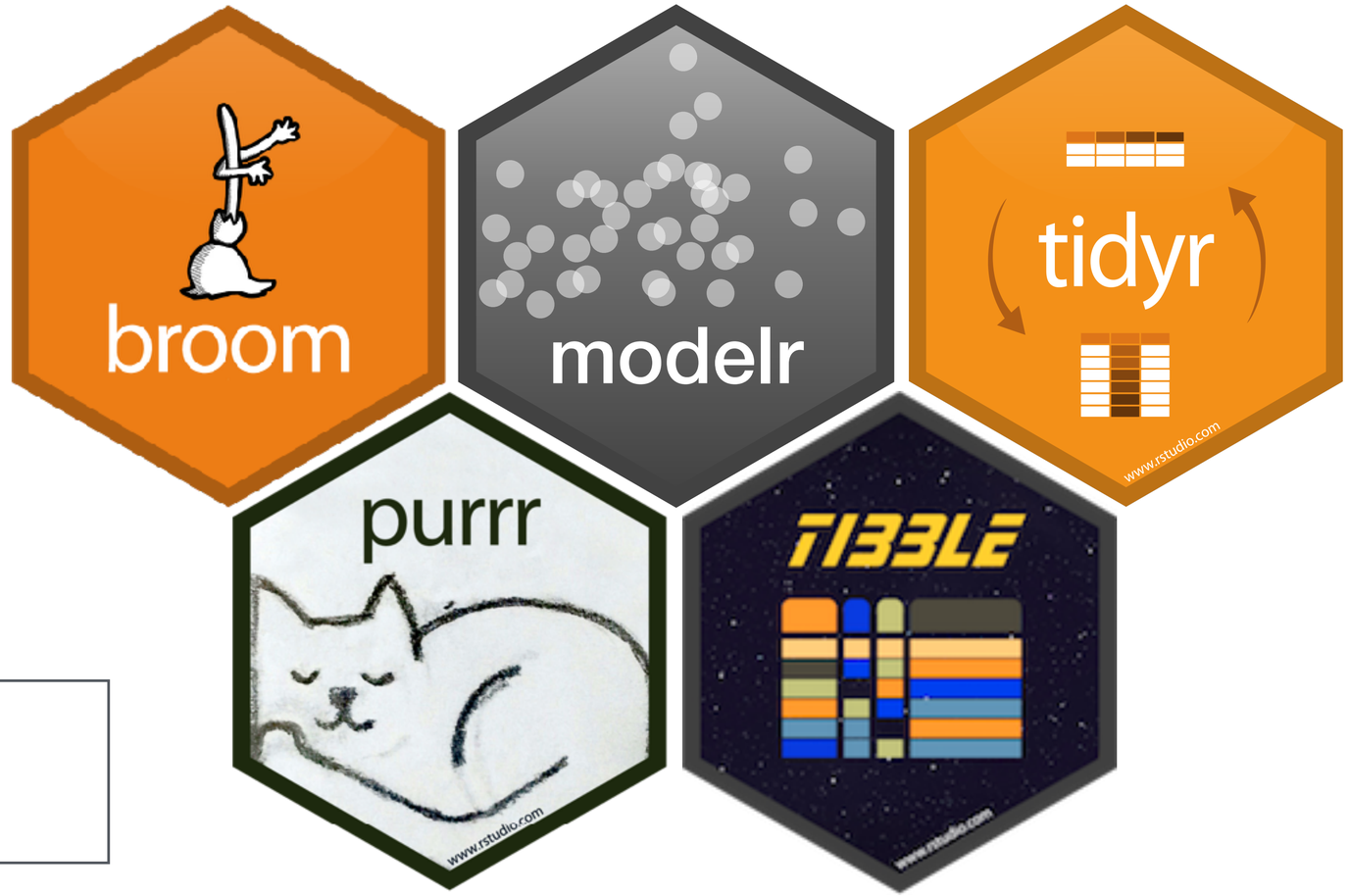
Your Turn 4

Challenge:

1. Create your own copy of `gapminder_nested` and then add one more list column: **output** which contains the output of **augment()** for each model.

2. Plot the residuals against time for the countries with large r-squared.

Organize with list columns



In R4DS

Many Models