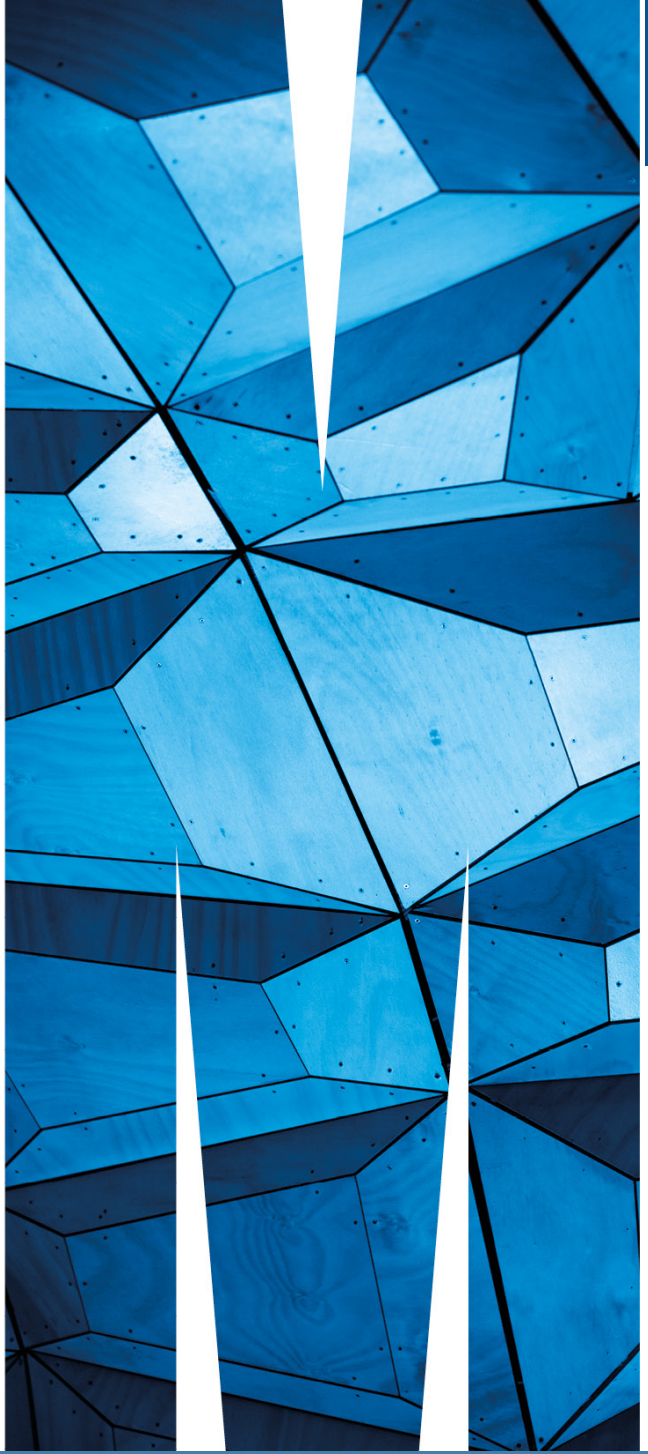
MONASH University

MONASH BUSINESS SCHOOL



Introduction to Multi-level Models using R

Professor Di Cook, Econometrics and Business Statistics

Workshop for the Institute for Safety, Compensation and Recovery Research

GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED EFMD EQUIS ACCREDITED ASSOCIATION AMBA ACCREDITED

Outline

- Session 1: Basic models, fitting multiple separate models
- Session 2: Putting it together, using mixed effects models
- Session 3: Summarising and visualising models
- Session 4: Advanced modeling

Session 1

- **Basic models, fitting multiple separate models**

Your turn

- What is a model?

Gapminder data

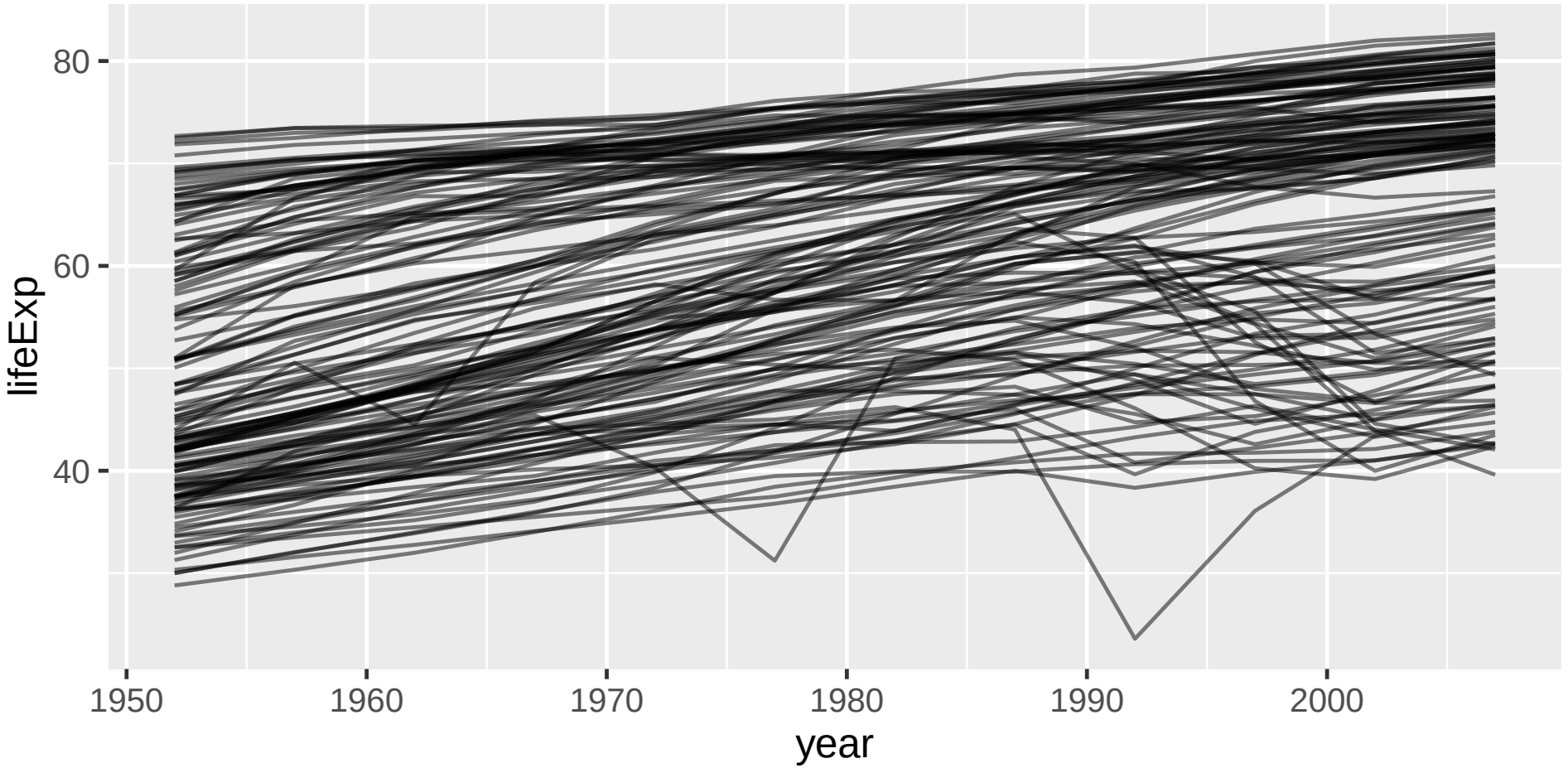
- Original source from Hans Rosling's software and TED talk
- Example modeling code by Hadley Wickham, enhanced by Heike Hofmann
- Demographic data by country and continent since 1952, life expectancy and GDP per capita (reduced subset in the R package from the original)

Gapminder

```
library(gapminder)
glimpse(gapminder)
# Observations: 1,704
# Variables: 6
# $ country   (fctr) Afghanistan, Afghanistan, Afghanistan, Af
# $ continent (fctr) Asia, Asia, Asia, Asia, Asia, Asia, Asia,
# $ year      (int) 1952, 1957, 1962, 1967, 1972, 1977, 1982,
# $ lifeExp   (dbl) 29, 30, 32, 34, 36, 38, 40, 41, 42, 42, 42
# $ pop       (int) 8425333, 9240934, 10267083, 11537966, 130
# $ gdpPercap (dbl) 779, 821, 853, 836, 740, 786, 978, 852, 62
```

Take a look

```
ggplot(data=gapminder, aes(x=year, y=lifeExp, group=country))
  geom_line(alpha=0.5)
```



How would you describe this plot?

Using models as exploratory tools

- Idea: fit a line to each one of the countries' life expectancies
- then use e.g. intercept and slope to characterise groups of countries

```
gapminder2 <- gapminder %>% mutate(year1950 = year-1950)
by_country <- gapminder2 %>%
  select(country, year1950, lifeExp, continent) %>%
  group_by(country, continent) %>%
  nest()
```

From a data frame . . . .

country	continent	year1950	lifeExp
Afghanistan	Asia	2	29
Afghanistan	Asia	7	30
Afghanistan	Asia	12	32
Afghanistan	Asia	17	34
Afghanistan	Asia	22	36
Afghanistan	Asia	27	38
Afghanistan	Asia	32	40
Afghanistan	Asia	37	41
Afghanistan	Asia	42	42
Afghanistan	Asia	47	42
Afghanistan	Asia	52	42
Afghanistan	Asia	57	44
Albania	Europe	2	55
Albania	Europe	7	59
Albania	Europe	12	65

. . . to a list of data frames

1-model-basics.Rmd × | by_country ×

Filter

	country	continent	data
1	Afghanistan	Asia	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
2	Albania	Europe	list
3	Algeria	Africa	list
4	Angola	Africa	list
5	Argentina	Americas	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
6	Australia	Oceania	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
7	Austria	Europe	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
8	Bahrain	Asia	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
9	Bangladesh	Asia	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
10	Belgium	Europe	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
11	Benin	Africa	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
12	Bolivia	Americas	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
13	Bosnia and Herzegovina	Europe	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...
14	Botswana	Africa	list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 4...

list(year1950 = c(2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57), lifeExp = c(28.801, 30.332, 31.997, 34.02, 36.088, 38.438, 39.854, 40.822, 41.674, 41.763, 42.129, 43.828))

Figure 1:

purrr applies function to each element of the list

```
by_country <- by_country %>%
  mutate(
    model = purrr::map(data, ~ lm(lifeExp ~ year1950,
                                  data = .))
  )
```

Fits a linear model to each country, e.g.
`lm(lifeExp ~ year1950, data=australia)`

Use broom to unnest the fitted models

```
by_country <- by_country %>%
  unnest(model %>% purrr::map(broom::tidy))
kable(head(by_country))
```

country	continent	term	estimate	std.error	statistic	p.valu
Afghanistan	Asia	(Intercept)	29.36	0.70	42	
Afghanistan	Asia	year1950	0.28	0.02	13	
Albania	Europe	(Intercept)	58.56	1.13	52	
Albania	Europe	year1950	0.33	0.03	10	
Algeria	Africa	(Intercept)	42.24	0.76	56	
Algeria	Africa	year1950	0.57	0.02	26	

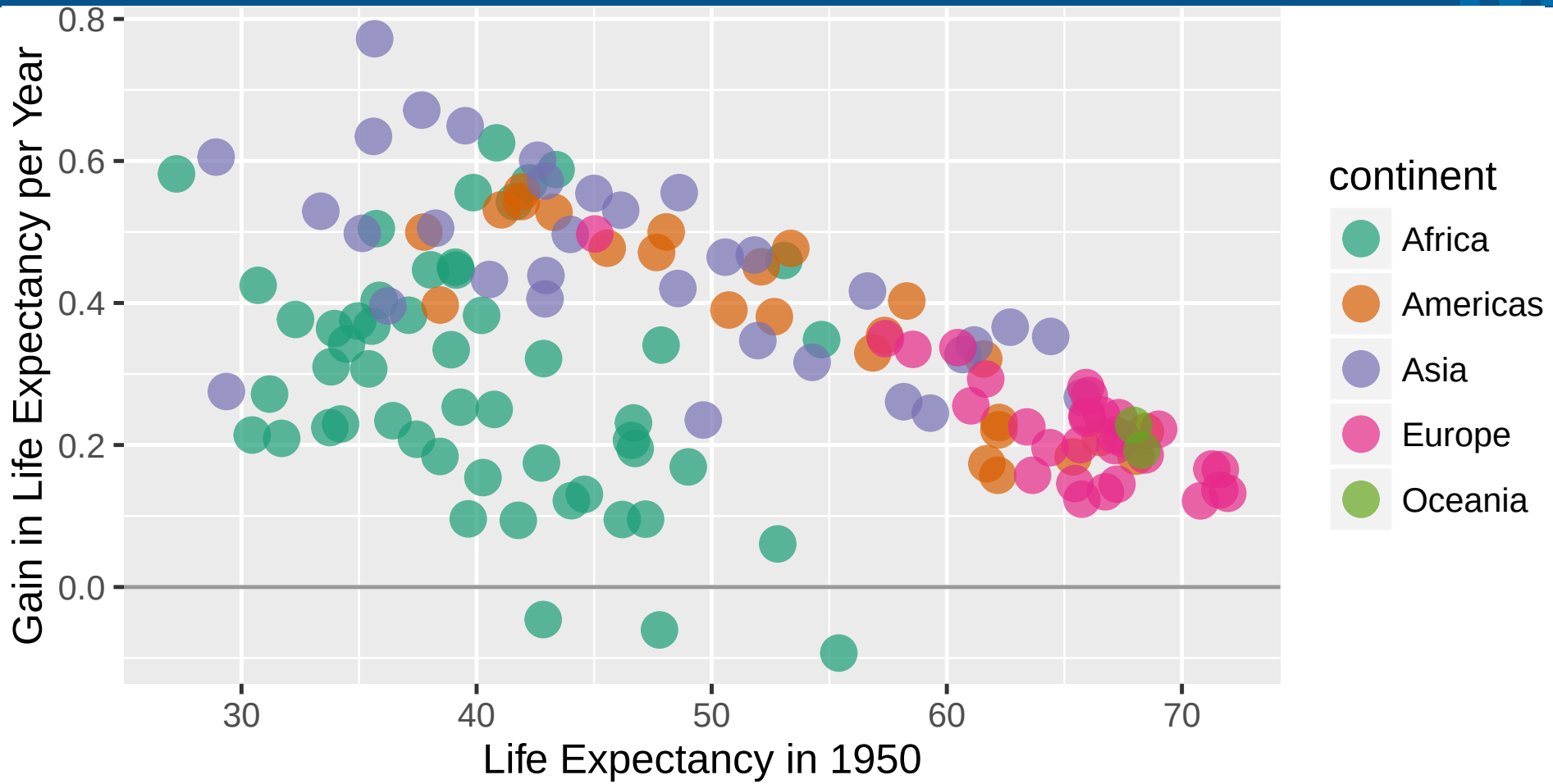
And tidyr::spread to keep desired items

```
country_coefs <- by_country %>%
  select(continent, country, term, estimate) %>%
  spread(term, estimate)
kable(head(country_coefs))
```

continent	country	(Intercept)	year1950
Africa	Algeria	42	0.57
Africa	Angola	32	0.21
Africa	Benin	39	0.33
Africa	Botswana	53	0.06
Africa	Burkina Faso	34	0.36
Africa	Burundi	40	0.15

Plot estimates

```
ggplot(data=country_coefs, aes(x=`(Intercept)`, y=year1950,
                               colour=continent)) +
  geom_hline(yintercept=0, colour="grey60") +
  geom_point(alpha=0.7, size=4) +
  scale_colour_brewer(palette="Dark2") +
  xlab("Life Expectancy in 1950") +
  ylab("Gain in Life Expectancy per Year")
```



What do we learn?

- High life expectancy in 1950 (e.g. Europe) tends to have smaller gains
- Most countries on the African continent observed lower respective gains, and three saw declines
- Largest gains occurred in Asia

Add interaction

```
ggplot(data=country_coefs, aes(x=`(Intercept)`, y=year1950,
                               colour=continent, label=country
  geom_hline(yintercept=0, colour="grey60") +
  geom_point(alpha=0.7, size=4) +
  scale_colour_brewer(palette="Dark2") +
  xlab("Life Expectancy in 1950") +
  ylab("Gain in Life Expectancy per Year")
ggplotly(tooltips="country")
```

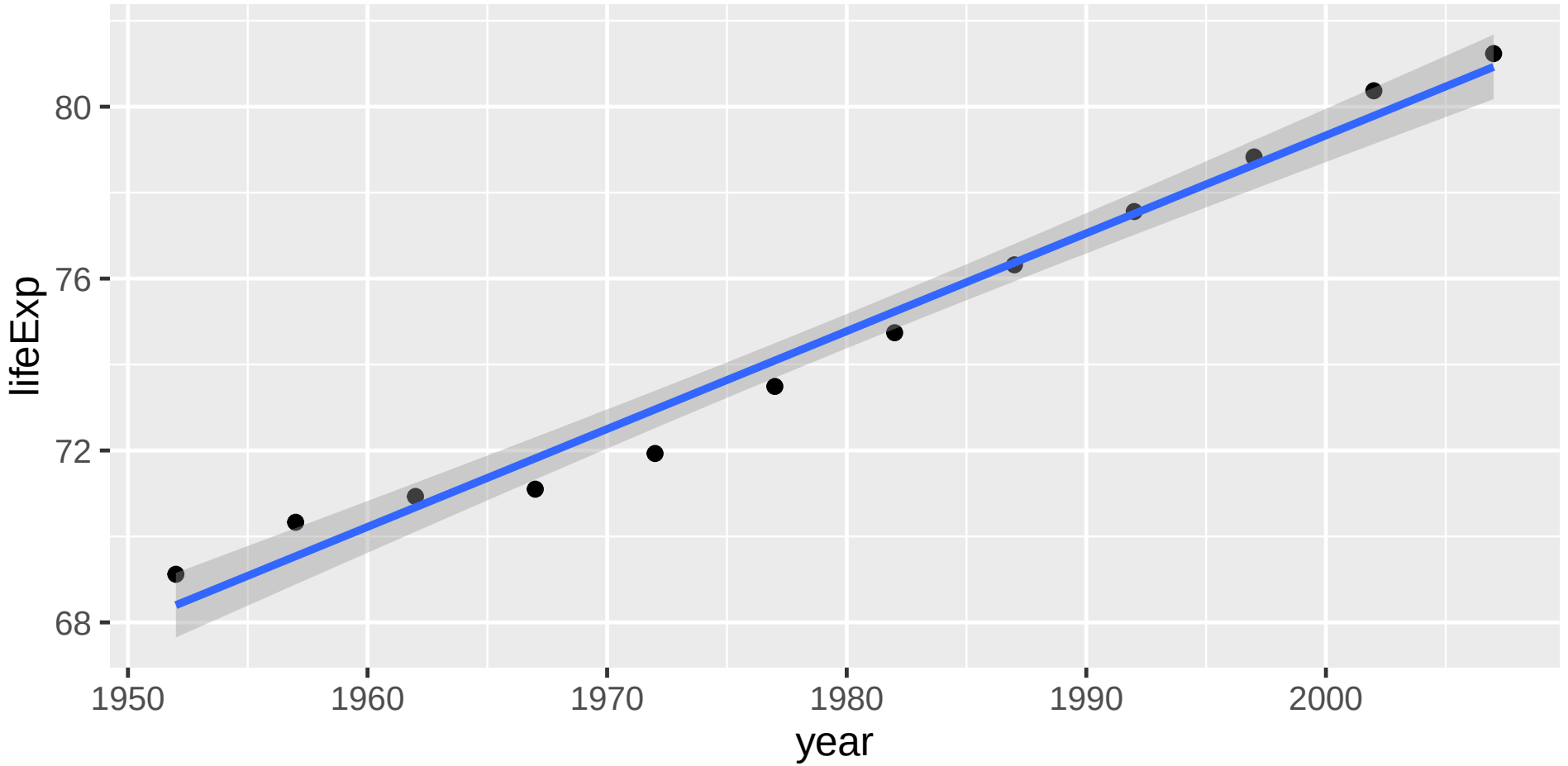
Australian Life Expectancy

```
oz <- gapminder %>% filter(country=="Australia")
kable(head(oz))
```

country	continent	year	lifeExp	pop	gdpPercap
Australia	Oceania	1952	69	8.7e+06	10040
Australia	Oceania	1957	70	9.7e+06	10950
Australia	Oceania	1962	71	1.1e+07	12217
Australia	Oceania	1967	71	1.2e+07	14526
Australia	Oceania	1972	72	1.3e+07	16789
Australia	Oceania	1977	73	1.4e+07	18334

Life Expectancy in Australia since 1950

```
ggplot(data=oz, aes(x=year, y=lifeExp)) +
  geom_point() +
  geom_smooth(method="lm")
```



```
oz.lm <- lm(lifeExp~year, data=oz)
oz.lm
#
# Call:
# lm(formula = lifeExp ~ year, data = oz)
#
# Coefficients:
# (Intercept)         year
#    -376.116        0.228
```

Intercept is estimated life expectancy at 0 BC - let's use 1950 for the first value:

```
oz <- oz %>% mutate(year = year-1950)
oz.lm <- lm(lifeExp~year, data=oz)
oz.lm
#
# Call:
# lm(formula = lifeExp ~ year, data = oz)
#
# Coefficients:
# (Intercept)         year
#      67.945        0.228
```

Nesting data

We don't want to subset the data for every country . . .
`nest()` makes a data frame part of another data frame:

```
by_country <- gapminder2 %>%
  group_by(continent, country) %>%
  nest()
head(by_country)
# Source: local data frame [6 x 3]
#
#   continent     country              data
#      (fctr)      (fctr)            (chr)
# 1      Asia Afghanistan <tbl_df [12,5]>
# 2    Europe     Albania <tbl_df [12,5]>
# 3    Africa     Algeria <tbl_df [12,5]>
# 4    Africa      Angola <tbl_df [12,5]>
# 5  Americas   Argentina <tbl_df [12,5]>
# 6   Oceania   Australia <tbl_df [12,5]>
```

Each element of the data variable in by_country is a dataset:

```
head(by_country$data[[1]])
# Source: local data frame [6 x 5]
#
#    year lifeExp      pop gdpPercap year1950
#   (int)   (dbl)    (int)     (dbl)    (dbl)
# 1  1952      29  8425333       779        2
# 2  1957      30  9240934       821        7
# 3  1962      32 10267083       853       12
# 4  1967      34 11537966       836       17
# 5  1972      36 13079460       740       22
# 6  1977      38 14880372       786       27
```

```
lm(lifeExp~year1950, data=by_country$data[[1]])
#
# Call:
# lm(formula = lifeExp ~ year1950, data = by_country$data[[1]]
#
# Coefficients:
# (Intercept)     year1950
#      29.357        0.275
```


Fitting multiple models

Now we are using the `map` function in the package `purrr`.
`map` allows us to apply a function to each element of a list.

```
by_country$model <- by_country$data %>%
  purrr::map(~lm(lifeExp~year1950, data=.))
head(by_country)
# Source: local data frame [6 x 4]
#
#   continent     country              data   model
#      (fctr)      (fctr)             (chr)   (chr)
# 1      Asia Afghanistan <tbl_df [12,5]> <S3:lm>
# 2    Europe     Albania <tbl_df [12,5]> <S3:lm>
# 3    Africa     Algeria <tbl_df [12,5]> <S3:lm>
# 4    Africa      Angola <tbl_df [12,5]> <S3:lm>
# 5  Americas   Argentina <tbl_df [12,5]> <S3:lm>
# 6   Oceania   Australia <tbl_df [12,5]> <S3:lm>
```

On to the broom package

broom allows to extract values from models on three levels:

- for each model: broom::glance
- for each coefficient in the model: broom::tidy
- for each value in the dataset: broom::augment

```
broom::glance(by_country$model[[1]])
#   r.squared adj.r.squared sigma statistic p.value df logLik
# 1      0.95          0.94   1.2       181 9.8e-08  2    -18
#   deviance df.residual
# 1       15          10
broom::tidy(by_country$model[[1]])
#          term estimate std.error statistic p.value
# 1 (Intercept)    29.36      0.70        42 1.4e-12
# 2    year1950     0.28      0.02        13 9.8e-08
```

```
broom::augment(by_country$model[[1]])
#    lifeExp year1950 .fitted .se.fit .resid  .hat .sigma .co
# 1       29        2      30    0.66 -1.106 0.295    1.2 2.4
# 2       30        7      31    0.58 -0.952 0.225    1.2 1.1
# 3       32       12      33    0.50 -0.664 0.169    1.3 3.6
# 4       34       17      34    0.44 -0.017 0.127    1.3 1.7
# 5       36       22      35    0.38  0.674 0.099    1.3 1.9
# 6       38       27      37    0.36  1.647 0.085    1.2 9.2
# 7       40       32      38    0.36  1.687 0.085    1.1 9.7
# 8       41       37      40    0.38  1.278 0.099    1.2 6.7
# 9       42       42      41    0.44  0.754 0.127    1.3 3.2
# 10      42       47      42    0.50 -0.534 0.169    1.3 2.3
# 11      42       52      44    0.58 -1.545 0.225    1.1 3.0
# 12      44       57      45    0.66 -1.222 0.295    1.2 3.0
```

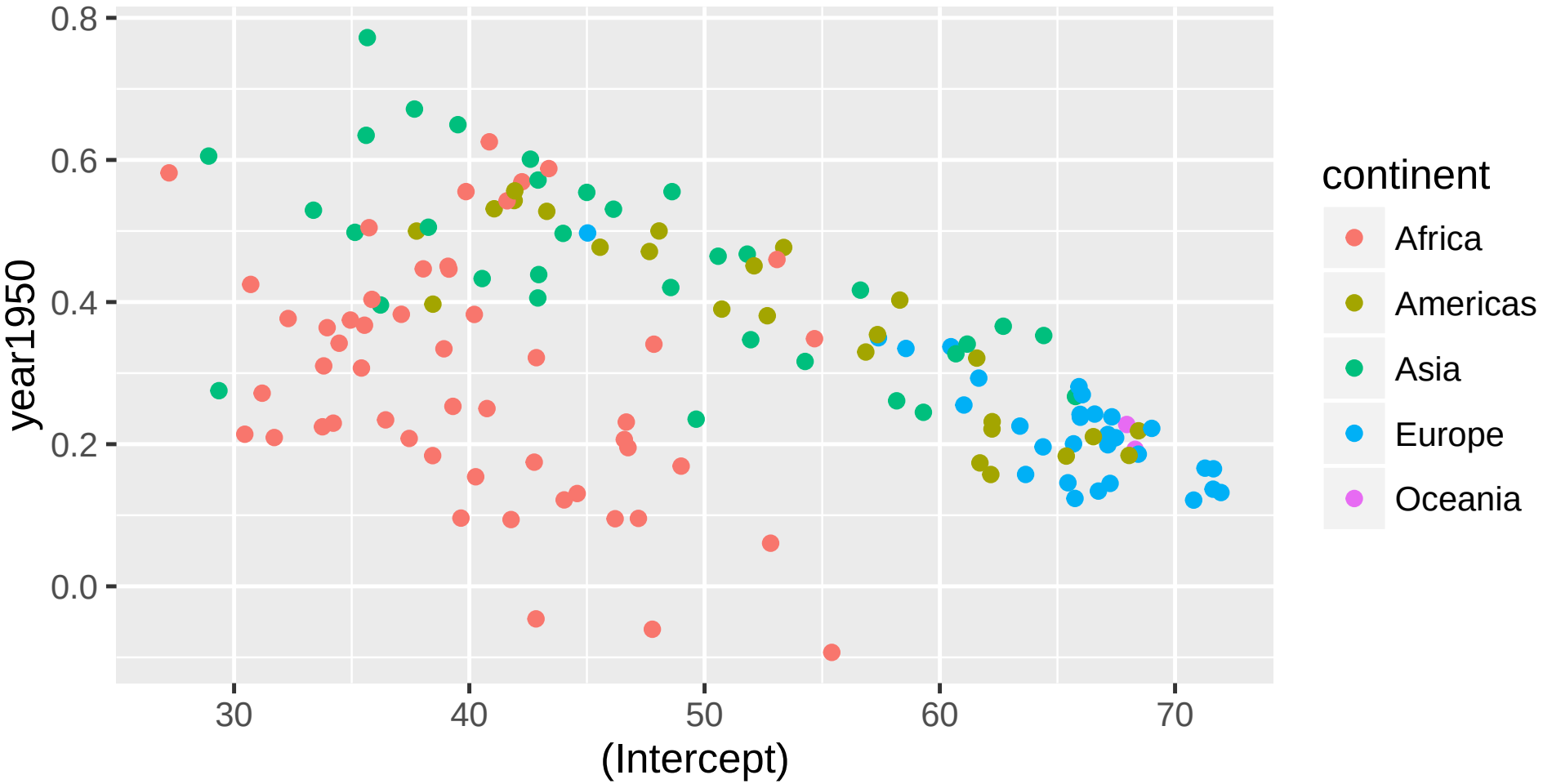
Extract values for each coefficient

Extract all countries automatically (hello `map` again!)

```
by_country_coefs <- by_country %>%
  unnest(model %>% purrr::map(broom::tidy))
coefs <- by_country_coefs %>%
  select(country, continent, term, estimate) %>%
  spread(term, estimate)
```

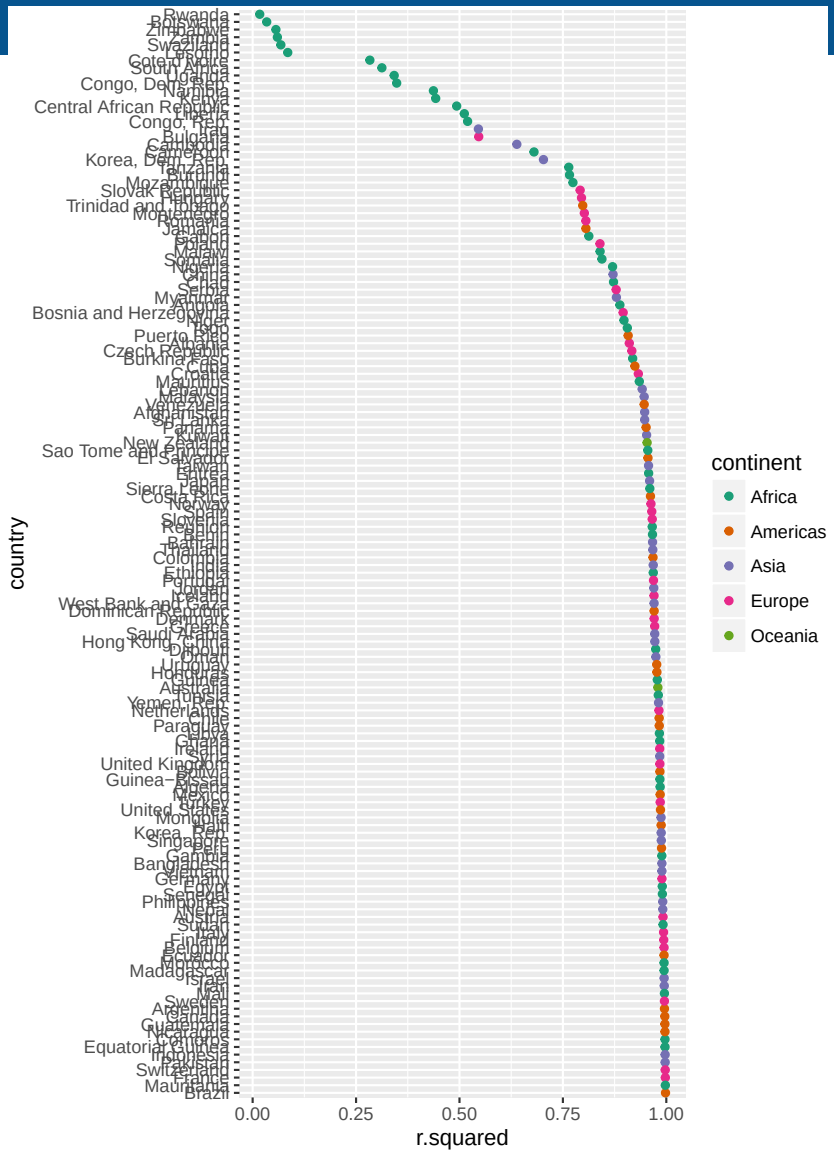
and finally, the visualisation:

```
ggplot(data=coefs, aes(x=`(Intercept)`, y=year1950,
                       colour=continent)) +
  geom_point()
```



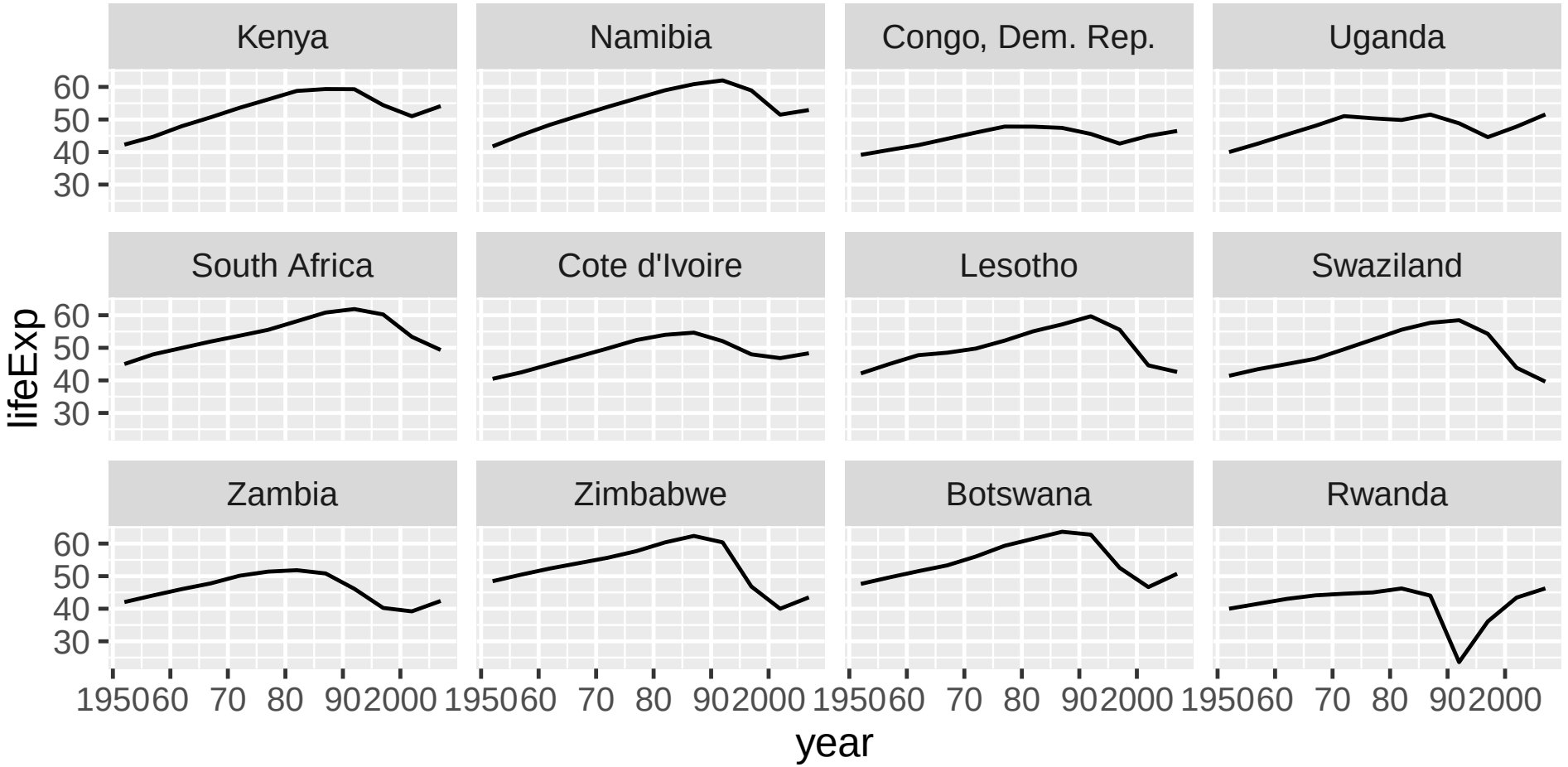
Extract other model diagnostics

```
by_country <- by_country %>%
  unnest(model %>%
           purrr::map(broom::glance))
by_country$country <- reorder(by_country$country,
                              -by_country$r.squared)
ggplot(data=by_country, aes(x=country, y=r.squared,
                            colour=continent)) +
  geom_point() +
  coord_flip() +
  scale_colour_brewer(palette="Dark2")
```



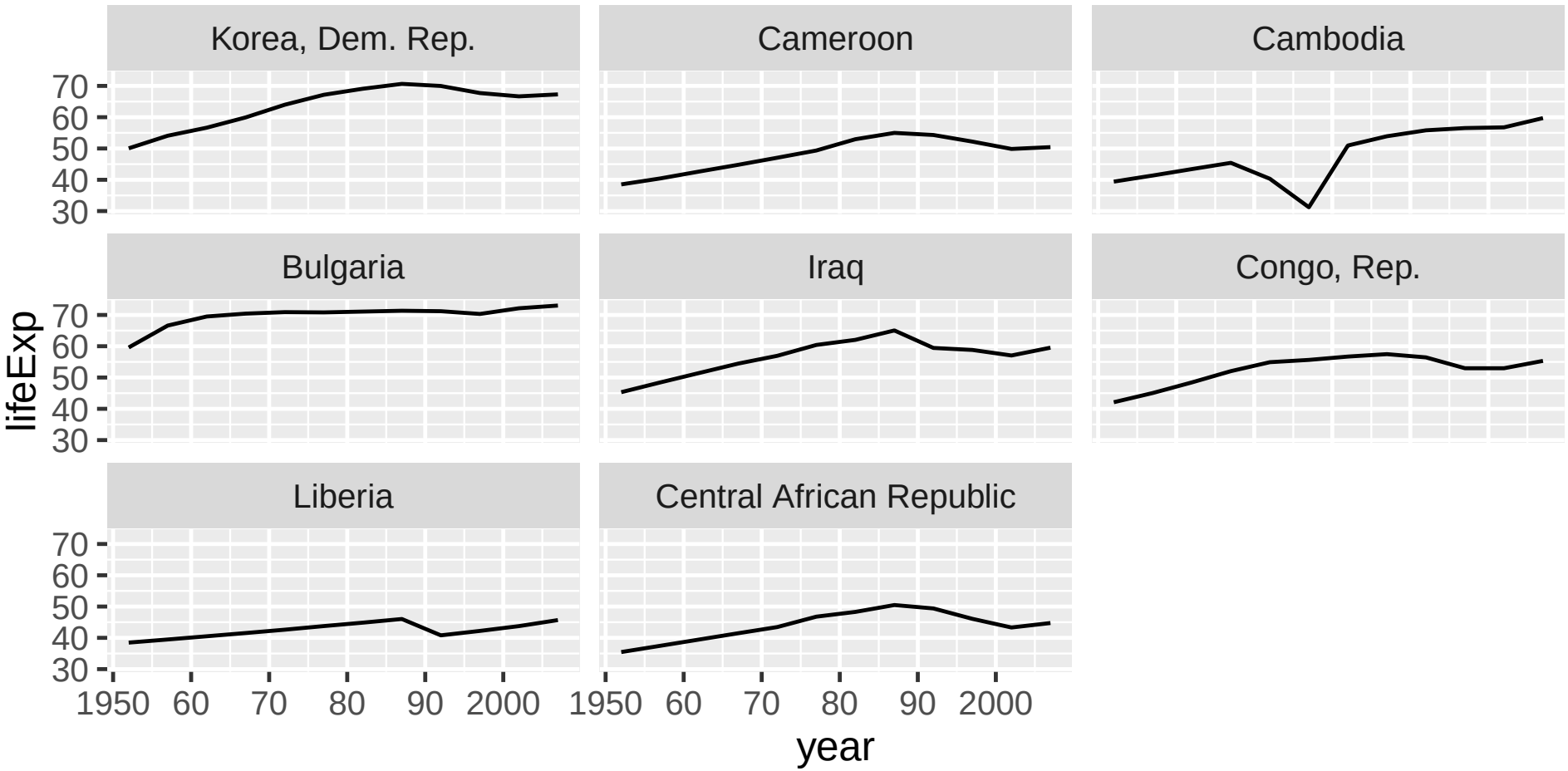
Examine countries with worst fit

```
country_all <- by_country %>%
  unnest(data)
ggplot(data=subset(country_all, r.squared <= 0.45),
       aes(x=year, y=lifeExp)) +
         geom_line() +
  facet_wrap(~country)
```



What patterns do you see, and what do they mean?

Now, let's look at the next worst



What patterns do you see, and what do they mean?

Your turn

- extract residuals for each of the models and store it in a dataset together with country and continent information
- plot residuals across the years and fit a smooth. What does the pattern mean?

Reference material

- Hadley Wickham's gapminder example:
  `http://wombat2016.org/slides/hadley.pdf`
- David Robinson's broom vignettes

Credits

Notes prepared by Di Cook, using material developed by Hadley Wickham and Heike Hofmann.