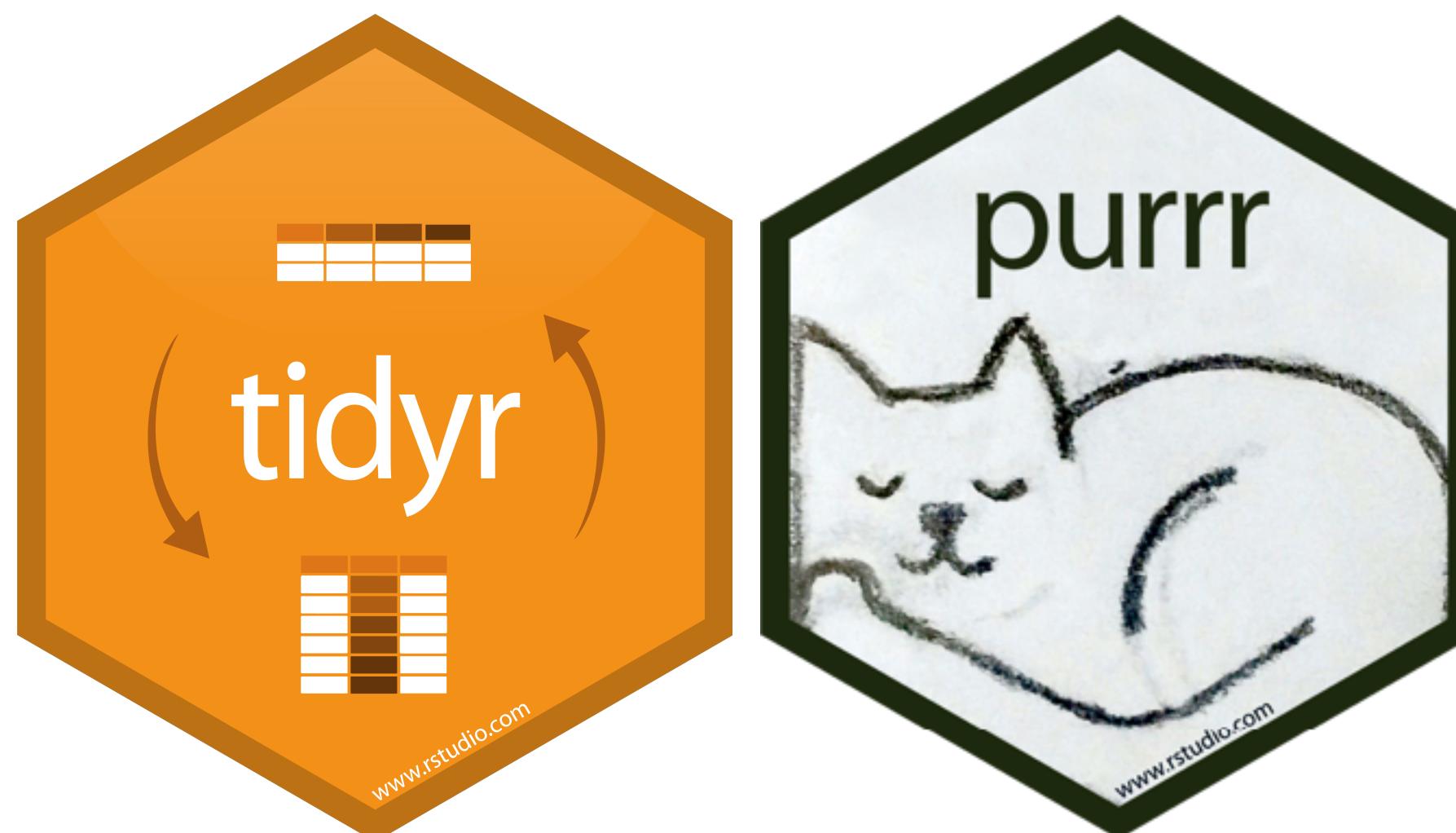


# List Columns



**Open 08-List-Columns.Rmd**

# gapminder



A subset of the data available at Hans Rosling's [gapminder.org](http://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

1-10 of 1,704 rows

Previous 1 2 3 4 5 6 ... 100 Next



# Your Turn 1

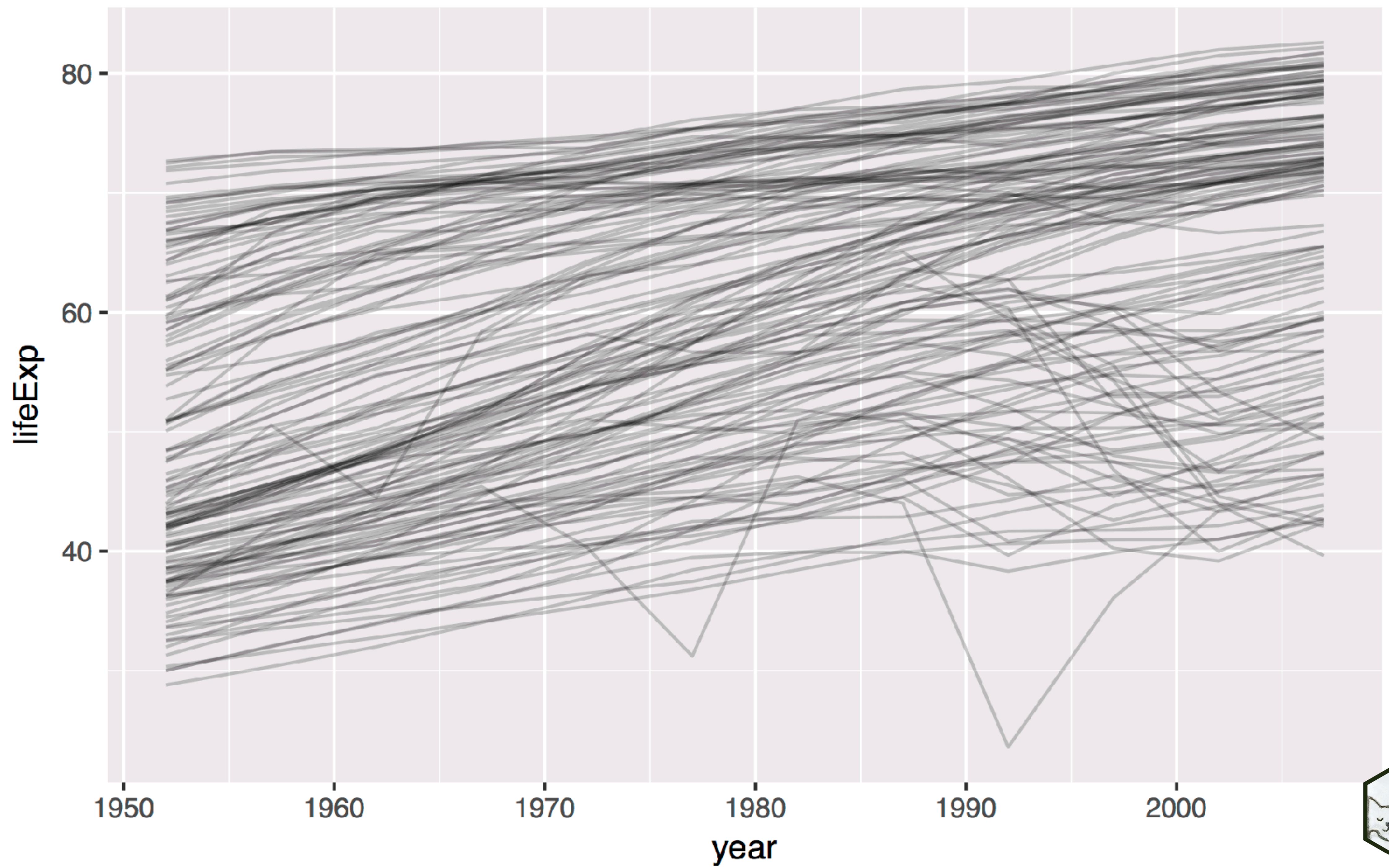
How has life expectancy changed since 1952?

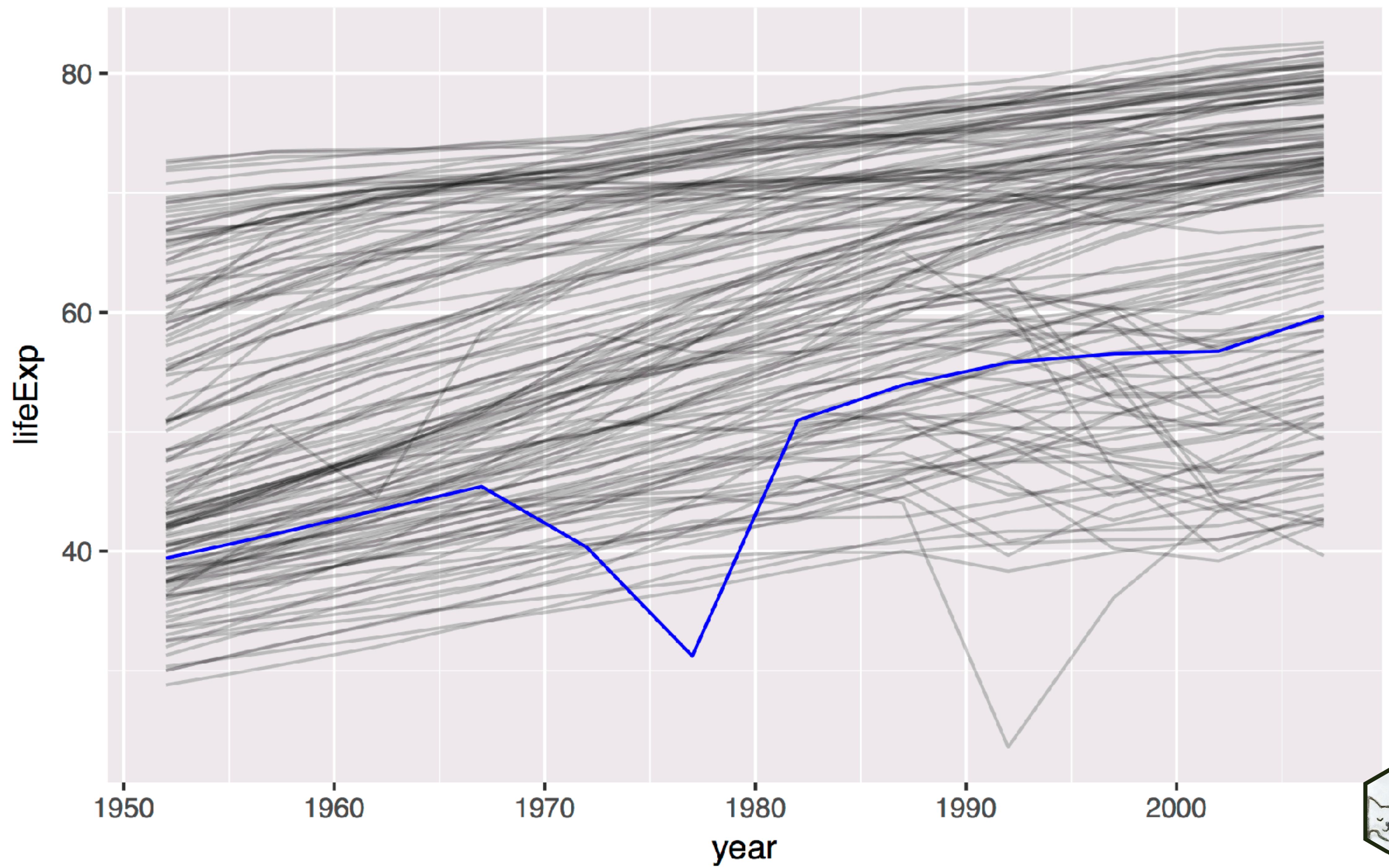
Make a line plot of **lifeExp** vs. **year** grouped by **country**. Set alpha to 0.2, to see the results better.

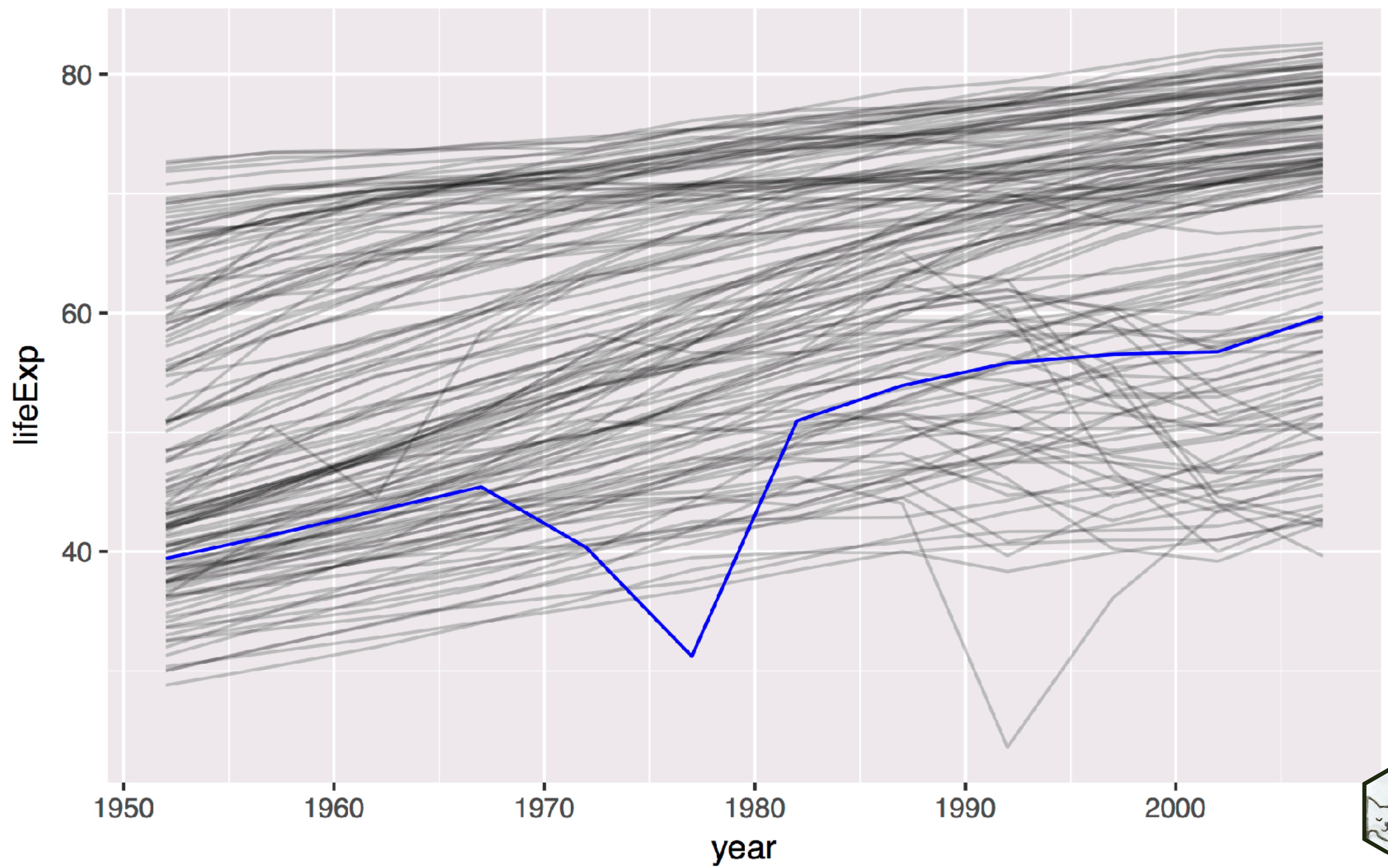


```
gapminder %>%  
  ggplot(mapping = aes(x = year, y = lifeExp, group = country)) +  
  geom_line(alpha = 0.2)
```





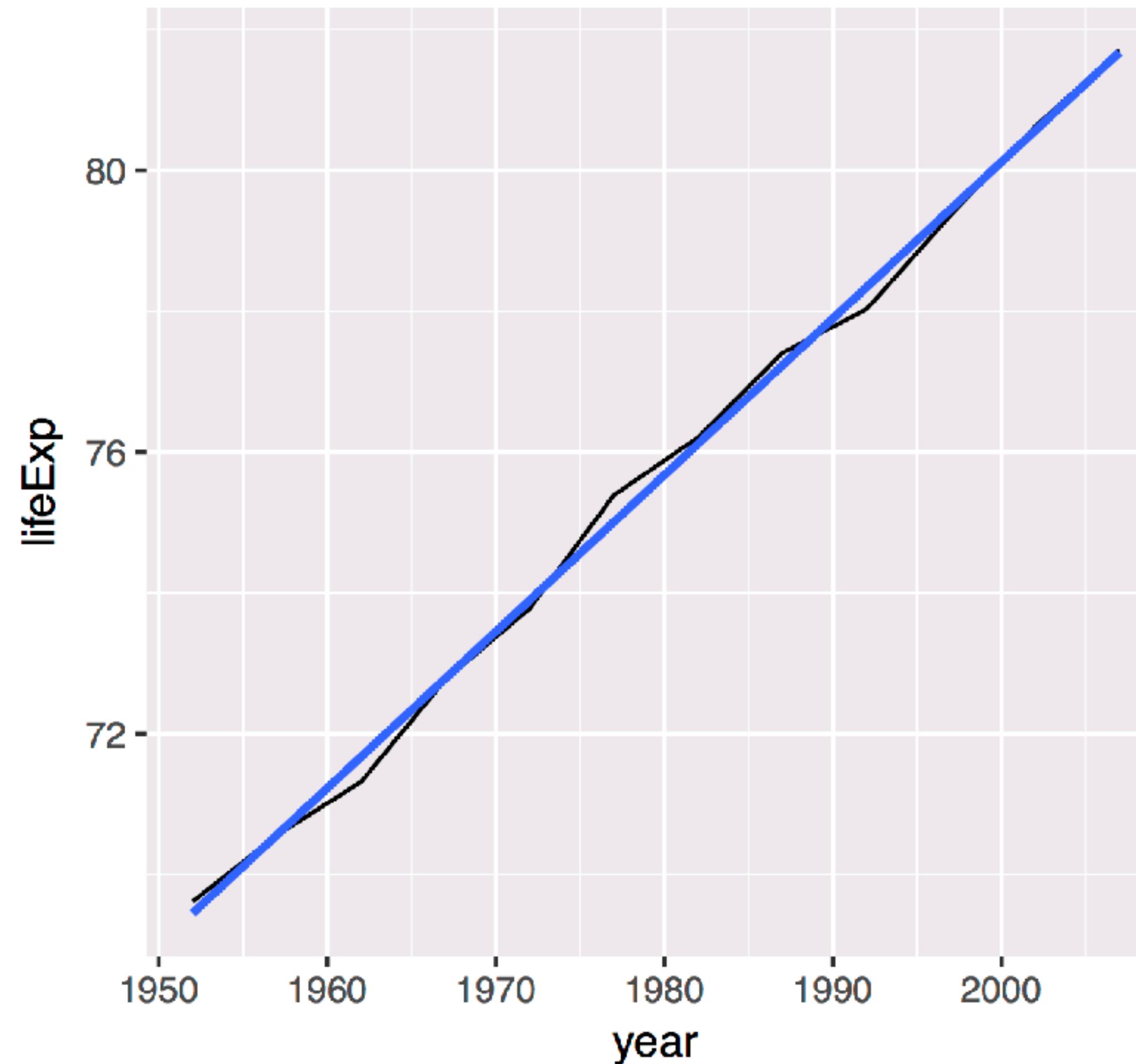




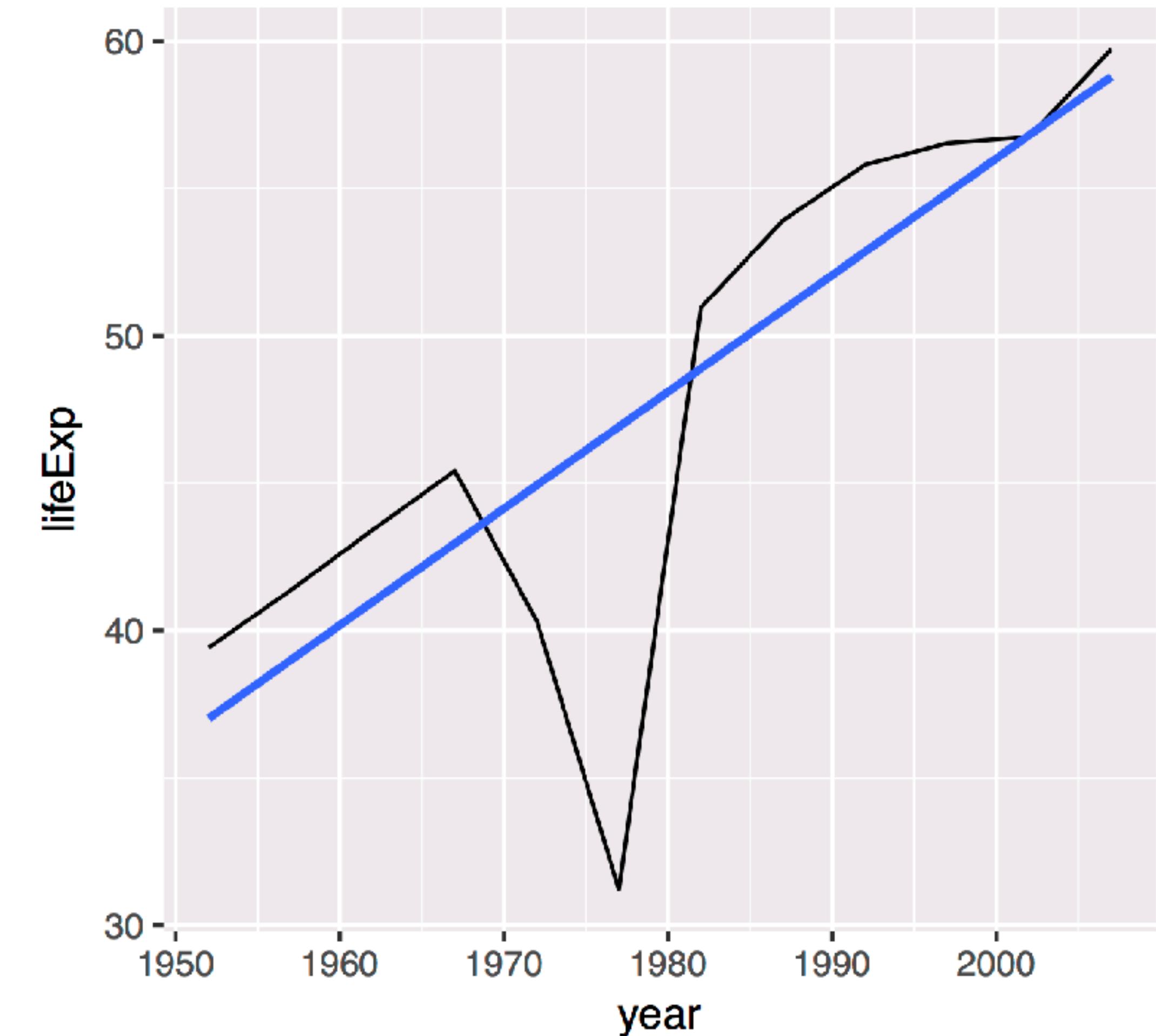
# Idea 1

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

Switzerland, R Squared = 0.99



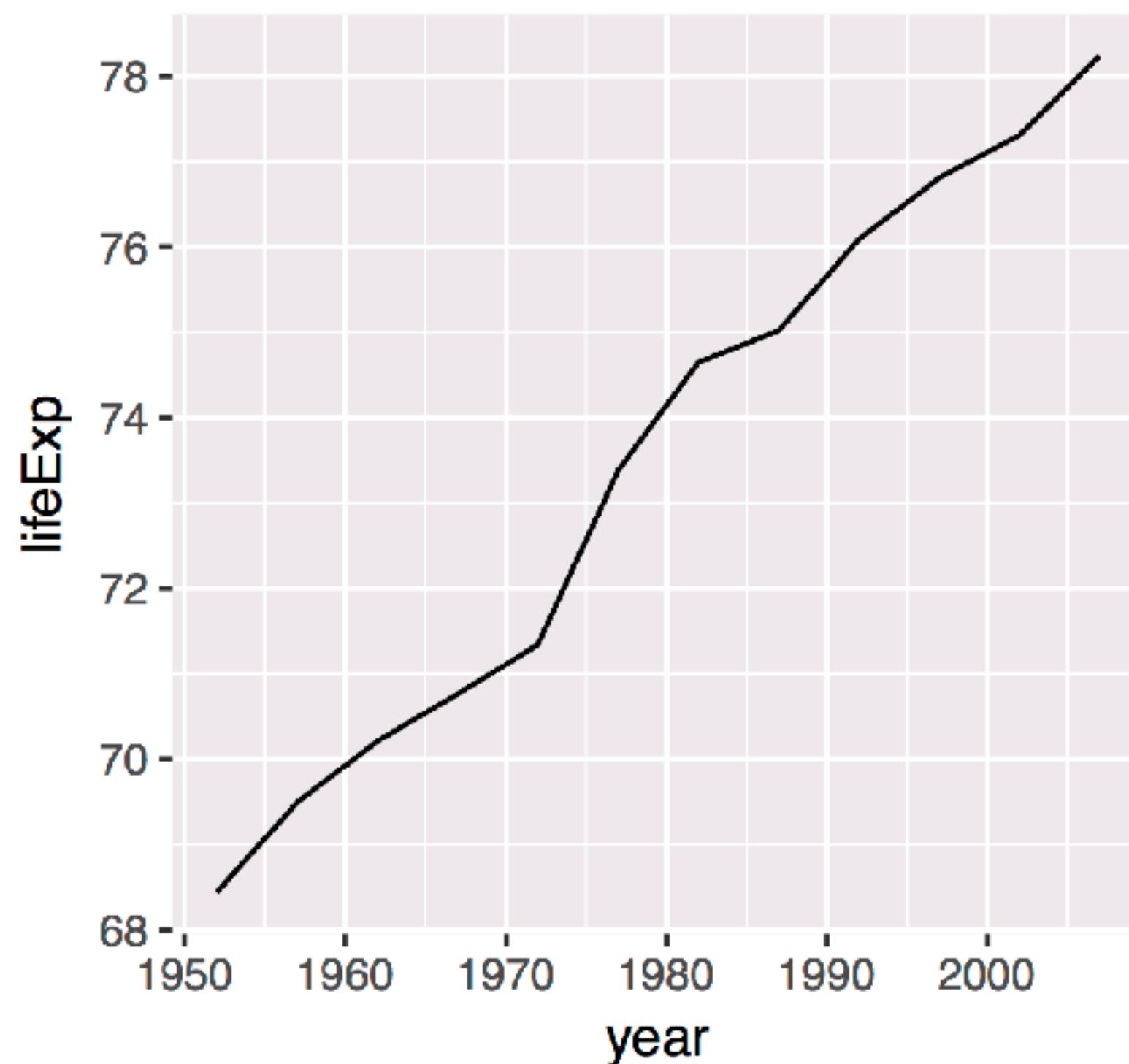
Cambodia, R Squared = 0.63



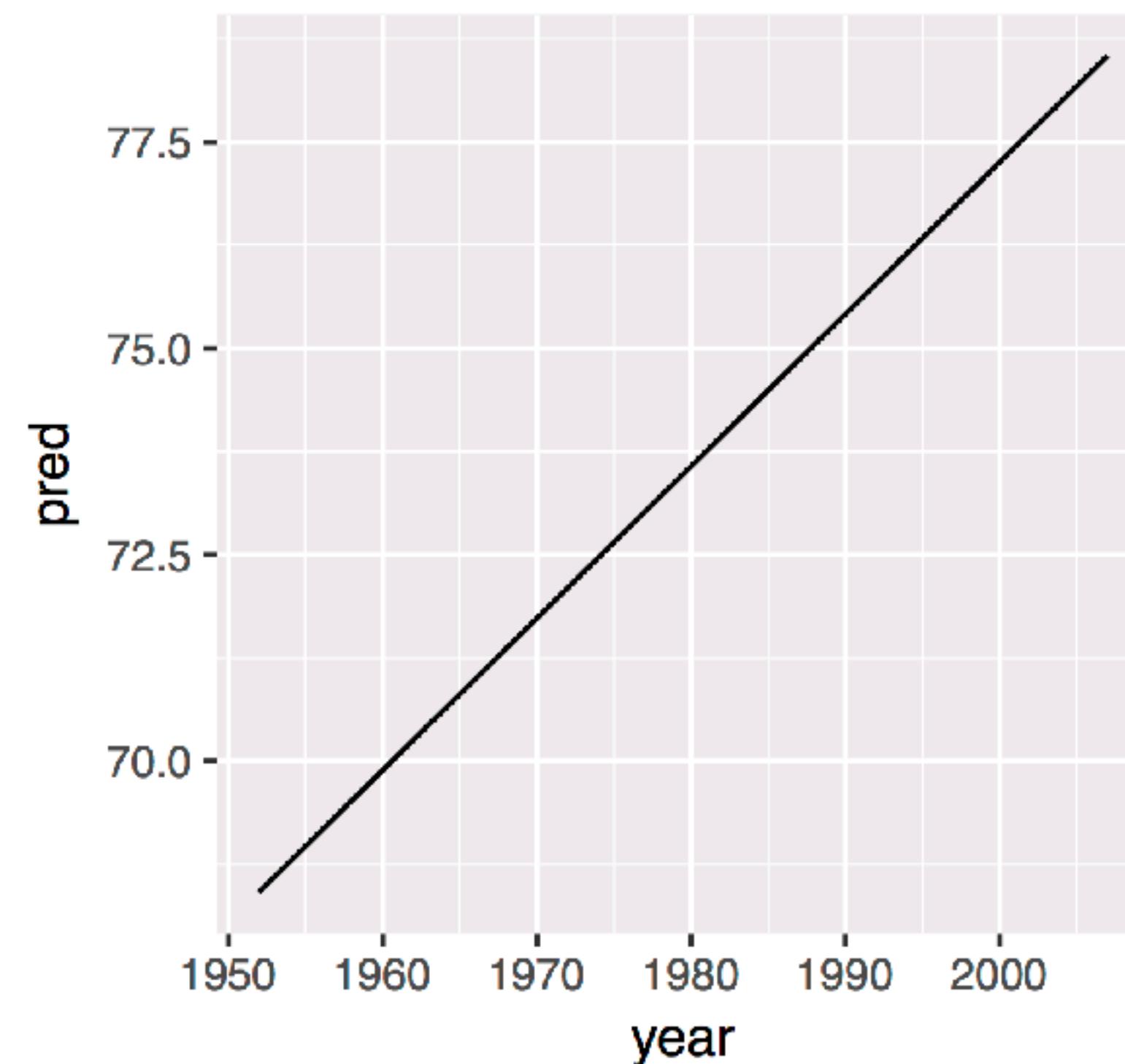
# Idea 2

Use residuals to study deviations from the trend to study country specific patterns.

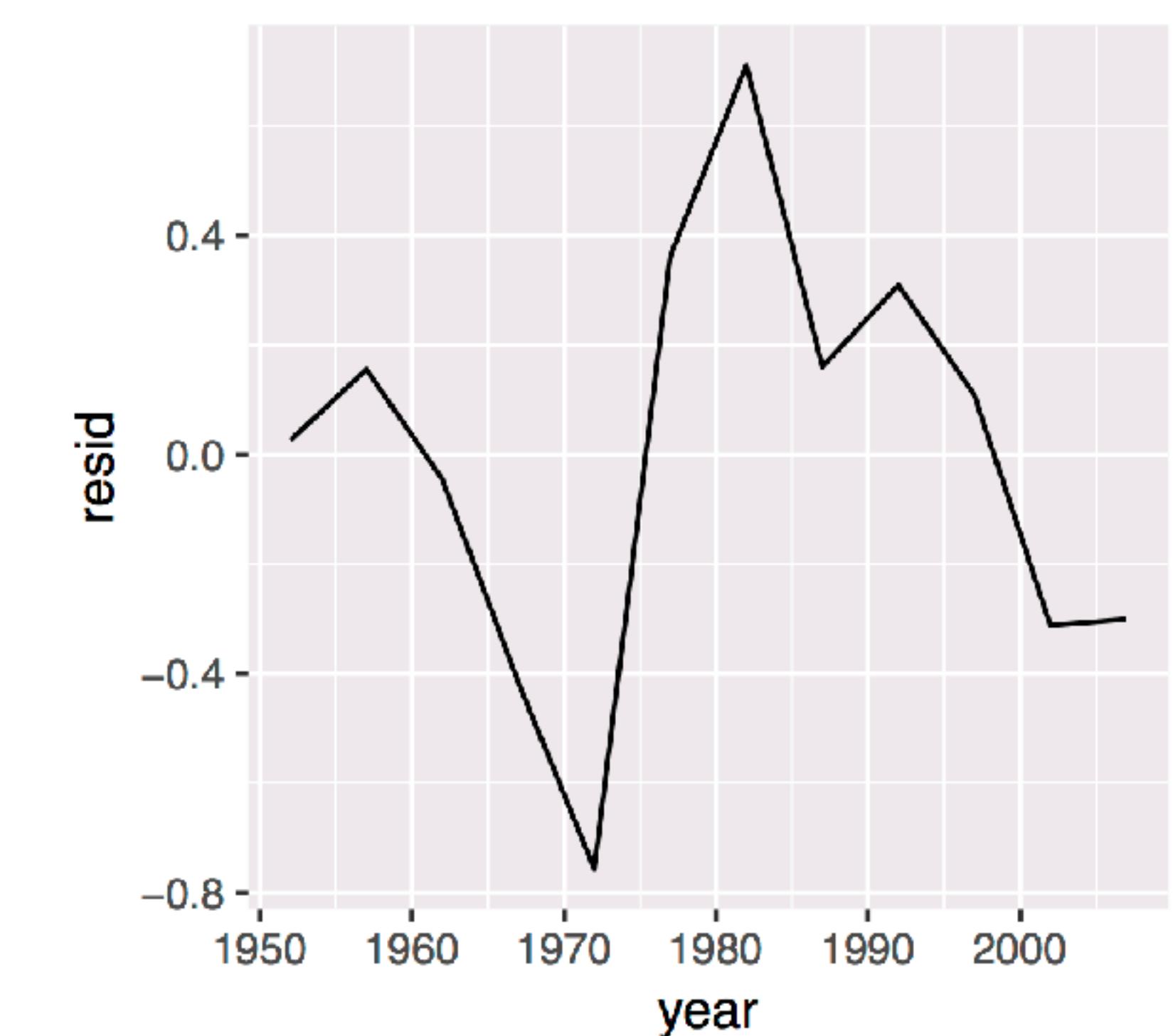
Full Data =



Linear Trend +



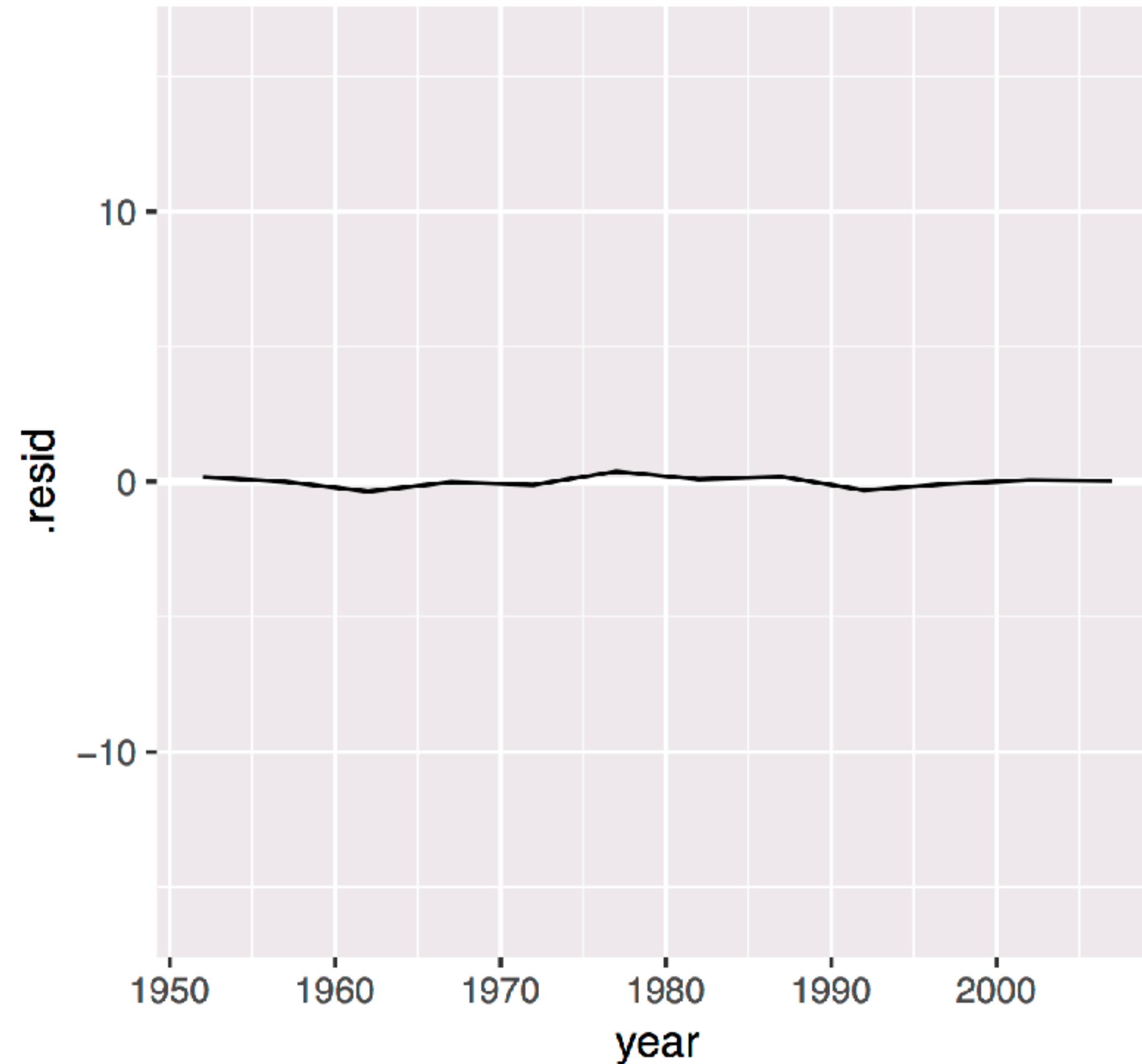
Remaining Pattern



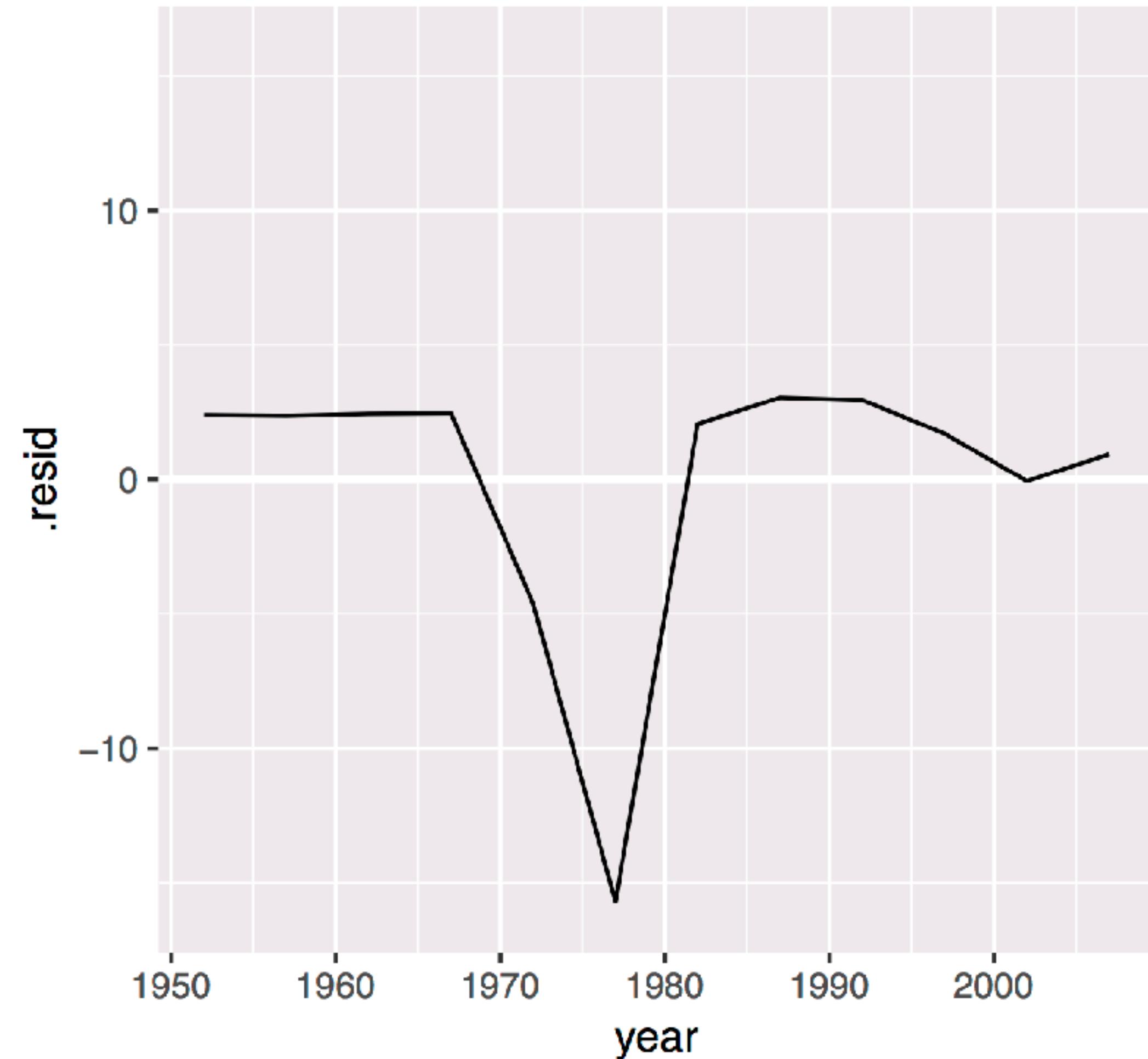
# Idea 2

Use residuals to study deviations from the trend to study country specific patterns.

Switzerland residuals



Cambodia residuals



# Goal

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

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



# dplyr do()



# group\_by() + do()

Run an expression on each group. The expression should return a data frame.

```
do(data, expression)
```

A grouped  
data frame

An R expression  
that returns a  
data frame

Use a ":" in the  
expression to pass  
input (like a pipe)



# group\_by() + do()

Run an expression on each group. The expression should return a data frame.

```
gapminder %>%  
  group_by(country, continent) %>%  
  do()
```

A grouped  
data frame



# group\_by() + do()

Run an expression on each group. The expression should return a data frame.

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .))
```

A grouped  
data frame

An R expression that  
uses a ":" to pass input

but DOES NOT return  
a data frame



# group\_by() + do()

Run an expression on each group. The expression should return a data frame.

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(glance(lm(lifeExp ~ year, data = .)))
```

A grouped  
data frame

An R expression that  
uses a ":" to pass input

...and returns a  
data frame



# group\_by() + do()

Run an expression on each group. The expression should return a data frame.

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% glance()
```

A grouped  
data frame

An R expression that  
uses a ":" to pass input

...and returns a  
data frame



# group\_by() + do()

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% glance()
```

country <fctr>	continent <fctr>	r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>
Afghanistan	Asia	0.94771226	0.942483483	1.2227880	181.2494098
Albania	Europe	0.91057777	0.901635545	1.9830615	101.8290138
Algeria	Africa	0.98511721	0.983628932	1.3230064	661.9170864
Angola	Africa	0.88781463	0.876596093	1.4070091	79.1381823
Argentina	Americas	0.99556810	0.995124905	0.2923072	2246.3663487
Australia	Oceania	0.97964774	0.977612511	0.6206086	481.3458627
Austria	Europe	0.99213401	0.991347414	0.4074094	1261.2962902
Bahrain	Asia	0.96673981	0.963413791	1.6395865	290.6597394
Bangladesh	Asia	0.98936087	0.988296956	0.9766908	929.9263688
Belgium	Europe	0.99454056	0.993994612	0.2929025	1821.6883955

1-10 of 142 rows | 1-6 of 13 columns

Previous 1 2 3 4 5 6 ... 15 Next



# group\_by() + do()

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% glance()
```

country <fctr>	continent <fctr>	r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>
Afghanistan	Asia	0.94771226	0.942483483	1.2227880	181.2494098
Albania	Europe	0.91057777	0.901635545	1.9830615	101.8290138
Algeria	Africa	0.98511721	0.983628932	1.3230064	661.9170864
Angola	Africa	0.88781463	0.876596093	1.4070091	79.1381823
Argentina	Americas	0.99556810	0.995124905	0.2923072	2246.3663487
Australia	Oceania	0.97964774	0.977612511	0.6206086	481.3458627
Austria	Europe	0.99213401	0.991347414	0.4074094	1261.2962902
Bahrain	Asia	0.96673981	0.963413791	1.6395865	290.6597394
Bangladesh	Asia	0.98936087	0.988296956	0.9766908	929.9263688
Belgium	Europe	0.99454056	0.993994612	0.2929025	1821.6883955

1-10 of 142 rows | 1-6 of 13 columns

Previous 1 2 3 4 5 6 ... 15 Next



# group\_by() + do()

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% tidy()
```

country	continent	term	estimate	std.error
<fctr>	<fctr>	<chr>	<dbl>	<dbl>
Afghanistan	Asia	(Intercept)	-5.075343e+02	4.048416e+01
Afghanistan	Asia	year	2.753287e-01	2.045093e-02
Albania	Europe	(Intercept)	-5.940725e+02	6.565536e+01
Albania	Europe	year	3.346832e-01	3.316639e-02
Algeria	Africa	(Intercept)	-1.067859e+03	4.380220e+01
Algeria	Africa	year	5.692797e-01	2.212707e-02
Angola	Africa	(Intercept)	-3.765048e+02	4.658337e+01
Angola	Africa	year	2.093399e-01	2.353200e-02
Argentina	Americas	(Intercept)	-3.896063e+02	9.677730e+00
Argentina	Americas	year	2.317084e-01	4.888791e-03

1-10 of 284 rows | 1-5 of 7 columns

Previous 1 2 3 4 5 6 ... 29 Next



# group\_by() + do()

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% augment()
```

country	continent	lifeExp	year	.fitted	.se.fit	.resid	.hat
<fctr>	<fctr>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>
Afghanistan	Asia	28.80100	1952	29.90729	0.66399954	-1.106295e+00	0.29487179
Afghanistan	Asia	30.33200	1957	31.28394	0.57994415	-9.519382e-01	0.22494172
Afghanistan	Asia	31.99700	1962	32.66058	0.50267990	-6.635816e-01	0.16899767
Afghanistan	Asia	34.02000	1967	34.03722	0.43583366	-1.722494e-02	0.12703963
Afghanistan	Asia	36.08800	1972	35.41387	0.38487259	6.741317e-01	0.09906760
Afghanistan	Asia	38.43800	1977	36.79051	0.35667194	1.647488e+00	0.08508159
Afghanistan	Asia	39.85400	1982	38.16716	0.35667194	1.686845e+00	0.08508159
Afghanistan	Asia	40.82200	1987	39.54380	0.38487259	1.278202e+00	0.09906760
Afghanistan	Asia	41.67400	1992	40.92044	0.43583366	7.535583e-01	0.12703963
Afghanistan	Asia	41.76300	1997	42.29709	0.50267990	-5.340851e-01	0.16899767

1-10 of 1,704 rows | 1-8 of 11 columns

Previous 1 2 3 4 5 6 ... 100 Next



# Your Turn 2

Group the data by **country** and **continent** then fit a model and collect the residuals *for each country*.

Plot the **residuals** vs **year** as a line graph, grouped by **country**, with alpha = 0.2.

Add the following to your plot to facet by **continent**:

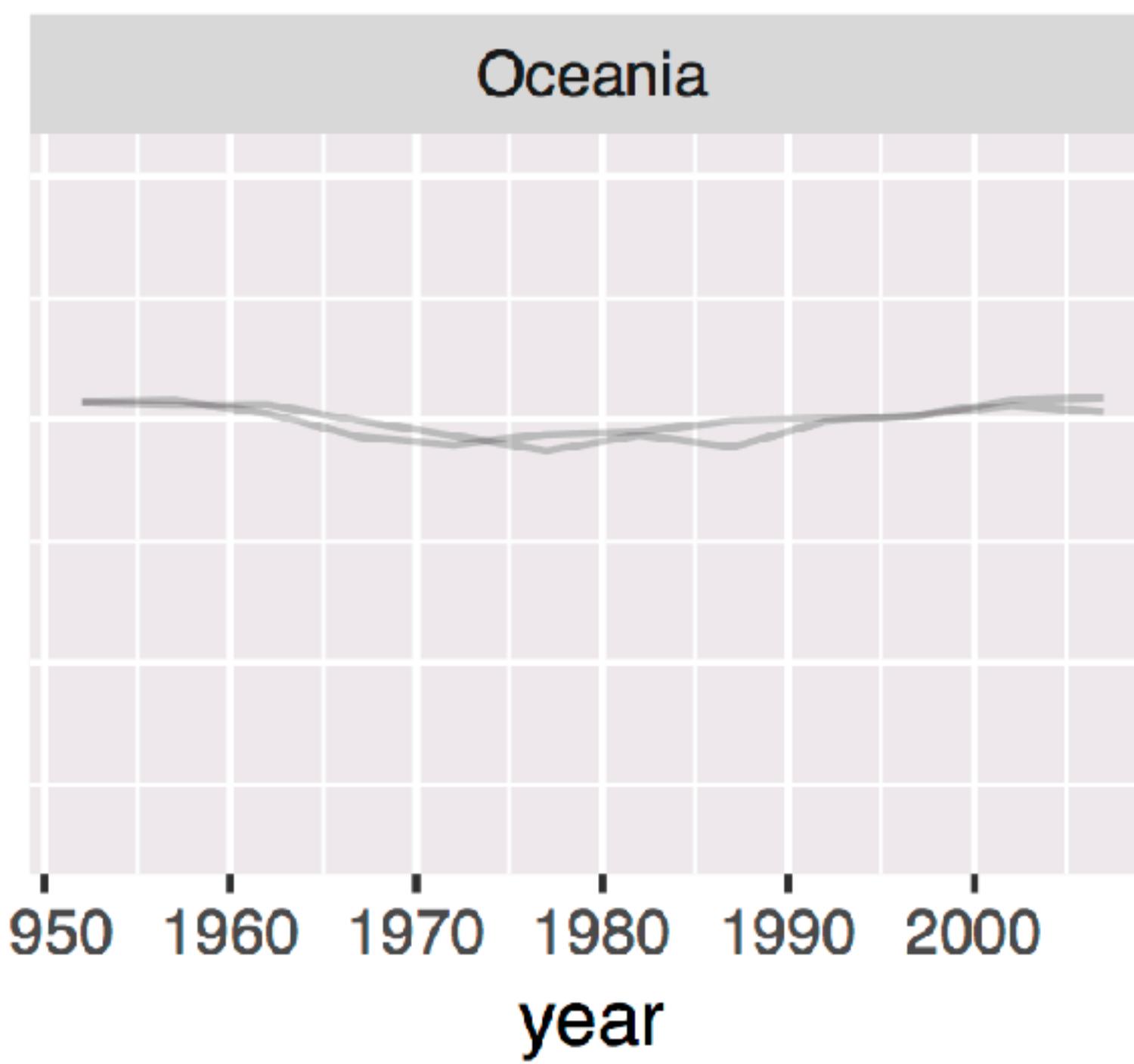
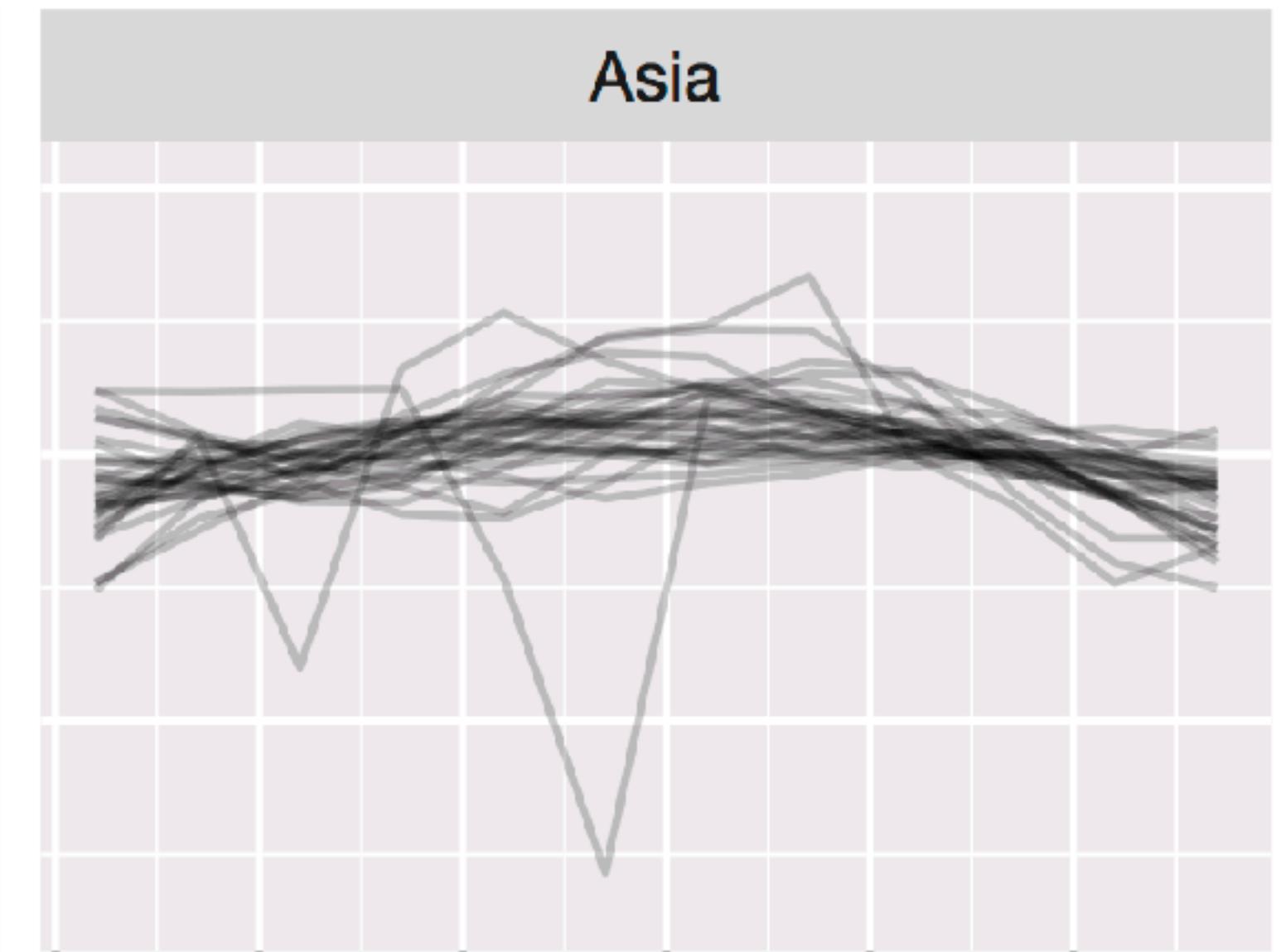
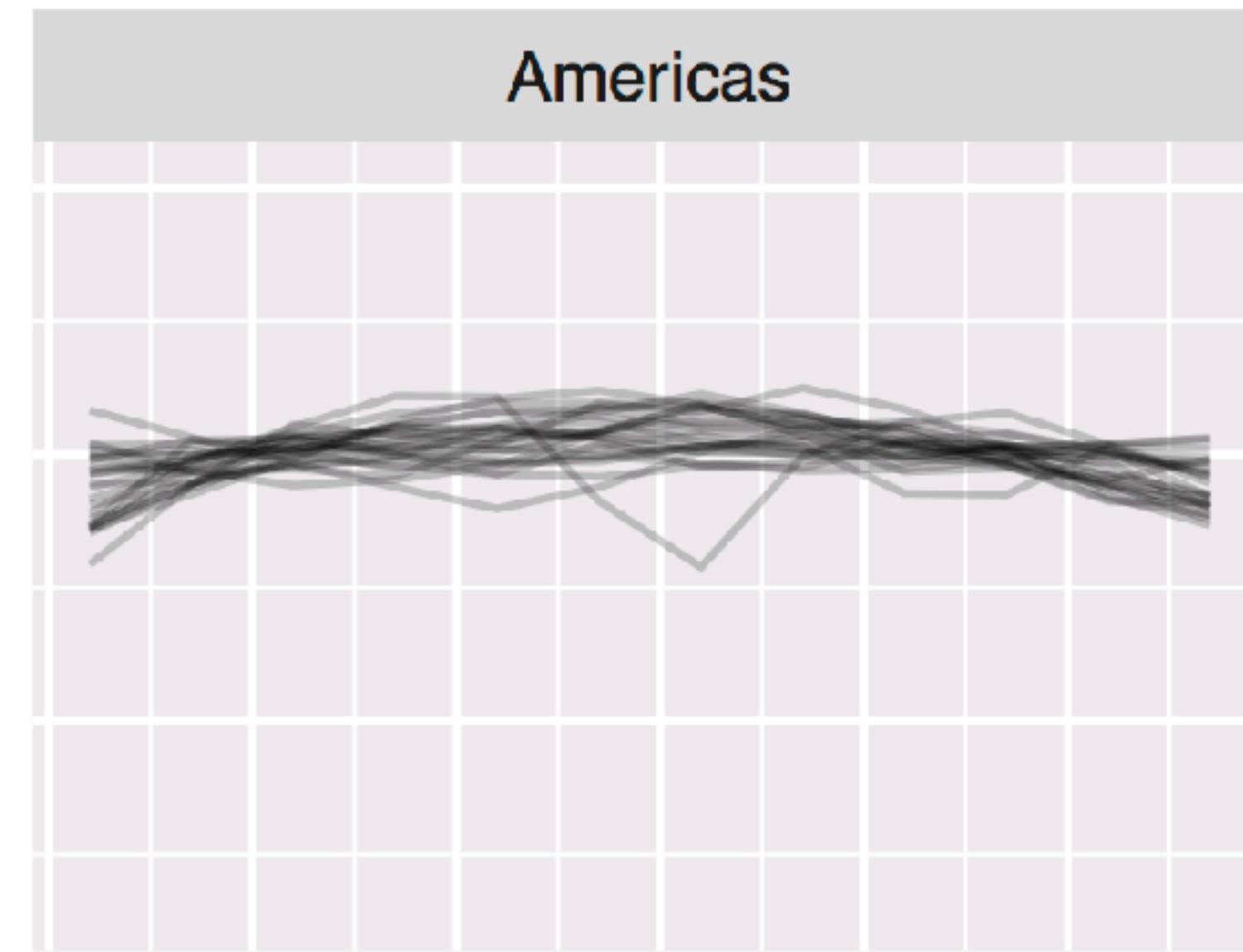
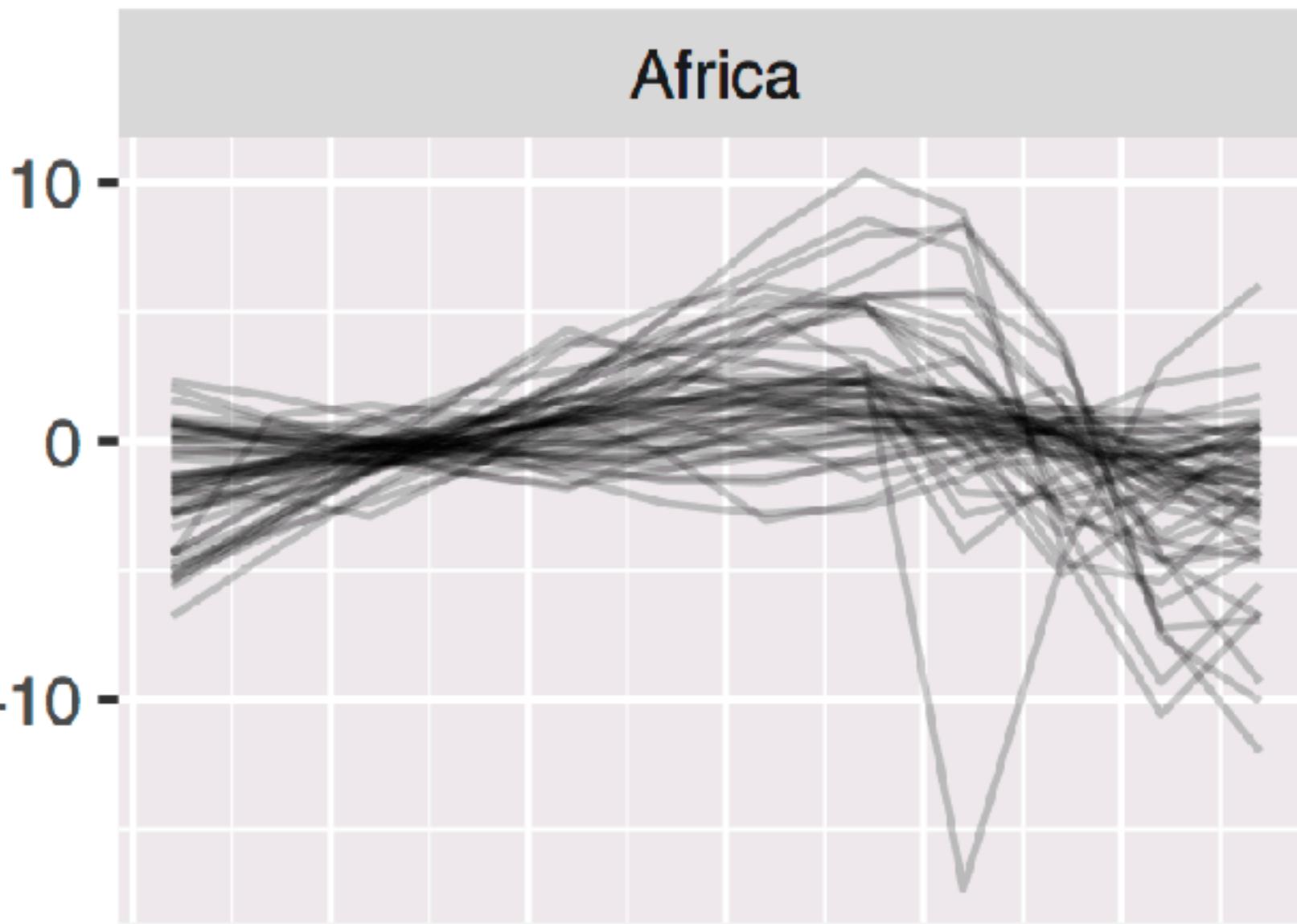
+ facet\_wrap(~continent)



```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .) %>% augment()) %>%  
  ggplot() +  
    geom_line(aes(year, .resid, group = country), alpha = 0.2) +  
    facet_wrap(~continent)
```



resid



1950 1960 1970 1980 1990 2000

year

1950 1960 1970 1980 1990 2000



# Quiz

Which broom function can we use to find the **r squared** for each model?

# Bad fits

Which countries had a non-linear progress?

```
gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% glance() %>%  
  filter(r.squared < 0.25)
```

country <fctr>	continent <fctr>	r.squared <dbl>	adj.r.squared <dbl>	sigma <dbl>	statistic <dbl>	►
Botswana	Africa	0.03402340	-0.062574259	6.112177	0.3522177	
Lesotho	Africa	0.08485635	-0.006658011	5.933934	0.9272463	
Rwanda	Africa	0.01715964	-0.081124401	6.558269	0.1745923	
Swaziland	Africa	0.06821087	-0.024968046	6.644091	0.7320419	
Zambia	Africa	0.05983644	-0.034179918	4.528713	0.6364471	
Zimbabwe	Africa	0.05623196	-0.038144842	7.205431	0.5958240	



# Your Turn 3

Complete the code to filter **residuals**, which is the dataset that you made in Your Turn 2 ,against **bad\_fits**.

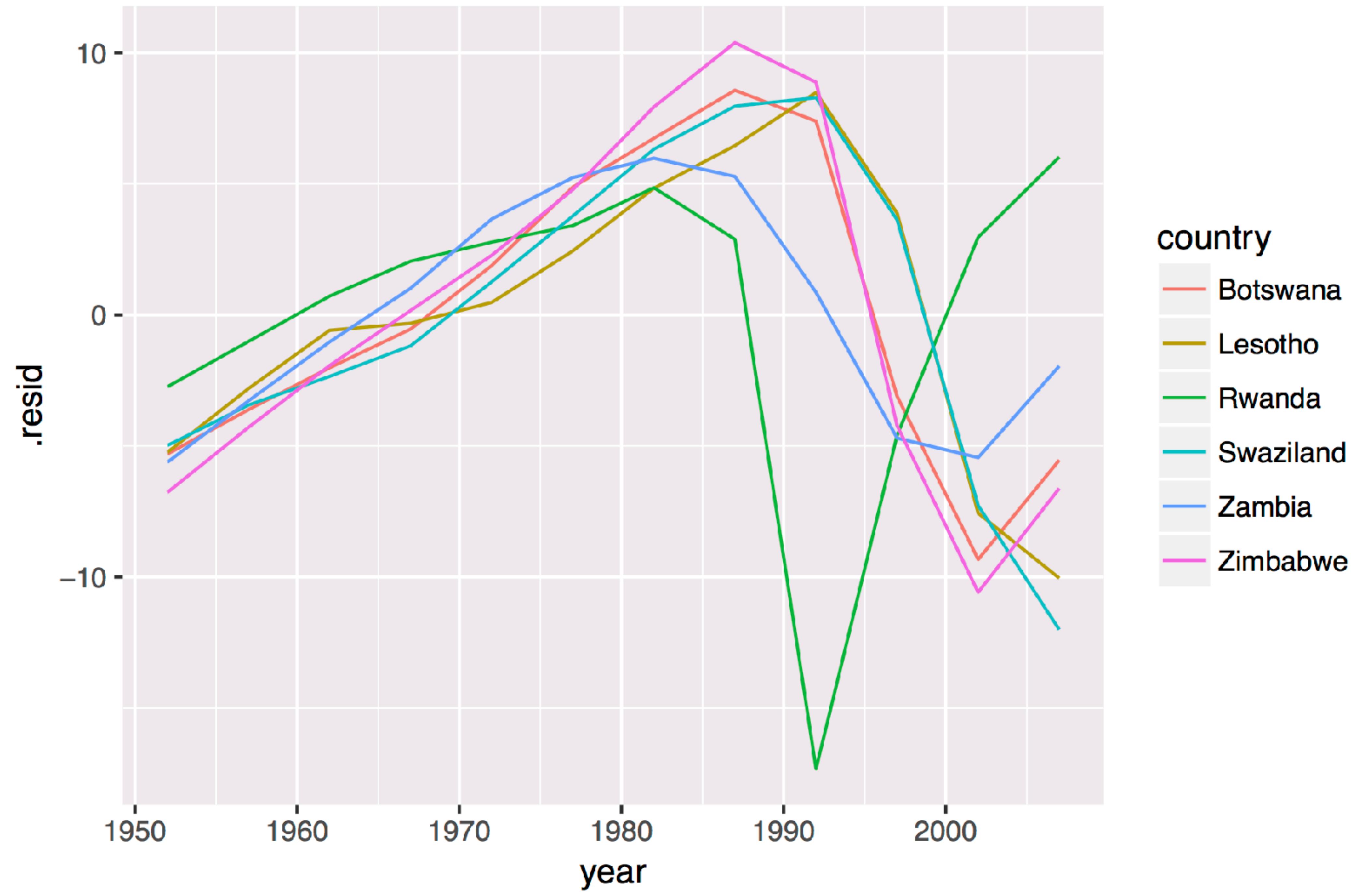
Use the result to plot a line graph of **year** vs. **.resid** colored by **country** for each country that had an r-squared < 0.25.



```
bad_fits <- gapminder %>%  
  group_by(country) %>%  
  do(lm(lifeExp ~ year, data = .) %>% glance()) %>%  
  filter(r.squared < 0.25)
```

```
residuals <- gapminder %>%  
  group_by(country) %>%  
  do(lm(lifeExp ~ year, data = .) %>% augment())
```

```
residuals %>%  
  semi_join(bad_fits) %>%  
  ggplot() +  
    geom_line(aes(year, .resid, color = country))
```



```
bad_fits <- gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% glance() %>%  
  filter(r.squared < 0.25)
```

```
residuals <- gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>% augment()
```

```
residuals %>%  
  semi_join(bad_fits) %>%  
  ggplot() +  
    geom_line(aes(year, .resid, color = country))
```

Fits the same model twice (to 142 countries each time)

```
bad_fits <- gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>%  glance() %>%  
  filter(r.squared < 0.25)
```

Fits the same model twice (to 142 countries each time)

```
residuals <- gapminder %>%  
  group_by(country, continent) %>%  
  do(lm(lifeExp ~ year, data = .)) %>%  augment()
```

```
residuals %>%  
  semi_join(bad_fits) %>%  
  ggplot() +  
    geom_line(aes(year, .resid, color = country))
```

Need to keep track of two separate data frames

# List columns

R

# bad\_fits

country	r.squared
Botswana	0.03
Lesotho	0.08
Rwanda	0.02
Swaziland	0.07
Zambia	0.06
Zimbabwe	0.06

# residuals

country	year	.resid
Botswana	1952	-5.3071154
Botswana	1957	-3.6144580
Botswana	1962	-2.0158007
Botswana	1967	-0.5411434
Botswana	1972	1.8815140
Botswana	1977	4.8731713
Botswana	1982	6.7348287
Botswana	1987	8.5694860
Botswana	1992	7.3891434
Botswana	1997	-3.1031993
Botswana	2002	-9.3285420
Botswana	2007	-5.5378846
Lesotho	1952	-5.2410256
Lesotho	1957	-2.8098543
Lesotho	1962	-0.5876830
Lesotho	1967	-0.3205117
Lesotho	1972	0.4766597
Lesotho	1977	2.4398310
Lesotho	1982	4.8320023
Lesotho	1987	6.4561737
Lesotho	1992	8.4833450
Lesotho	1997	3.8785163
Lesotho	2002	-7.5643124
Lesotho	2007	-10.0431410
Rwanda	1952	-2.7419487
Rwanda	1957	-1.0127914
Rwanda	1962	0.7162660



# master

country	r.squared	data																										
Botswana	0.03	<table><thead><tr><th>year</th><th>.resid</th></tr></thead><tbody><tr><td>1952</td><td>-5.3071154</td></tr><tr><td>1957</td><td>-3.6144580</td></tr><tr><td>1962</td><td>-2.0158007</td></tr><tr><td>1967</td><td>-0.5411434</td></tr><tr><td>1972</td><td>1.8815140</td></tr><tr><td>1977</td><td>4.8731713</td></tr><tr><td>1982</td><td>6.7348287</td></tr><tr><td>1987</td><td>8.5694860</td></tr><tr><td>1992</td><td>7.3891434</td></tr><tr><td>1997</td><td>-3.1031993</td></tr><tr><td>2002</td><td>-9.3285420</td></tr><tr><td>2007</td><td>-5.5378846</td></tr></tbody></table>	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
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																											
Lesotho	0.08	<table><thead><tr><th>year</th><th>.resid</th></tr></thead><tbody><tr><td>1952</td><td>-5.2410256</td></tr><tr><td>1957</td><td>-2.8098543</td></tr><tr><td>1962</td><td>-0.5876830</td></tr><tr><td>1967</td><td>-0.3205117</td></tr><tr><td>1972</td><td>0.4766597</td></tr><tr><td>1977</td><td>2.4398310</td></tr><tr><td>1982</td><td>4.8320023</td></tr><tr><td>1987</td><td>6.4561737</td></tr><tr><td>1992</td><td>8.4833450</td></tr><tr><td>1997</td><td>3.8785163</td></tr><tr><td>2002</td><td>-7.5643124</td></tr><tr><td>2007</td><td>-10.0431410</td></tr></tbody></table>	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
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																											
		<table><thead><tr><th>year</th><th>.resid</th></tr></thead><tbody><tr><td>1952</td><td>-2.7419487</td></tr><tr><td>1957</td><td>-1.0127914</td></tr></tbody></table>	year	.resid	1952	-2.7419487	1957	-1.0127914																				
year	.resid																											
1952	-2.7419487																											
1957	-1.0127914																											



# master

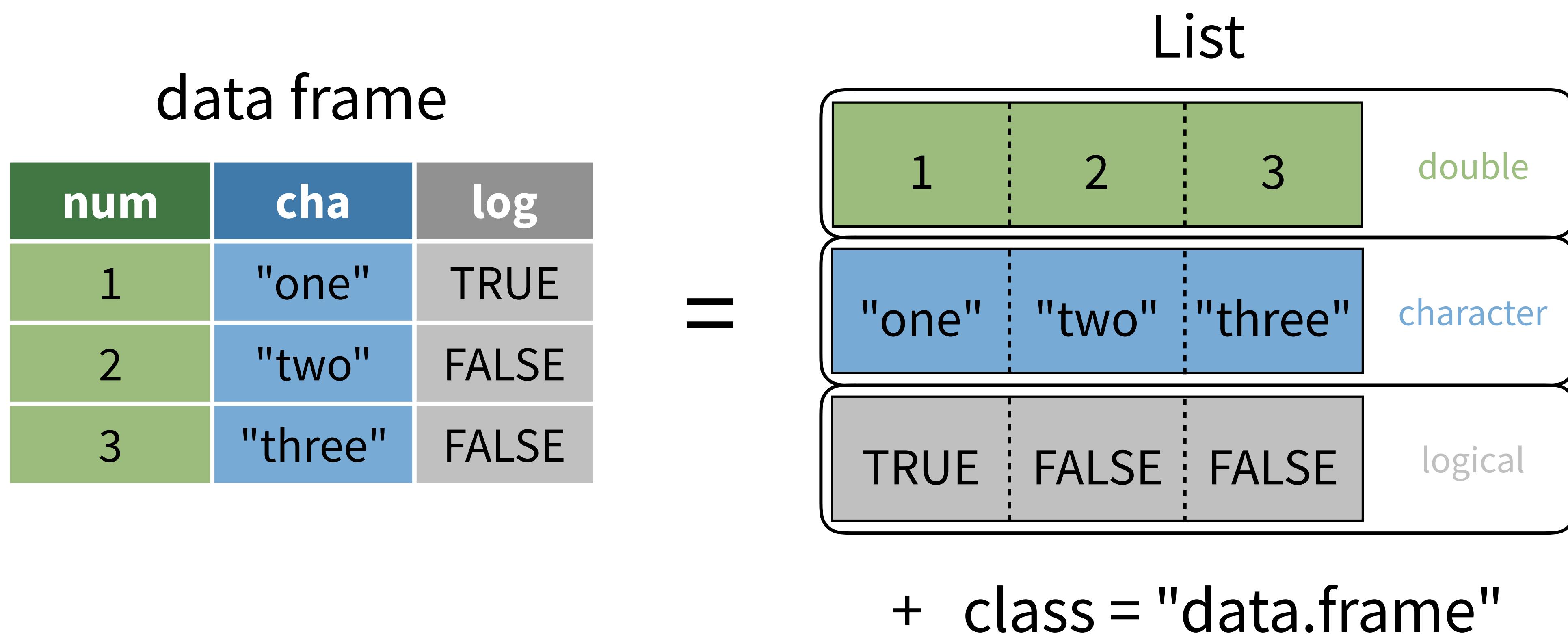
country	r.squared	data	model																														
Botswana	0.03	<table border="1"> <thead> <tr> <th>year</th><th>.resid</th></tr> </thead> <tbody> <tr><td>1952</td><td>-5.3071154</td></tr> <tr><td>1957</td><td>-3.6144580</td></tr> <tr><td>1962</td><td>-2.0158007</td></tr> <tr><td>1967</td><td>-0.5411434</td></tr> <tr><td>1972</td><td>1.8815140</td></tr> <tr><td>1977</td><td>4.8731713</td></tr> <tr><td>1982</td><td>6.7348287</td></tr> <tr><td>1987</td><td>8.5694860</td></tr> <tr><td>1992</td><td>7.3891434</td></tr> <tr><td>1997</td><td>-3.1031993</td></tr> <tr><td>2002</td><td>-9.3285420</td></tr> <tr><td>2007</td><td>-5.5378846</td></tr> </tbody> </table>	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	<p>Call:  <code>lm(formula = lifeExp ~ year, data = .)</code></p> <p>Coefficients:</p> <table> <tr> <td>(Intercept)</td> <td>year</td> </tr> <tr> <td><b>-65.49586</b></td> <td><b>0.06067</b></td> </tr> </table>	(Intercept)	year	<b>-65.49586</b>	<b>0.06067</b>
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																																
(Intercept)	year																																
<b>-65.49586</b>	<b>0.06067</b>																																
Lesotho	0.08	<table border="1"> <thead> <tr> <th>year</th><th>.resid</th></tr> </thead> <tbody> <tr><td>1952</td><td>-5.2410256</td></tr> <tr><td>1957</td><td>-2.8098543</td></tr> <tr><td>1962</td><td>-0.5876830</td></tr> <tr><td>1967</td><td>-0.3205117</td></tr> <tr><td>1972</td><td>0.4766597</td></tr> <tr><td>1977</td><td>2.4398310</td></tr> <tr><td>1982</td><td>4.8320023</td></tr> <tr><td>1987</td><td>6.4561737</td></tr> <tr><td>1992</td><td>8.4833450</td></tr> <tr><td>1997</td><td>3.8785163</td></tr> <tr><td>2002</td><td>-7.5643124</td></tr> <tr><td>2007</td><td>-10.0431410</td></tr> </tbody> </table>	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	<p>Call:  <code>lm(formula = lifeExp ~ year, data = .)</code></p> <p>Coefficients:</p> <table> <tr> <td>(Intercept)</td> <td>year</td> </tr> <tr> <td><b>-139.16529</b></td> <td><b>0.09557</b></td> </tr> </table>	(Intercept)	year	<b>-139.16529</b>	<b>0.09557</b>
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																																
(Intercept)	year																																
<b>-139.16529</b>	<b>0.09557</b>																																
		<table border="1"> <thead> <tr> <th>year</th><th>.resid</th></tr> </thead> <tbody> <tr><td>1952</td><td>-2.7419487</td></tr> <tr><td>1957</td><td>-1.0127914</td></tr> </tbody> </table>	year	.resid	1952	-2.7419487	1957	-1.0127914																									
year	.resid																																
1952	-2.7419487																																
1957	-1.0127914																																



# Quiz

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

# A data frame/tibble is a list!



# 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)

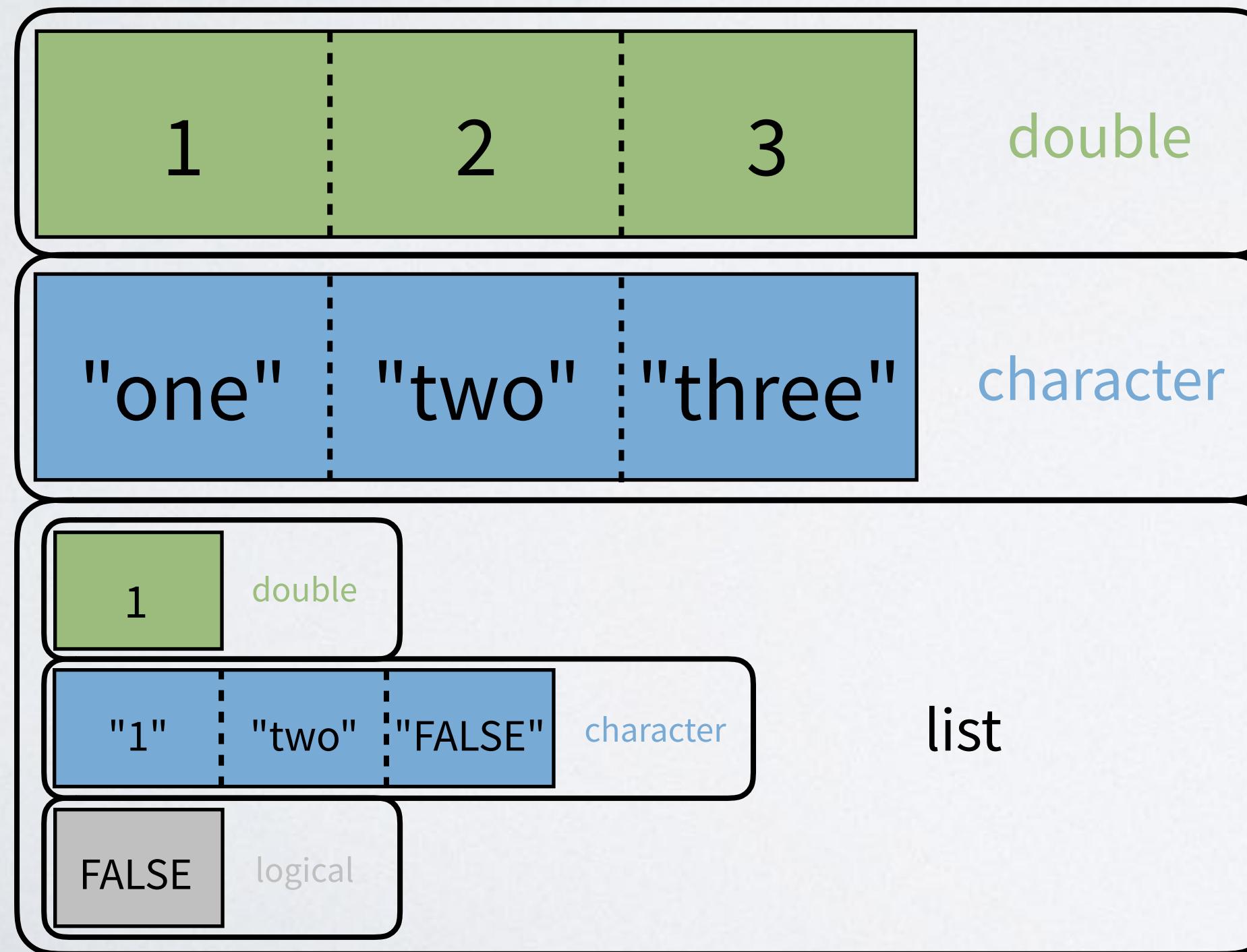
num
1
2
3



# 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)  
)
```

<b>num</b> <code>&lt;dbl&gt;</code>	<b>cha</b> <code>&lt;chr&gt;</code>	<b>listcol</b> <code>&lt;list&gt;</code>
1	one	<code>&lt;dbl [1]&gt;</code>
2	two	<code>&lt;chr [3]&gt;</code>
3	three	<code>&lt;lgl [1]&gt;</code>

3 rows



# nesting



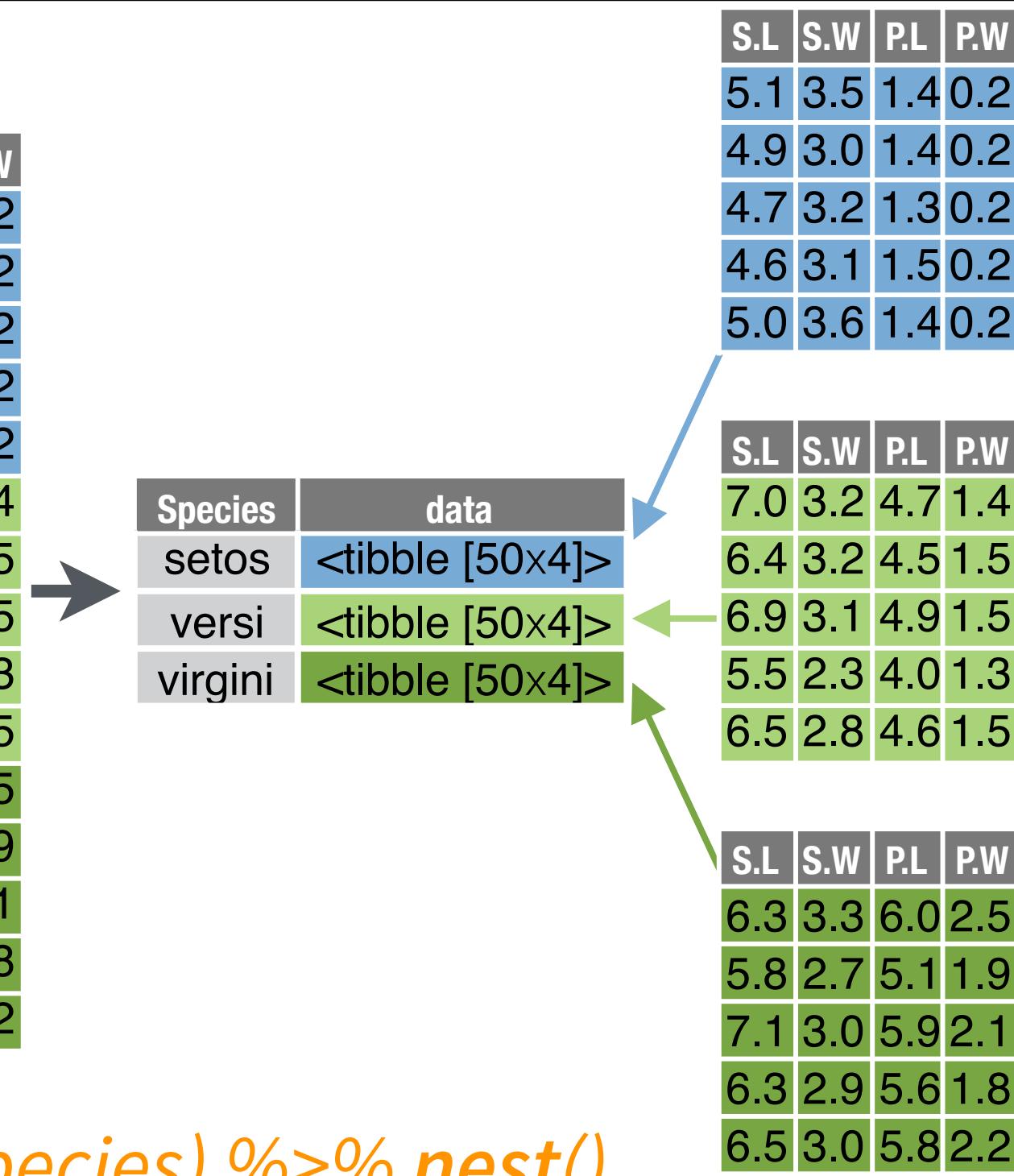
# nest()

Places grouped cases into a list column.

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

Species	S.L	S.W	P.L	P.W
setosa	5.1	3.5	1.4	0.2
setosa	4.9	3.0	1.4	0.2
setosa	4.7	3.2	1.3	0.2
setosa	4.6	3.1	1.5	0.2
setosa	5.0	3.6	1.4	0.2
versi	7.0	3.2	4.7	1.4
versi	6.4	3.2	4.5	1.5
versi	6.9	3.1	4.9	1.5
versi	5.5	2.3	4.0	1.3
versi	6.5	2.8	4.6	1.5
virgini	6.3	3.3	6.0	2.5
virgini	5.8	2.7	5.1	1.9
virgini	7.1	3.0	5.9	2.1
virgini	6.3	2.9	5.6	1.8
virgini	6.5	3.0	5.8	2.2

Species	S.L	S.W	P.L	P.W
setosa	5.1	3.5	1.4	0.2
setosa	4.9	3.0	1.4	0.2
setosa	4.7	3.2	1.3	0.2
setosa	4.6	3.1	1.5	0.2
setosa	5.0	3.6	1.4	0.2
versi	7.0	3.2	4.7	1.4
versi	6.4	3.2	4.5	1.5
versi	6.9	3.1	4.9	1.5
versi	5.5	2.3	4.0	1.3
versi	6.5	2.8	4.6	1.5
virgini	6.3	3.3	6.0	2.5
virgini	5.8	2.7	5.1	1.9
virgini	7.1	3.0	5.9	2.1
virgini	6.3	2.9	5.6	1.8
virgini	6.5	3.0	5.8	2.2



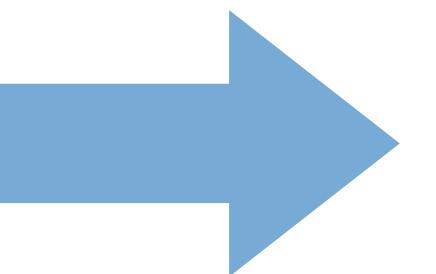
`n_iris <- iris %>% group_by(Species) %>% nest()`



# nest()

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

country	year	.resid
Botswana	1952	-5.3071154
Botswana	1957	-3.6144580
Botswana	1962	-2.0158007
Botswana	1967	-0.5411434
Botswana	1972	1.8815140
Botswana	1977	4.8731713
Botswana	1982	6.7348287
Botswana	1987	8.5694860
Botswana	1992	7.3891434
Botswana	1997	-3.1031993
Botswana	2002	-9.3285420
Botswana	2007	-5.5378846
Lesotho	1952	-5.2410256
Lesotho	1957	-2.8098543
Lesotho	1962	-0.5876830
Lesotho	1967	-0.3205117
Lesotho	1972	0.4766597
Lesotho	1977	2.4398310
Lesotho	1982	4.8320023
Lesotho	1987	6.4561737



country	data	
	year	.resid
Botswana	1952	-5.3071154
Botswana	1957	-3.6144580
Botswana	1962	-2.0158007
Botswana	1967	-0.5411434
Botswana	1972	1.8815140
Botswana	1977	4.8731713
Botswana	1982	6.7348287
Botswana	1987	8.5694860
Botswana	1992	7.3891434
Botswana	1997	-3.1031993
Botswana	2002	-9.3285420
Botswana	2007	-5.5378846
country	data	
	year	.resid
Lesotho	1952	-5.2410256
Lesotho	1957	-2.8098543
Lesotho	1962	-0.5876830
Lesotho	1967	-0.3205117
Lesotho	1972	0.4766597
Lesotho	1977	2.4398310
Lesotho	1982	4.8320023
Lesotho	1987	6.4561737



# 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

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

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

```
master$data[[1]]
```

**country**  
<fctr>

Afghanistan

Albania

Algeria

Angola

Argentina

Australia

Austria

Bahrain

Bangladesh

Belgium

**data**  
<list>

<tibble>

continent	year	lifeExp	pop	gdpPercap
<fctr>	<int>	<dbl>	<int>	<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

1-10 of 142 rows

Previous 1 2 3 4 5 6 ... 15 Next

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

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

**country**  
<fctr>

Afghanistan

Albania

Algeria

Angola

Argentina

Australia

Austria

Bahrain

Bangladesh

Belgium

**map()**  
**takes a list**

**data**  
<list>

<tibble> <S3: lm>

**...and  
returns a list**

```
master$model[[1]]
```

country	data	model
<fctr>	<list>	<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
```

```
master <- master %>%  
  mutate(r.squared = map dbl(model, get_rsq))
```

country	data	model	r.squared
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 4

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



```
fit_model <- function(df) lm(lifeExp ~ year, data = df)
get_rsq <- function(mod) glance(mod)$r.squared

master <- gapminder %>%
  group_by(country) %>%
  nest() %>%
  mutate(model = map(data, fit_model),
         r.squared = map_dbl(model, get_rsq),
         output = map(model, augment))
master
```



country	data	model	r.squared	output
<fctr>	<list>	<list>	<dbl>	<list>
Afghanistan	<tibble>	<S3: lm>	0.94771226	<data.frame [12 × 9]>
Albania	<tibble>	<S3: lm>	0.91057777	<data.frame [12 × 9]>
Algeria	<tibble>	<S3: lm>	0.98511721	<data.frame [12 × 9]>
Angola	<tibble>	<S3: lm>	0.88781463	<data.frame [12 × 9]>
Argentina	<tibble>	<S3: lm>	0.99556810	<data.frame [12 × 9]>
Australia	<tibble>	<S3: lm>	0.97964774	<data.frame [12 × 9]>
Austria	<tibble>	<S3: lm>	0.99213401	<data.frame [12 × 9]>
Bahrain	<tibble>	<S3: lm>	0.96673981	<data.frame [12 × 9]>
Bangladesh	<tibble>	<S3: lm>	0.98936087	<data.frame [12 × 9]>
Belgium	<tibble>	<S3: lm>	0.99454056	<data.frame [12 × 9]>

1-10 of 142 rows

Previous

1

2

3

4

5

6

...

15

Next

# master\$output[[1]]

country	data	model	r.squared	output				
	<list>	<list>	<dbl>	<list>				
Afghanistan	<tibble>	<S3: lm>	0.94771226	<data.frame [12 × 9]>				
Albania	<tibble>	<S3: lm>	0.91057777	<data.frame [12 × 9]>				
Algeria				12 × 9]>				
Angola	lifeExp <dbl>	year <int>	.fitted <dbl>	.se.fit <dbl>	.resid <dbl>	.hat <dbl>	.sigma <dbl>	12 × 9]>
Argentina	28.801	1952	29.90729	0.6639995	-1.10629487	0.29487179	1.211813	12 × 9]>
Australia	30.332	1957	31.28394	0.5799442	-0.95193823	0.22494172	1.237512	12 × 9]>
Austria	31.997	1962	32.66058	0.5026799	-0.66358159	0.16899767	1.265886	12 × 9]>
Bahrain	34.020	1967	34.03722	0.4358337	-0.01722494	0.12703963	1.288917	12 × 9]>
Bangladesh	36.088	1972	35.41387	0.3848726	0.67413170	0.09906760	1.267003	12 × 9]>
Belgium	38.438	1977	36.79051	0.3566719	1.64748834	0.08508159	1.154002	12 × 9]>
Belgium	39.854	1982	38.16716	0.3566719	1.68684499	0.08508159	1.147076	12 × 9]>
Belgium	40.822	1987	39.54380	0.3848726	1.27820163	0.09906760	1.208243	12 × 9]>
Belgium	41.674	1992	40.92044	0.4358337	0.75355828	0.12703963	1.260583	12 × 9]>
Belgium	41.763	1997	42.29709	0.5026799	-0.53408508	0.16899767	1.274051	12 × 9]>
1-10 of 14					15	Next		
1-10 of 12 rows   1-7 of 9 columns					Previous	1	2	Next

# Benefits

Data and models stay in correspondence across manipulations

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

country	data	model	r.squared	output
<fctr>	<list>	<list>	<dbl>	<list>
Pakistan	<tibble>	<S3: lm>	0.9972497	<data.frame [12 × 9]>
Panama	<tibble>	<S3: lm>	0.9511952	<data.frame [12 × 9]>
Paraguay	<tibble>	<S3: lm>	0.9829865	<data.frame [12 × 9]>
Peru	<tibble>	<S3: lm>	0.9884740	<data.frame [12 × 9]>
Philippines	<tibble>	<S3: lm>	0.9914226	<data.frame [12 × 9]>
Poland	<tibble>	<S3: lm>	0.8396631	<data.frame [12 × 9]>
Portugal	<tibble>	<S3: lm>	0.9690351	<data.frame [12 × 9]>
Puerto Rico	<tibble>	<S3: lm>	0.9078191	<data.frame [12 × 9]>

8 rows

But how can we plot?

# unnest()

Unnests one or more list columns

```
master %>%  
  unnest(data, output, .drop = FALSE)
```

A nested data frame

list columns to unnest (should contain data frames)

Drop remaining list columns from the result?



# unnest()

Unnests one or more list columns

```
master %>%  
  unnest(data, output, .drop = FALSE)
```

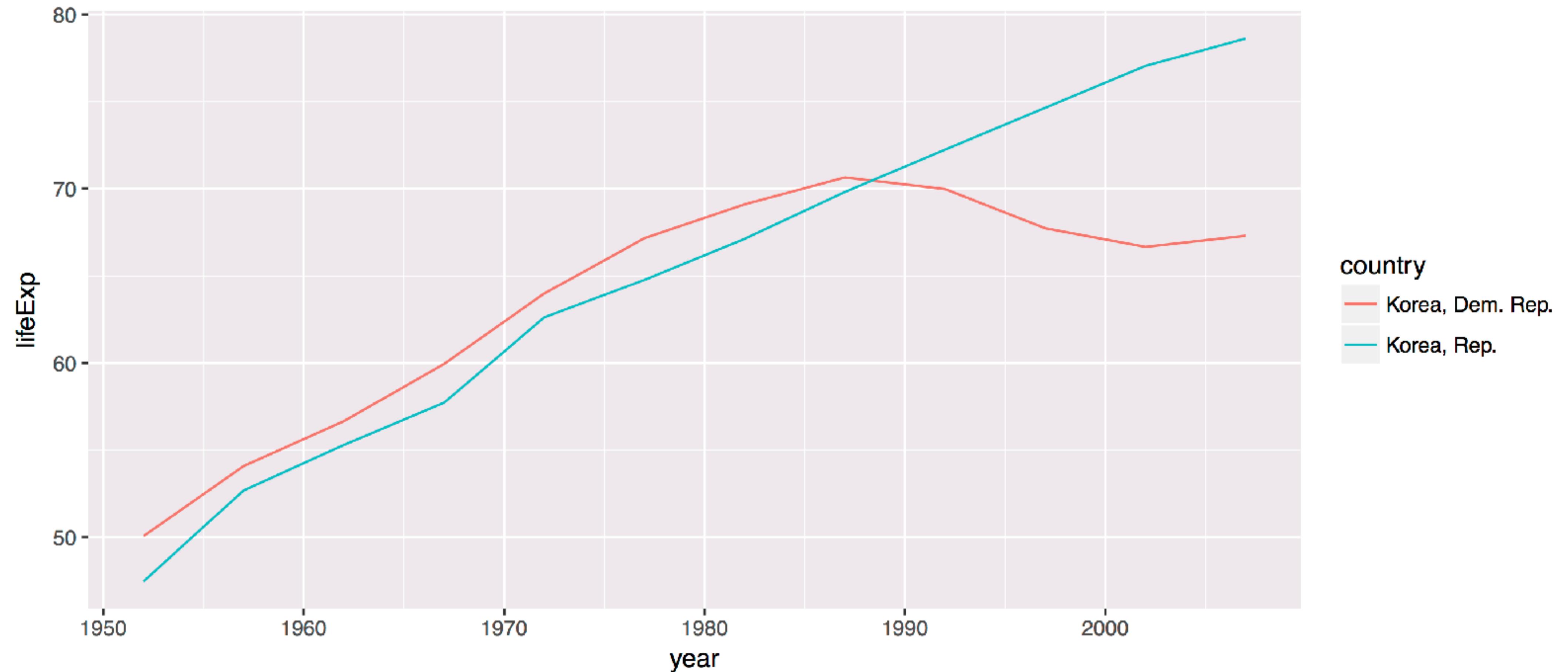
country <fctr>	r.squared <dbl>	continent <fctr>	year <int>	lifeExp <dbl>	pop <int>	gdpPerCap <dbl>	▶
Afghanistan	0.94771226	Asia	1952	28.80100	8425333	779.4453	
Afghanistan	0.94771226	Asia	1957	30.33200	9240934	820.8530	
Afghanistan	0.94771226	Asia	1962	31.99700	10267083	853.1007	
Afghanistan	0.94771226	Asia	1967	34.02000	11537966	836.1971	
Afghanistan	0.94771226	Asia	1972	36.08800	13079460	739.9811	
Afghanistan	0.94771226	Asia	1977	38.43800	14880372	786.1134	
Afghanistan	0.94771226	Asia	1982	39.85400	12881816	978.0114	
Afghanistan	0.94771226	Asia	1987	40.82200	13867957	852.3959	
Afghanistan	0.94771226	Asia	1992	41.67400	16317921	649.3414	
Afghanistan	0.94771226	Asia	1997	41.76300	22227415	635.3414	

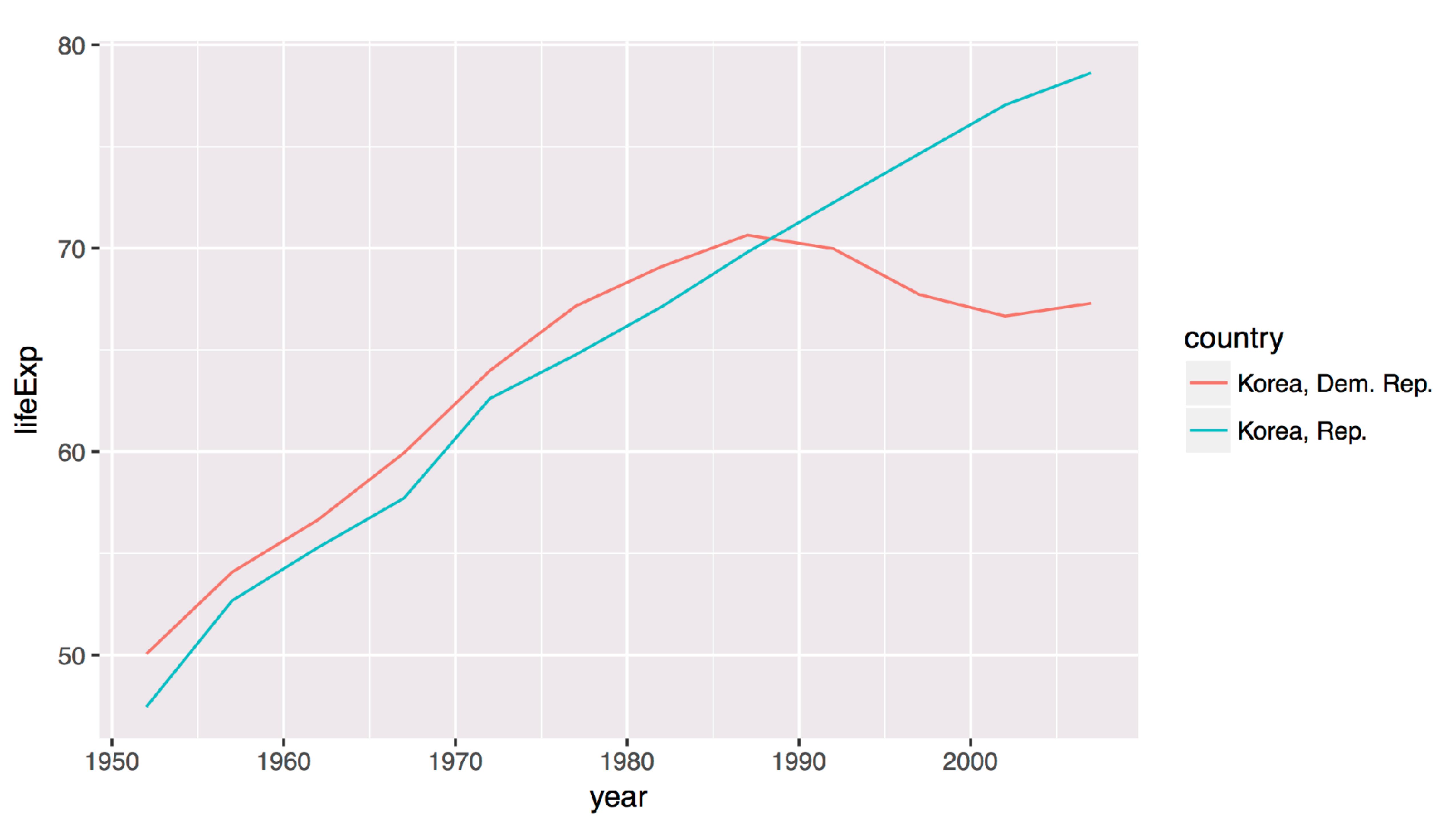


```
master %>%  
  filter(str_detect(country, pattern = "Korea")) %>%  
  unnest(data)
```

country <fctr>	r.squared <dbl>	continent <fctr>	year <int>	lifeExp <dbl>	pop <int>	gdpPerCap <dbl>
Korea, Dem. Rep.	0.7030631	Asia	1952	50.056	8865488	1088.278
Korea, Dem. Rep.	0.7030631	Asia	1957	54.081	9411381	1571.135
Korea, Dem. Rep.	0.7030631	Asia	1962	56.656	10917494	1621.694
Korea, Dem. Rep.	0.7030631	Asia	1967	59.942	12617009	2143.541
Korea, Dem. Rep.	0.7030631	Asia	1972	63.983	14781241	3701.622
Korea, Dem. Rep.	0.7030631	Asia	1977	67.159	16325320	4106.301
Korea, Dem. Rep.	0.7030631	Asia	1982	69.100	17647518	4106.525
Korea, Dem. Rep.	0.7030631	Asia	1987	70.647	19067554	4106.492
Korea, Dem. Rep.	0.7030631	Asia	1992	69.978	20711375	3726.064
Korea, Dem. Rep.	0.7030631	Asia	1997	67.727	21585105	1690.757

```
master %>%
  filter(str_detect(country, pattern = "Korea")) %>%
  unnest(data) %>%
  ggplot() +
  geom_line(aes(year, lifeExp, color = country))
```



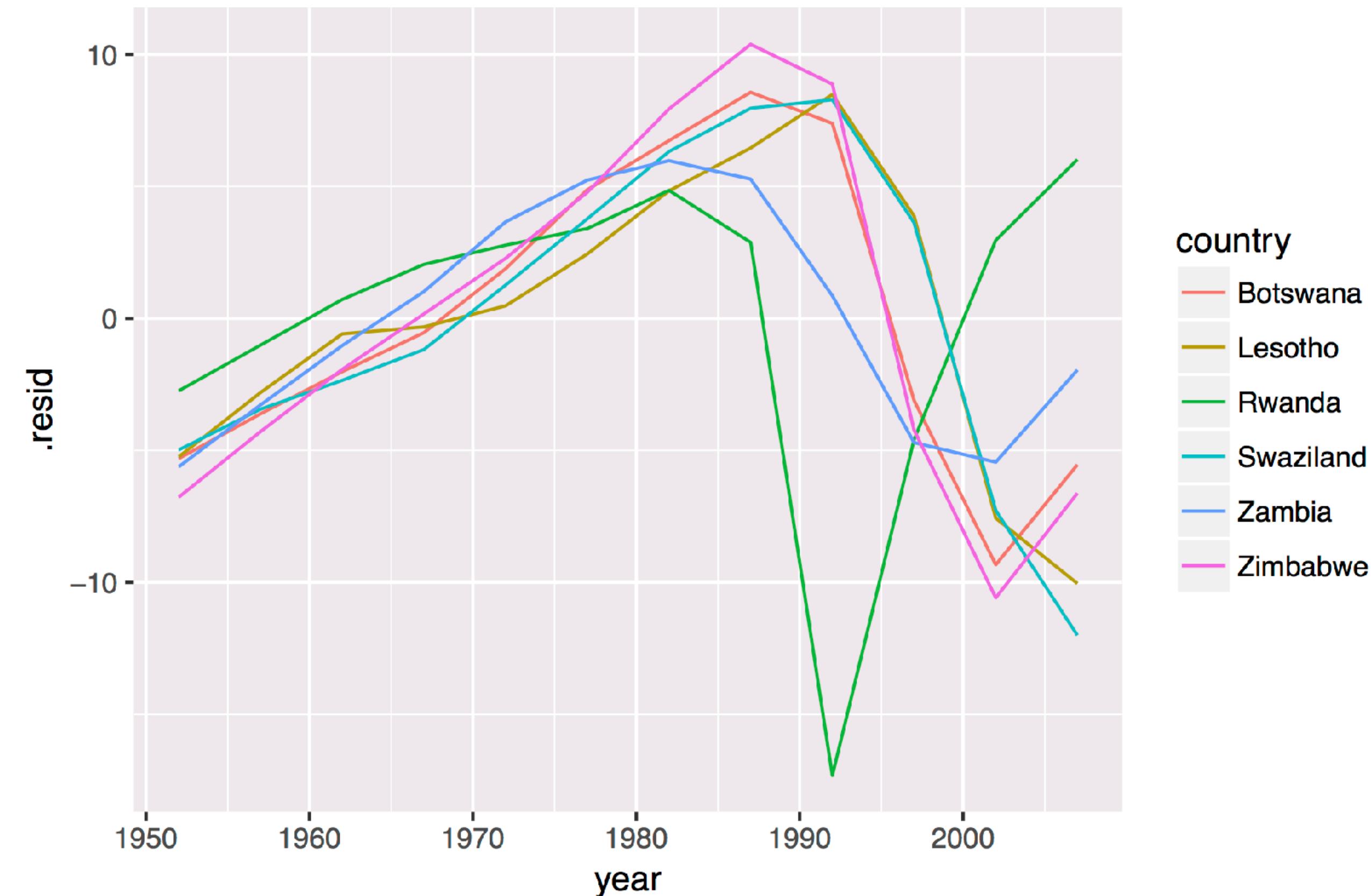


# Your Turn 5

Use master to recreate our plot of the **residuals vs year** for the six countries with an r squared less than 0.25.



```
master %>%
  filter(r.squared < 0.25) %>%
  unnest(output) %>%
  ggplot() +
  geom_line(mapping = aes(x = year, y = .resid, color = country))
```



# Take Away

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

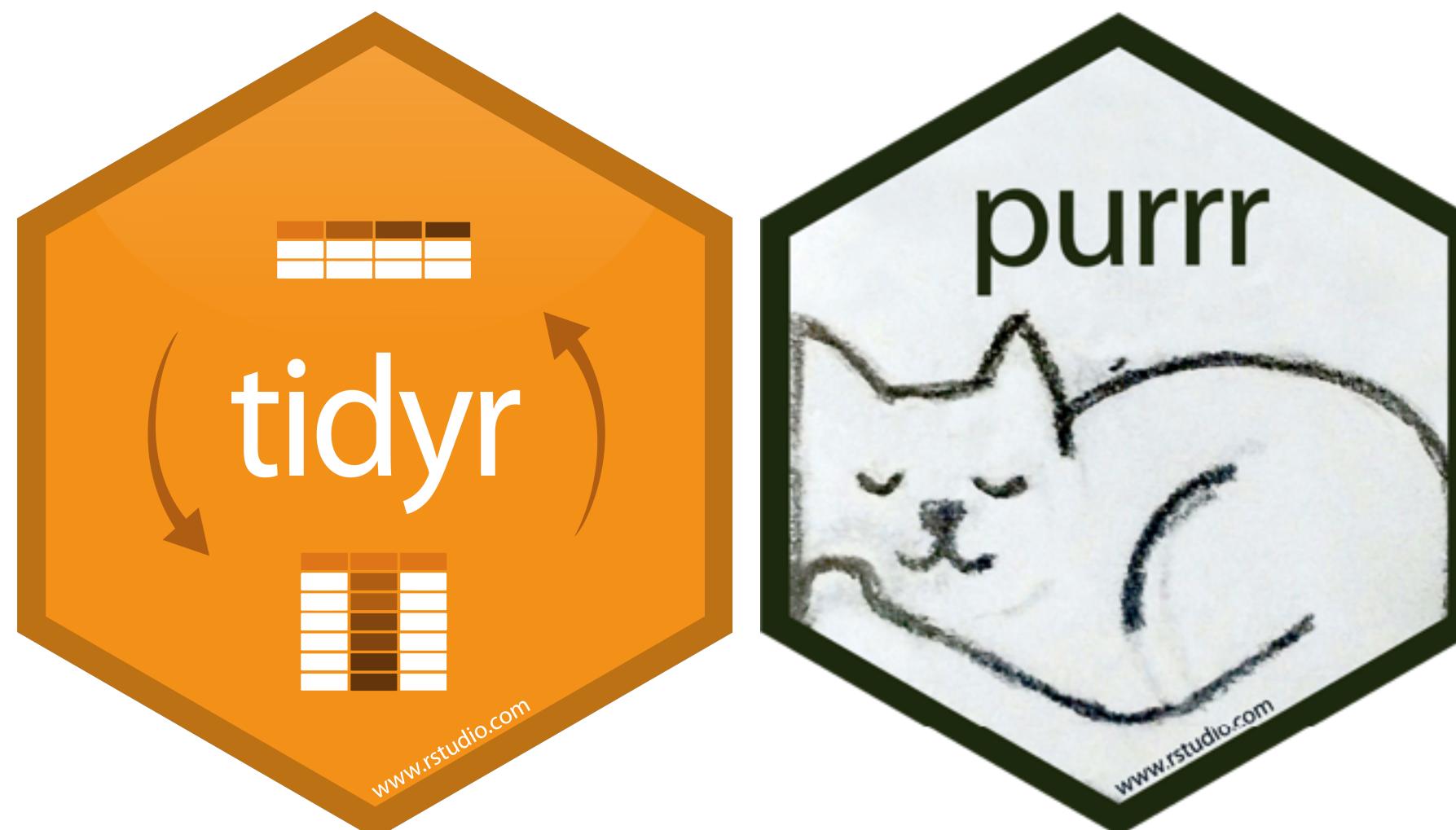
country	r.squared	data	model																										
Botswana	0.03	<table><thead><tr><th>year</th><th>.resid</th></tr></thead><tbody><tr><td>1952</td><td>-5.3071154</td></tr><tr><td>1957</td><td>-3.6144580</td></tr><tr><td>1962</td><td>-2.0158007</td></tr><tr><td>1967</td><td>-0.5411434</td></tr><tr>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



"Better experimental design = simpler statistics.  
Better data model = simpler analysis."

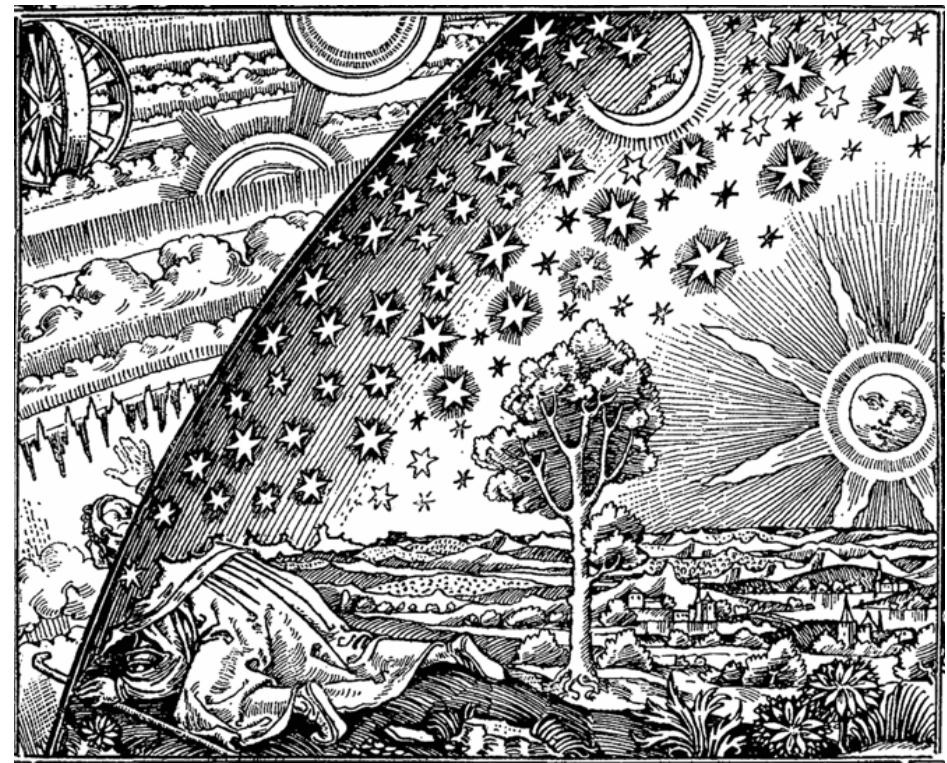
- Jenny Bryan (2016)

# List Columns

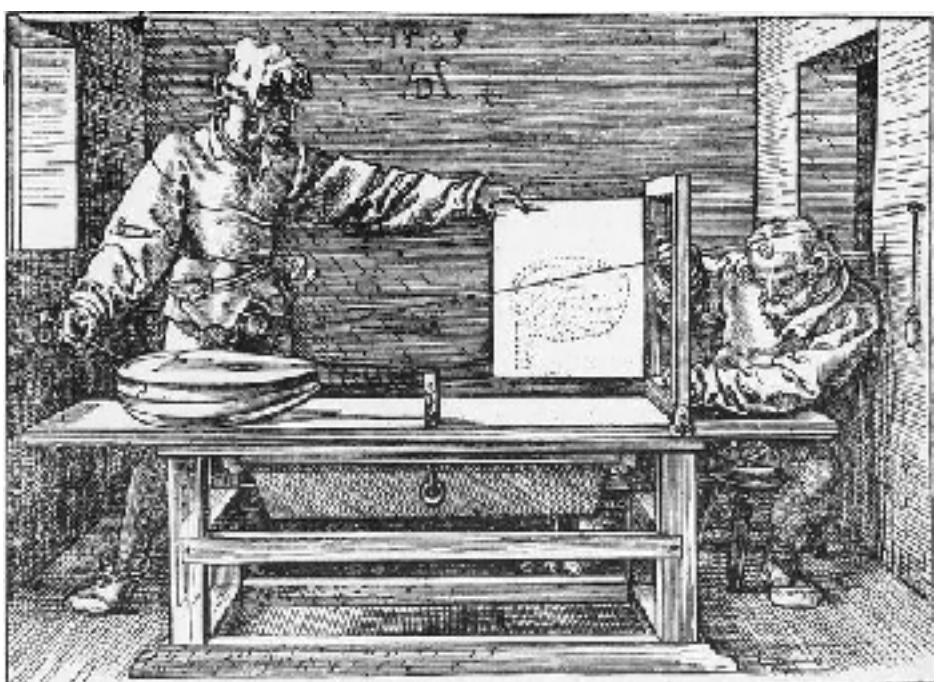


# Thank You





Solutions at  
[github.com/rstudio/master-the-tidyverse-solutions](https://github.com/rstudio/master-the-tidyverse-solutions)



Please take the class survey  
[www.surveymonkey.com/r/SX9X69R](https://www.surveymonkey.com/r/SX9X69R)