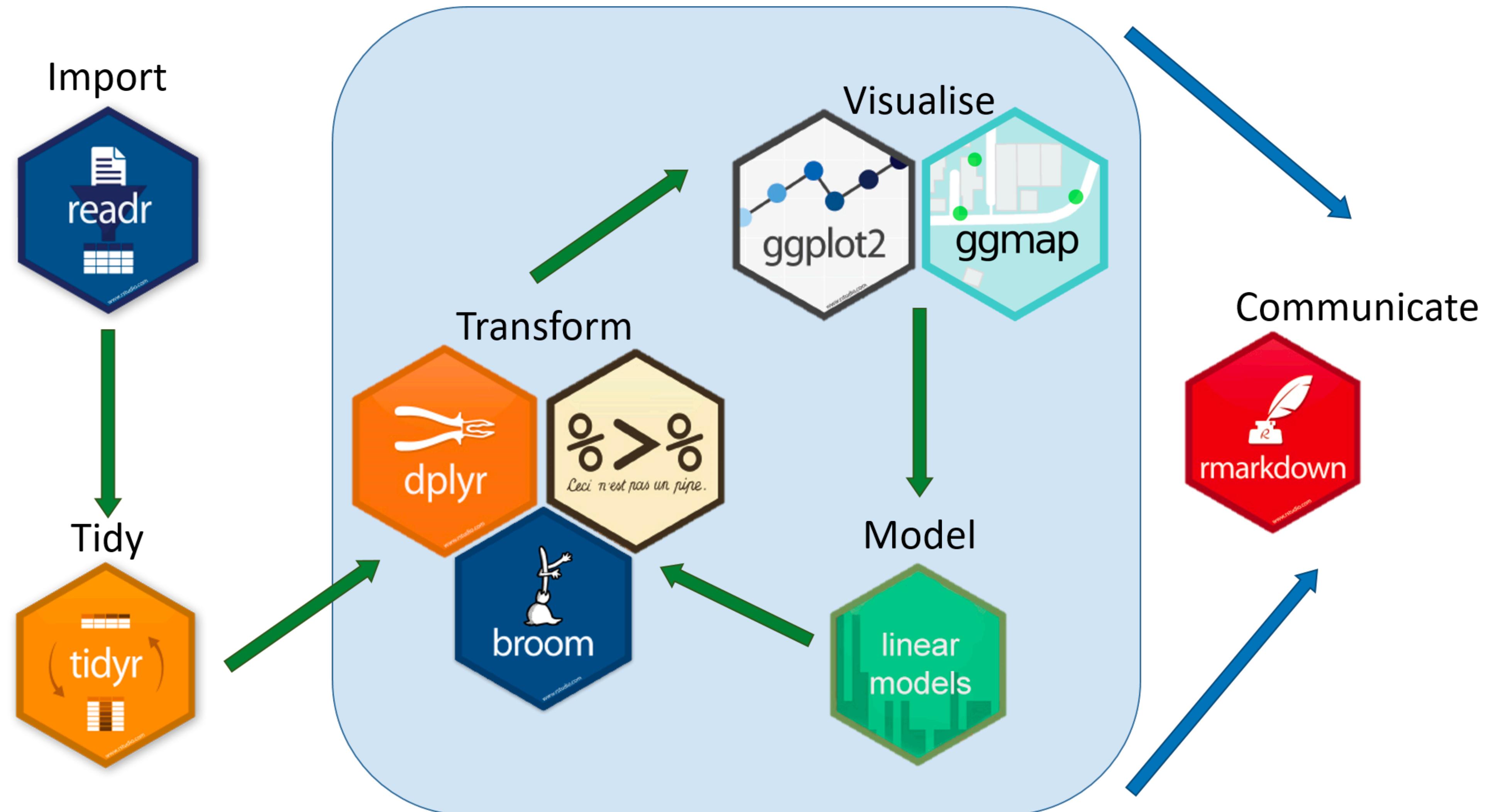


# **The widyr package**

**Pairwise correlations, clustering, and dimensionality  
reduction in the tidyverse**

**David Robinson, 2020-08-15**

# The tidyverse makes many data explorations fluid



# Example: the gapminder dataset of country statistics

```
library(gapminder)  
gapminder
```

# “Find the average life expectancy per year”

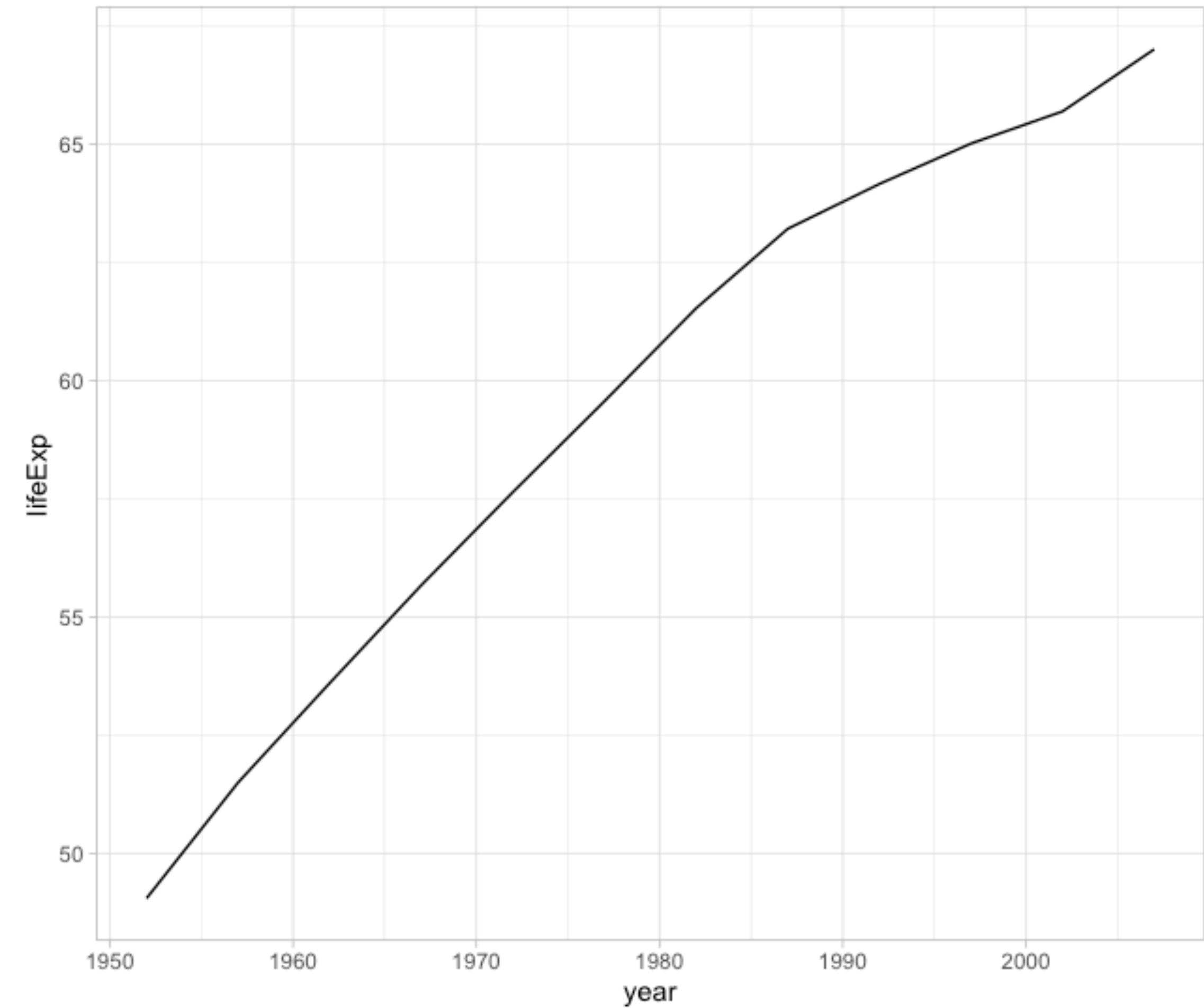
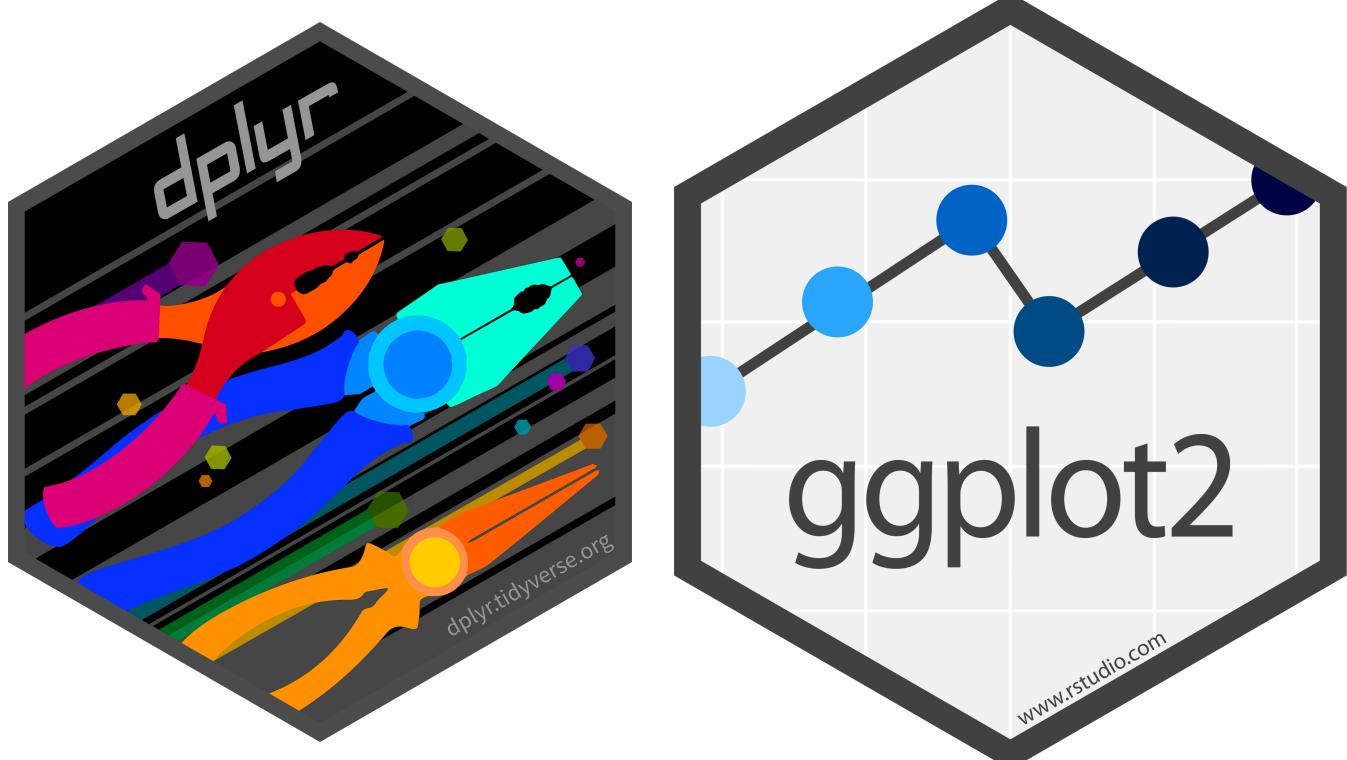
```
gapminder %>%  
  group_by(year) %>%  
  summarize(lifeExp = mean(lifeExp))
```

# A tibble: 12 x 2		
	year	lifeExp
	<int>	<dbl>
1	1952	49.1
2	1957	51.5
3	1962	53.6
4	1967	55.7
5	1972	57.6
6	1977	59.6
7	1982	61.5
8	1987	63.2
9	1992	64.2
10	1997	65.0
11	2002	65.7
12	2007	67.0



# “Plot the average life expectancy per year”

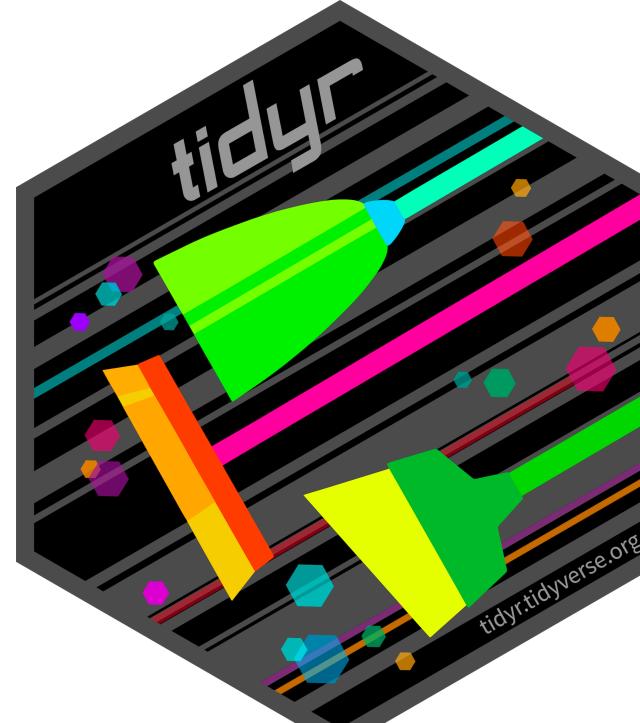
```
gapminder %>%
  group_by(year) %>%
  summarize(lifeExp = mean(lifeExp)) %>%
  ggplot(aes(year, lifeExp)) +
  geom_line()
```



# “Find the slope of increasing life expectancy by country”

```
gapminder %>%
  group_by(country) %>%
  summarize(model = list(lm(lifeExp ~ year))) %>%
  mutate(tidied = map(model, tidy)) %>%
  unnest(tidied) %>%
  filter(term == "year")
```

	country	model	term	estimate	std.error	statistic	p.value
	<fct>	<list>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Afghanistan	<lm>	year	0.275	0.0205	13.5	9.84e- 8
2	Albania	<lm>	year	0.335	0.0332	10.1	1.46e- 6
3	Algeria	<lm>	year	0.569	0.0221	25.7	1.81e-10
4	Angola	<lm>	year	0.209	0.0235	8.90	4.59e- 6
5	Argentina	<lm>	year	0.232	0.00489	47.4	4.22e-13
6	Australia	<lm>	year	0.228	0.0104	21.9	8.67e-10
7	Austria	<lm>	year	0.242	0.00681	35.5	7.44e-12
8	Bahrain	<lm>	year	0.468	0.0274	17.0	1.02e- 8
9	Bangladesh	<lm>	year	0.498	0.0163	30.5	3.37e-11
10	Belgium	<lm>	year	0.209	0.00490	42.7	1.20e-12
# ... with 132 more rows							



**“How is each country’s life expectancy correlated with each other?”**

...

# “How is each country’s life expectancy correlated with each other?”

```
library(widyr)  
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

# A tibble: 20,022 x 3			
	item1 <fct>	item2 <fct>	correlation <dbl>
1	Mauritania	Indonesia	1.00
2	Indonesia	Mauritania	1.00
3	Senegal	Morocco	1.00
4	Morocco	Senegal	1.00
5	West Bank and Gaza	Saudi Arabia	1.00
6	Saudi Arabia	West Bank and Gaza	1.00
7	France	Brazil	0.999
8	Brazil	France	0.999
9	Reunion	Bahrain	0.999
10	Bahrain	Reunion	0.999
# ... with 20,012 more rows			

# How pairwise operations work

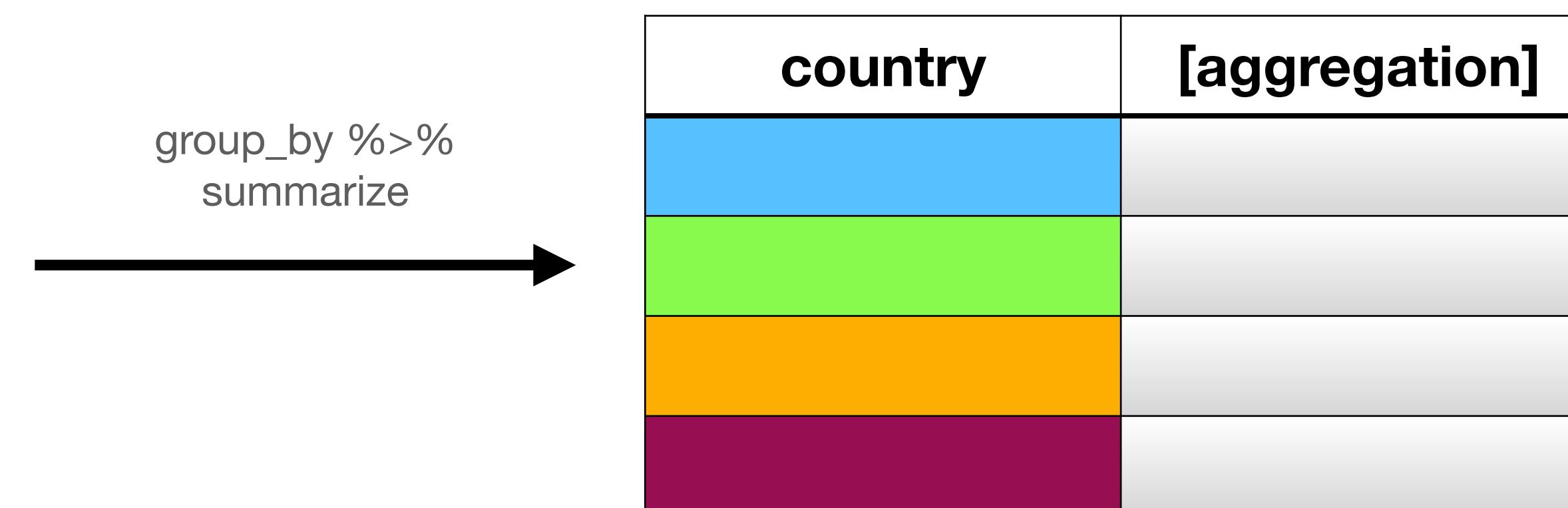
# dplyr is well suited for “aggregate within groups”

country	year	lifeExp
United States	1950	70.8
United States	1951	71.0
United States	1952	71.2
United States	1953	71.4
United States	1954	71.6
United States	1955	71.8
United States	1956	72.0
United States	1957	72.2
United States	1958	72.4
United States	1959	72.6
United States	1960	72.8
United States	1961	73.0
United States	1962	73.2
United States	1963	73.4
United States	1964	73.6
United States	1965	73.8
United States	1966	74.0
United States	1967	74.2
United States	1968	74.4
United States	1969	74.6
United States	1970	74.8
United States	1971	75.0
United States	1972	75.2
United States	1973	75.4
United States	1974	75.6
United States	1975	75.8
United States	1976	76.0
United States	1977	76.2
United States	1978	76.4
United States	1979	76.6
United States	1980	76.8
United States	1981	77.0
United States	1982	77.2
United States	1983	77.4
United States	1984	77.6
United States	1985	77.8
United States	1986	78.0
United States	1987	78.2
United States	1988	78.4
United States	1989	78.6
United States	1990	78.8
United States	1991	79.0
United States	1992	79.2
United States	1993	79.4
United States	1994	79.6
United States	1995	79.8
United States	1996	80.0
United States	1997	80.2
United States	1998	80.4
United States	1999	80.6
United States	2000	80.8
United States	2001	81.0
United States	2002	81.2
United States	2003	81.4
United States	2004	81.6
United States	2005	81.8
United States	2006	82.0
United States	2007	82.2
United States	2008	82.4
United States	2009	82.6
United States	2010	82.8
United States	2011	83.0
United States	2012	83.2
United States	2013	83.4
United States	2014	83.6
United States	2015	83.8
United States	2016	84.0
United States	2017	84.2
United States	2018	84.4
United States	2019	84.6
United States	2020	84.8
United States	2021	85.0
United States	2022	85.2
United States	2023	85.4
United States	2024	85.6
United States	2025	85.8
United States	2026	86.0
United States	2027	86.2
United States	2028	86.4
United States	2029	86.6
United States	2030	86.8
United States	2031	87.0
United States	2032	87.2
United States	2033	87.4
United States	2034	87.6
United States	2035	87.8
United States	2036	88.0
United States	2037	88.2
United States	2038	88.4
United States	2039	88.6
United States	2040	88.8
United States	2041	89.0
United States	2042	89.2
United States	2043	89.4
United States	2044	89.6
United States	2045	89.8
United States	2046	90.0
United States	2047	90.2
United States	2048	90.4
United States	2049	90.6
United States	2050	90.8
United States	2051	91.0
United States	2052	91.2
United States	2053	91.4
United States	2054	91.6
United States	2055	91.8
United States	2056	92.0
United States	2057	92.2
United States	2058	92.4
United States	2059	92.6
United States	2060	92.8
United States	2061	93.0
United States	2062	93.2
United States	2063	93.4
United States	2064	93.6
United States	2065	93.8
United States	2066	94.0
United States	2067	94.2
United States	2068	94.4
United States	2069	94.6
United States	2070	94.8
United States	2071	95.0
United States	2072	95.2
United States	2073	95.4
United States	2074	95.6
United States	2075	95.8
United States	2076	96.0
United States	2077	96.2
United States	2078	96.4
United States	2079	96.6
United States	2080	96.8
United States	2081	97.0
United States	2082	97.2
United States	2083	97.4
United States	2084	97.6
United States	2085	97.8
United States	2086	98.0
United States	2087	98.2
United States	2088	98.4
United States	2089	98.6
United States	2090	98.8
United States	2091	99.0
United States	2092	99.2
United States	2093	99.4
United States	2094	99.6
United States	2095	99.8
United States	2096	100.0
United States	2097	100.2
United States	2098	100.4
United States	2099	100.6
United States	2100	100.8
United States	2101	101.0
United States	2102	101.2
United States	2103	101.4
United States	2104	101.6
United States	2105	101.8
United States	2106	102.0
United States	2107	102.2
United States	2108	102.4
United States	2109	102.6
United States	2110	102.8
United States	2111	103.0
United States	2112	103.2
United States	2113	103.4
United States	2114	103.6
United States	2115	103.8
United States	2116	104.0
United States	2117	104.2
United States	2118	104.4
United States	2119	104.6
United States	2120	104.8
United States	2121	105.0
United States	2122	105.2
United States	2123	105.4
United States	2124	105.6
United States	2125	105.8
United States	2126	106.0
United States	2127	106.2
United States	2128	106.4
United States	2129	106.6
United States	2130	106.8
United States	2131	107.0
United States	2132	107.2
United States	2133	107.4
United States	2134	107.6
United States	2135	107.8
United States	2136	108.0
United States	2137	108.2
United States	2138	108.4
United States	2139	108.6
United States	2140	108.8
United States	2141	109.0
United States	2142	109.2
United States	2143	109.4
United States	2144	109.6
United States	2145	109.8
United States	2146	110.0
United States	2147	110.2
United States	2148	110.4
United States	2149	110.6
United States	2150	110.8
United States	2151	111.0
United States	2152	111.2
United States	2153	111.4
United States	2154	111.6
United States	2155	111.8
United States	2156	112.0
United States	2157	112.2
United States	2158	112.4
United States	2159	112.6
United States	2160	112.8
United States	2161	113.0
United States	2162	113.2
United States	2163	113.4
United States	2164	113.6
United States	2165	113.8
United States	2166	114.0
United States	2167	114.2
United States	2168	114.4
United States	2169	114.6
United States	2170	114.8
United States	2171	115.0
United States	2172	115.2
United States	2173	115.4
United States	2174	115.6
United States	2175	115.8
United States	2176	116.0
United States	2177	116.2
United States	2178	116.4
United States	2179	116.6
United States	2180	116.8
United States	2181	117.0
United States	2182	117.2
United States	2183	117.4
United States	2184	117.6
United States	2185	117.8
United States	2186	118.0
United States	2187	118.2
United States	2188	118.4
United States	2189	118.6
United States	2190	118.8
United States	2191	119.0
United States	2192	119.2
United States	2193	119.4
United States	2194	119.6
United States	2195	119.8
United States	2196	120.0
United States	2197	120.2
United States	2198	120.4
United States	2199	120.6
United States	2200	120.8
United States	2201	121.0
United States	2202	121.2
United States	2203</	

# dplyr is well suited for “aggregate within groups”

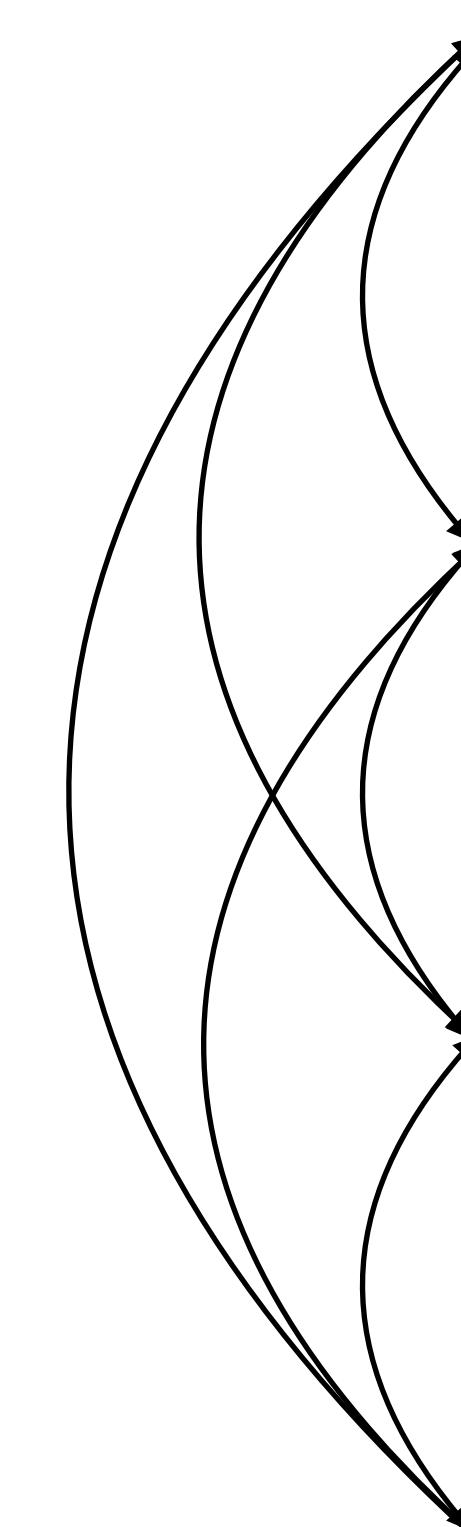
country	year	lifeExp
blue	1962	55.8
blue	1963	56.1
blue	1964	56.4
blue	1965	56.7
green	1962	55.8
green	1963	56.1
green	1964	56.4
green	1965	56.7
orange	1962	55.8
orange	1963	56.1
orange	1964	56.4
orange	1965	56.7
purple	1962	55.8
purple	1963	56.1
purple	1964	56.4
purple	1965	56.7

group\_by %>%  
summarize



# **pairwise\_** operations compare each *pair* of items

country	year	lifeExp
blue	teal	light gray
blue	teal	light gray
blue	dark teal	light gray
green	teal	light gray
green	teal	light gray
green	dark teal	light gray
orange	teal	light gray
orange	teal	light gray
orange	dark teal	light gray
purple	teal	light gray
purple	teal	light gray
purple	dark teal	light gray



# pairwise\_ operations compare each *pair* of items

country	year	lifeExp
blue	cyan	light gray
blue	cyan	light gray
blue	teal	light gray
blue	dark teal	light gray
green	cyan	light gray
green	cyan	light gray
green	teal	light gray
green	dark teal	light gray
orange	cyan	light gray
orange	cyan	light gray
orange	teal	light gray
orange	dark teal	light gray
purple	cyan	light gray
purple	cyan	light gray
purple	teal	light gray
purple	dark teal	light gray

pairwise\_ 

item1	item2	[comparison]
blue	green	light gray
blue	cyan	light gray
blue	orange	light gray
green	blue	light gray
green	orange	light gray
orange	blue	light gray
orange	green	light gray
orange	cyan	light gray
orange	purple	light gray
purple	blue	light gray
purple	green	light gray
purple	orange	light gray

# Correlations in R are traditionally done on matrices

```
bill_length_mm bill_depth_mm flipper_length_mm body_mass_g  
[1,]      39.1       18.7        181     3750  
[2,]      39.5       17.4        186     3800  
[3,]      40.3       18.0        195     3250  
[4,]      36.7       19.3        193     3450  
[5,]      39.3       20.6        190     3650  
[6,]      38.9       17.8        181     3625  
[7,]      39.2       19.6        195     4675  
[8,]      41.1       17.6        182     3200  
[9,]      38.6       21.2        191     3800  
[10,]     34.6       21.1        198     4400  
[11,]     36.6       17.8        185     3700  
[12,]     38.7       19.0        195     3450  
[13,]     42.5       20.7        197     4500  
[14,]     34.4       18.4        184     3325  
[15,]     46.0       21.5        194     4200  
[16,]     37.8       18.3        174     3400  
[17,]     37.7       18.7        180     3600  
[18,]     35.9       19.2        189     3800  
[19,]     38.2       18.1        185     3950  
[20,]     38.8       17.2        180     3800
```



	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
bill_length_mm	1.000000	-0.2286256	0.6530956	0.5894511
bill_depth_mm	-0.2286256	1.0000000	-0.5777917	-0.4720157
flipper_length_mm	0.6530956	-0.5777917	1.0000000	0.8729789
body_mass_g	0.5894511	-0.4720157	0.8729789	1.0000000

```
cor(penguin_matrix)
```



**Me working with any  
data format that's  
not a tidy table**

# The widen-operate-retidy pattern

```
gapminder %>%
  select(country, year, lifeExp) %>%
  pivot_wider(names_from = country, values_from = lifeExp) %>%
  select(-year) %>%
  cor(use = "pairwise.complete.obs") %>%
  as_tibble(rownames = "item1") %>%
  pivot_longer(cols = -item1, names_to = "item2")
```

```
library(widyr)

gapminder %>%
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

# The widen-operate-retidy pattern

```
gapminder %>%  
  select(country, year, lifeExp) %>%  
  pivot_wider(names_from = country, values_from = lifeExp) %>%  
  select(-year) %>%  
  cor(use = "pairwise.complete.obs") %>%  
  as_tibble(rownames = "item1") %>%  
  pivot_longer(cols = -item1, names_to = "item2")
```

Widen

```
library(widyr)  
  
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

# The widen-operate-retidy pattern

```
gapminder %>%
  select(country, year, lifeExp) %>%
  pivot_wider(names_from = country, values_from = lifeExp) %>%
  select(-year) %>%
  cor(use = "pairwise.complete.obs") %>%
  as_tibble(rownames = "item1") %>%
  pivot_longer(cols = -item1, names_to = "item2")
```

Operate

```
library(widyr)

gapminder %>%
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

# The widen-operate-retidy pattern

```
gapminder %>%
  select(country, year, lifeExp) %>%
  pivot_wider(names_from = country, values_from = lifeExp) %>%
  select(-year) %>%
  cor(use = "pairwise.complete.obs") %>%
  as_tibble(rownames = "item1") %>%
  pivot_longer(cols = -item1, names_to = "item2")
```

Re-tidy

```
library(widyr)

gapminder %>%
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

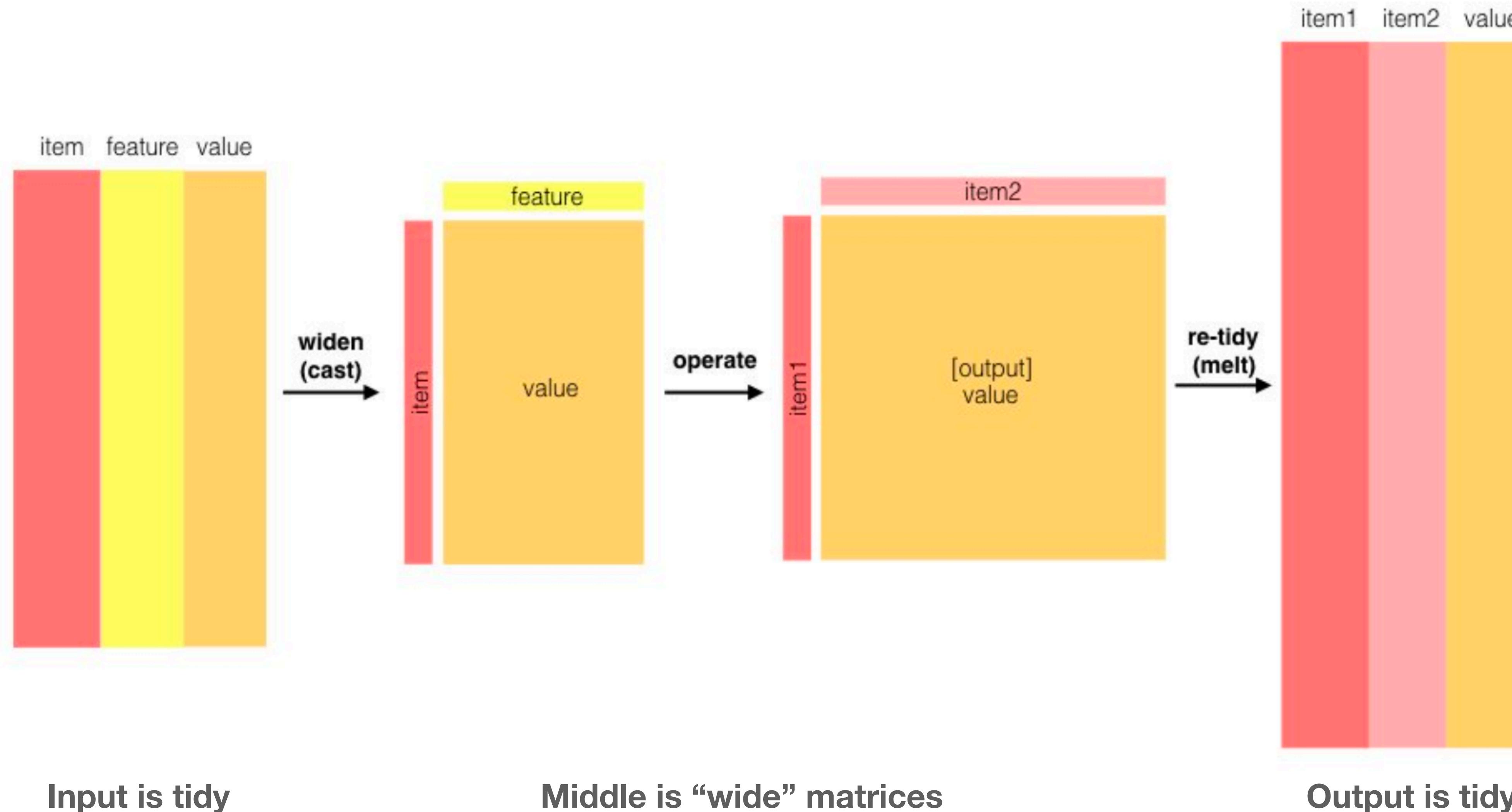
# The widen-operate-retidy pattern

```
gapminder %>%  
  select(country, year, lifeExp) %>%  
  pivot_wider(names_from = country, values_from = lifeExp) %>%  
  select(-year) %>%  
  cor(use = "pairwise.complete.obs") %>%  
  as_tibble(rownames = "item1") %>%  
  pivot_longer(cols = -item1, names_to = "item2")
```

Widen  
Operate  
Re-tidy

```
library(widyr)  
  
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

# The widen-operate-retidy pattern

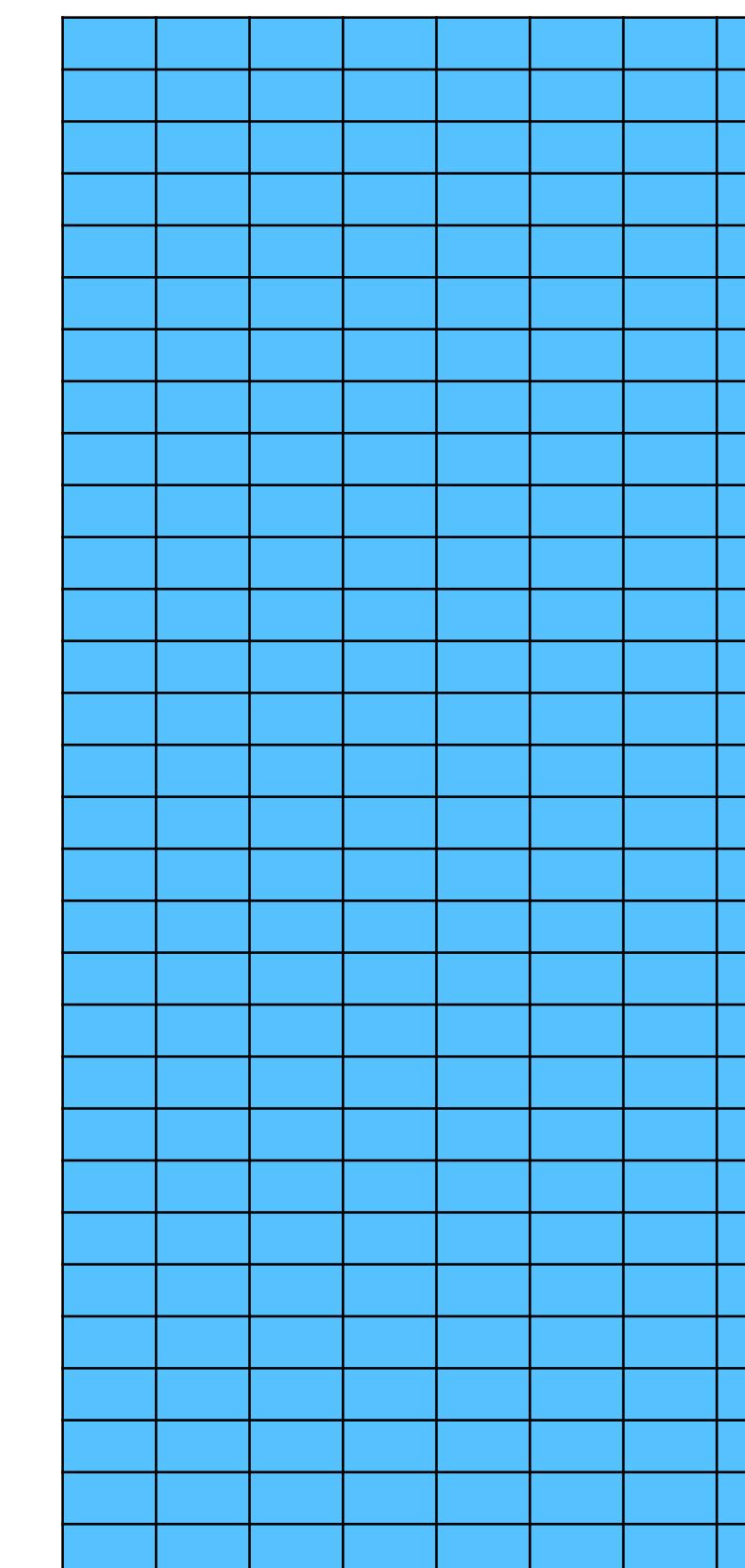


**pairwise\_operations** compares pairs of *items*

gapminder %>%

```
pairwise_cor(country, year, lifeExp, sort = TRUE)
```

## feature



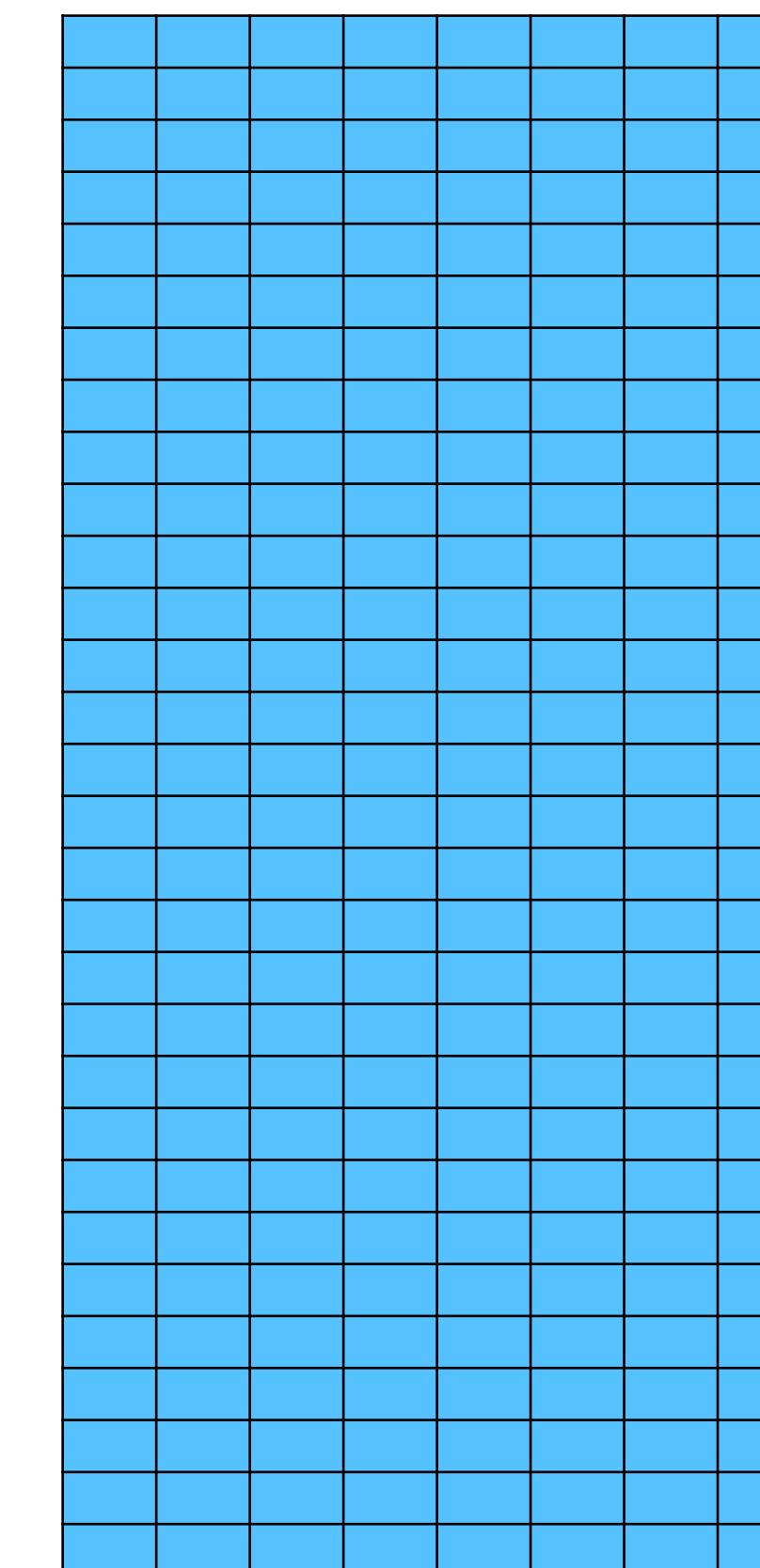
**A *feature* is the second dimension, that links observations together**

A treemap visualization showing the distribution of life expectancy data across countries and years. The data is organized into three main columns: **country**, **year**, and **lifeExp**. The **country** column contains 15 colored rectangles representing different countries. The **year** column contains 15 light blue rectangles representing different years. The **lifeExp** column contains 15 gray rectangles representing different life expectancy values. Arrows point from the **country** and **year** columns towards the **lifeExp** column, indicating a relationship or flow between these categories.

country	year	lifeExp
Blue	Light Blue	Gray
Blue	Light Blue	Gray
Blue	Light Blue	Gray
Blue	Dark Teal	Gray
Green	Light Blue	Gray
Green	Light Blue	Gray
Green	Dark Teal	Gray
Orange	Light Blue	Gray
Orange	Light Blue	Gray
Orange	Dark Teal	Gray
Purple	Light Blue	Gray
Purple	Light Blue	Gray
Purple	Dark Teal	Gray
Red	Dark Teal	Gray

```
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

## feature



item

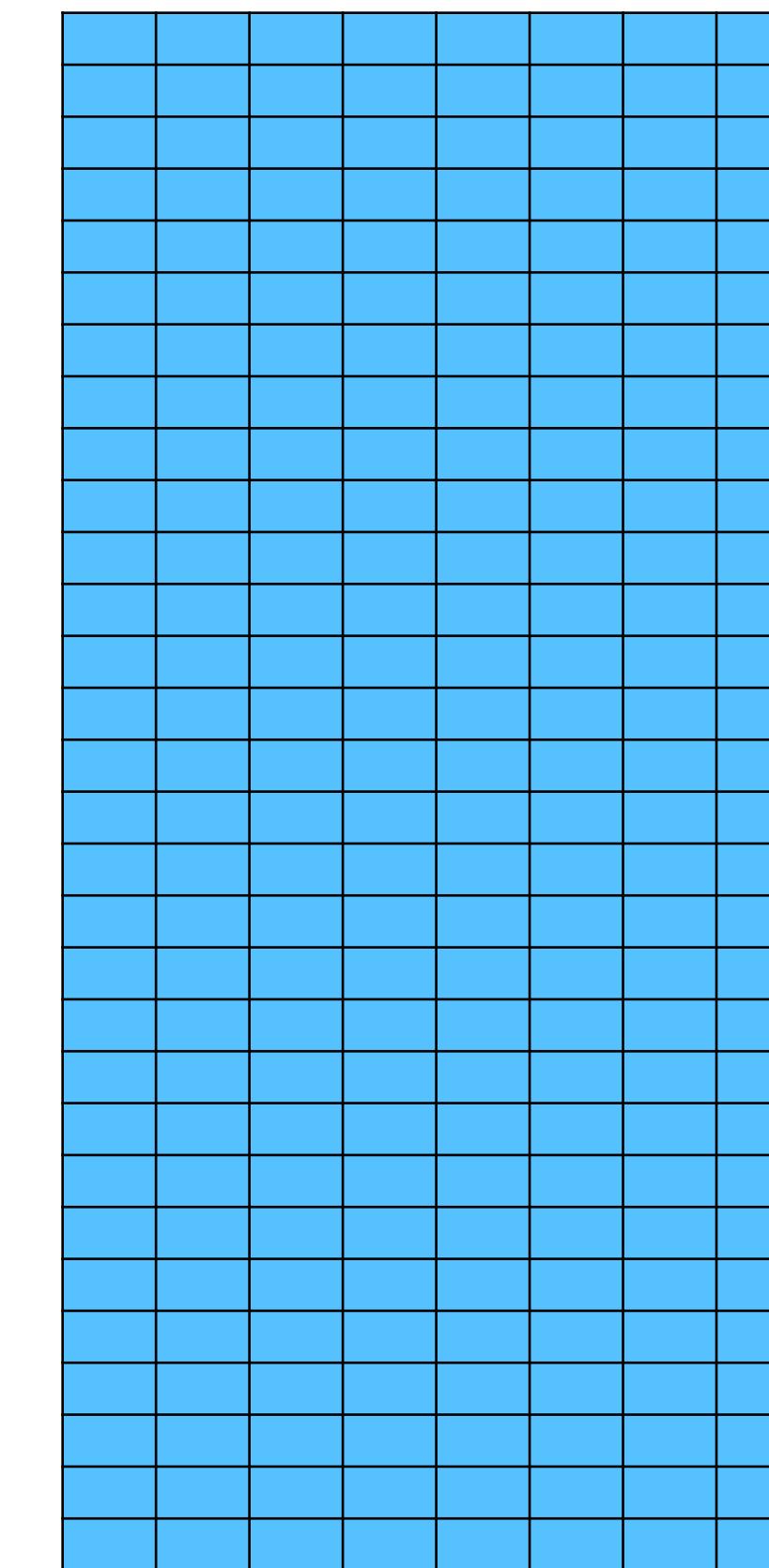
**A *feature* is the second dimension, that links items together**

A treemap visualization showing the distribution of life expectancy across countries and years. The data is organized into three columns: **country**, **year**, and **lifeExp**. The **country** column contains 15 categories, each represented by a different color. The **year** column contains 15 categories, each represented by a different shade of cyan. The **lifeExp** column is represented by a light gray background. A black curved arrow starts from the top right and points towards the bottom left, indicating a flow or relationship between the columns.

country	year	lifeExp
Category 1	Year 1	
Category 2	Year 2	
Category 3	Year 3	
Category 4	Year 4	
Category 5	Year 5	
Category 6	Year 6	
Category 7	Year 7	
Category 8	Year 8	
Category 9	Year 9	
Category 10	Year 10	
Category 11	Year 11	
Category 12	Year 12	
Category 13	Year 13	
Category 14	Year 14	
Category 15	Year 15	

```
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

## feature



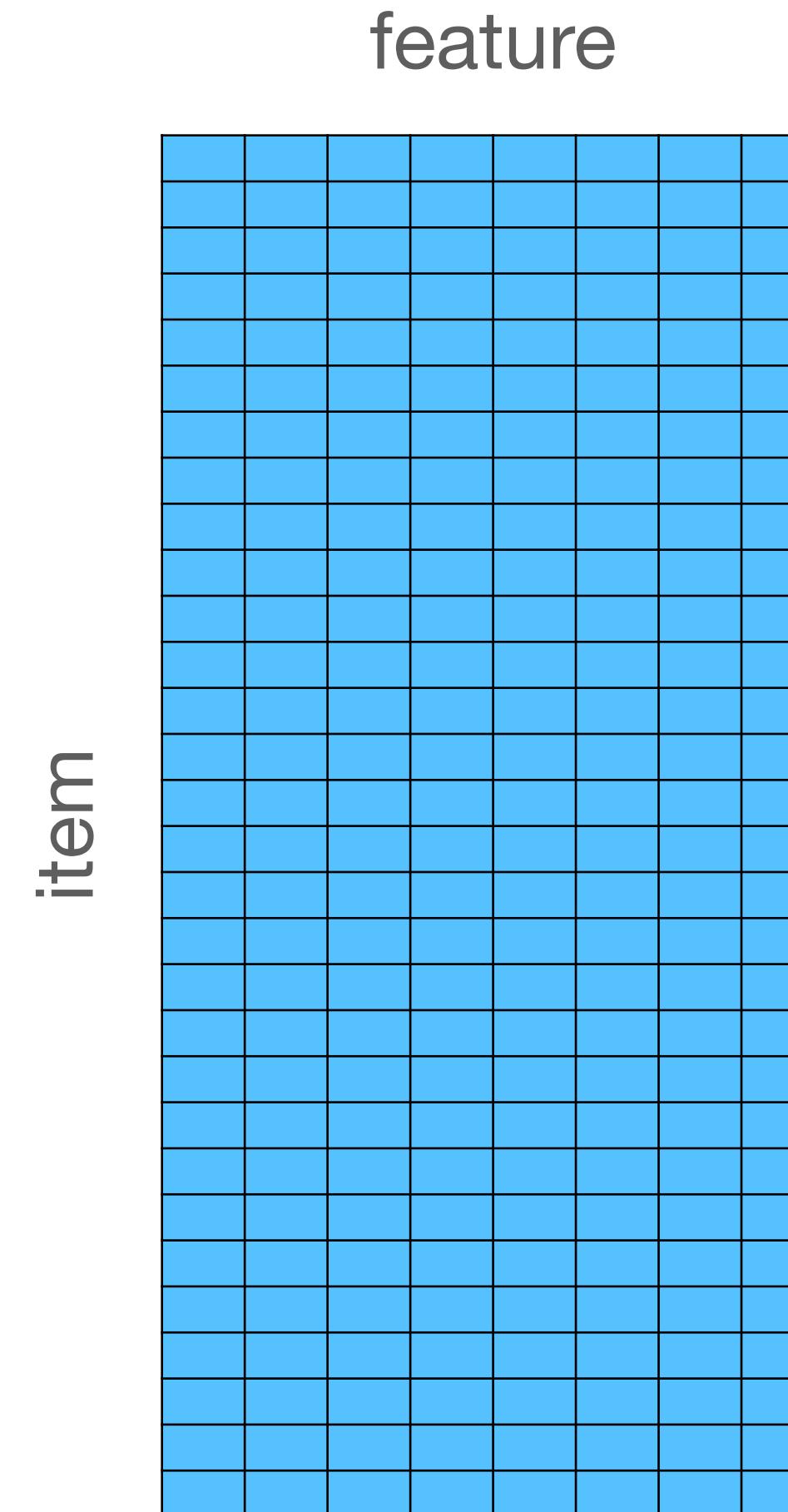
item

*A feature* is the second dimension, that links items together

A treemap visualization showing the distribution of life expectancy across different countries and years. The data is organized into three columns: **country**, **year**, and **lifeExp**. The **country** column contains 12 categories, each represented by a colored rectangle. The **year** column contains 12 categories, each represented by a light gray rectangle. The **lifeExp** column contains 12 categories, each represented by a dark gray rectangle. Arrows point from the **country** and **year** columns towards the **lifeExp** column, indicating a relationship or flow between these variables.

country	year	lifeExp
Blue	Light Gray 1	Dark Gray 1
Blue	Light Gray 2	Dark Gray 2
Blue	Light Gray 3	Dark Gray 3
Green	Light Gray 4	Dark Gray 4
Green	Light Gray 5	Dark Gray 5
Green	Light Gray 6	Dark Gray 6
Orange	Light Gray 7	Dark Gray 7
Orange	Light Gray 8	Dark Gray 8
Orange	Light Gray 9	Dark Gray 9
Purple	Light Gray 10	Dark Gray 10
Purple	Light Gray 11	Dark Gray 11
Purple	Light Gray 12	Dark Gray 12

```
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```

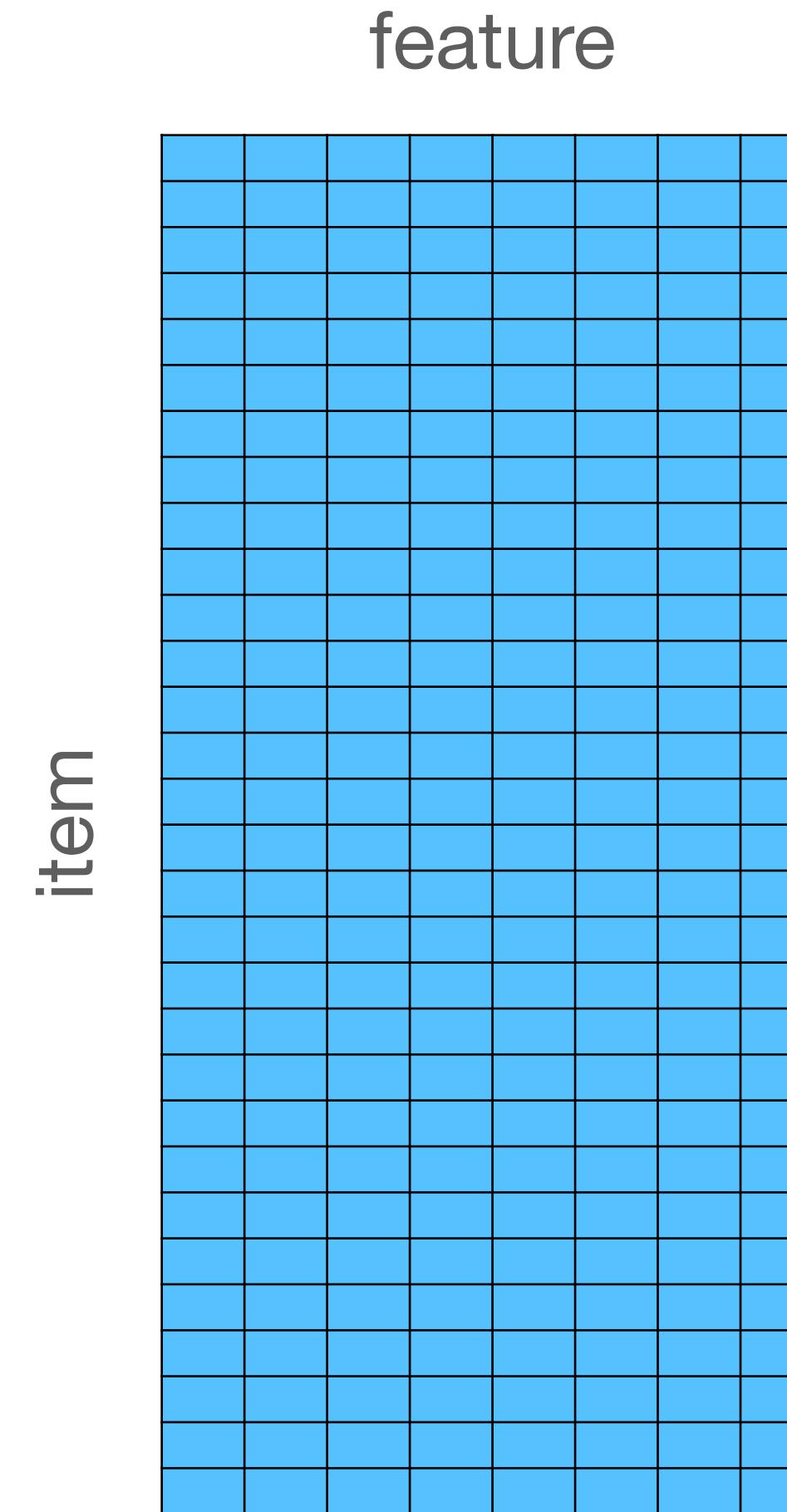


**A *feature* is the second dimension, that links items together**

A treemap visualization showing the distribution of life expectancy across countries and years. The data is organized into three columns: **country**, **year**, and **lifeExp**. The **country** column uses a color gradient from light blue to dark red. The **year** column uses a color gradient from light cyan to dark teal. The **lifeExp** column is represented by a gray gradient background. A black curved arrow points from the right side of the visualization towards the center, highlighting the relationship between the **year** and **lifeExp** columns.

country	year	lifeExp
Light Blue	Light Cyan	Light Gray
Light Blue	Cyan	Light Gray
Light Blue	Dark Cyan	Light Gray
Light Blue	Dark Teal	Light Gray
Light Green	Light Cyan	Light Gray
Light Green	Cyan	Light Gray
Light Green	Dark Cyan	Light Gray
Light Green	Dark Teal	Light Gray
Orange	Light Cyan	Light Gray
Orange	Cyan	Light Gray
Orange	Dark Cyan	Light Gray
Orange	Dark Teal	Light Gray
Maroon	Light Cyan	Light Gray
Maroon	Cyan	Light Gray
Maroon	Dark Cyan	Light Gray
Maroon	Dark Teal	Light Gray

```
gapminder %>%  
  pairwise_cor(country, year, lifeExp, sort = TRUE)
```



**Pairwise example:  
United Nations voting**

# United Nations voting data

```
library(unvotes)
```

```
# A tibble: 733,404 x 4
  rcid country      country_code vote
  <int> <chr>        <chr>       <dbl>
1     3 United States of America US      1
2     3 Canada          CA      -1
3     3 Cuba            CU      1
4     3 Haiti           HT      1
5     3 Dominican Republic DO      1
6     3 Mexico          MX      1
7     3 Guatemala        GT      1
8     3 Honduras         HN      1
9     3 El Salvador      SV      1
10    3 Nicaragua        NI      1
# ... with 733,394 more rows
```

# United Nations voting data

```
library(unvotes)
```

	rqid	country	country_code	vote
	<int>	<chr>	<chr>	<dbl>
1	3	United States of America	US	1
2	3	Canada	CA	-1
3	3	Cuba	CU	1
4	3	Haiti	HT	1
5	3	Dominican Republic	DO	1
6	3	Mexico	MX	1
7	3	Guatemala	GT	1
8	3	Honduras	HN	1
9	3	El Salvador	SV	1
10	3	Nicaragua	NI	1
# ... with 733,394 more rows				

1: Yes

0: Abstain

-1: No

# United Nations voting data

```
library(unvotes)
```

# A tibble: 733,404 x 4

	rcid	country	country_code	vote
	<int>	<chr>	<chr>	<dbl>
1	3	United States of America	US	1
2	3	Canada	CA	-1
3	3	Cuba	CU	1
4	3	Haiti	HT	1
5	3	Dominican Republic	DO	1
6	3	Mexico	MX	1
7	3	Guatemala	GT	1
8	3	Honduras	HN	1
9	3	El Salvador	SV	1
10	3	Nicaragua	NI	1
# ... with 733,394 more rows				

# **Roll call ID (rcid) is our “feature”: How we know which pairs of votes to compare**

# What countries agree/disagree with each other?



	rcid	country	country_code	vote
	<int>	<chr>	<chr>	<dbl>
1	3	United States of America	US	1
2	3	Canada	CA	-1
3	3	Cuba	CU	1
4	3	Haiti	HT	1
5	3	Dominican Republic	DO	1
6	3	Mexico	MX	1
7	3	Guatemala	GT	1
8	3	Honduras	HN	1
9	3	El Salvador	SV	1
10	3	Nicaragua	NI	1
# ... with 733,394 more rows				

# Pairwise correlations of votes

```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE)
```

	item1	item2	correlation
	<chr>	<chr>	<dbl>
1	Slovakia	Czech Republic	0.989
2	Czech Republic	Slovakia	0.989
3	Lithuania	Estonia	0.971
4	Estonia	Lithuania	0.971
5	Lithuania	Latvia	0.970
6	Latvia	Lithuania	0.970
7	Germany	Liechtenstein	0.968
8	Liechtenstein	Germany	0.968
9	Slovakia	Slovenia	0.966
10	Slovenia	Slovakia	0.966
	# ... with 38,602 more rows		

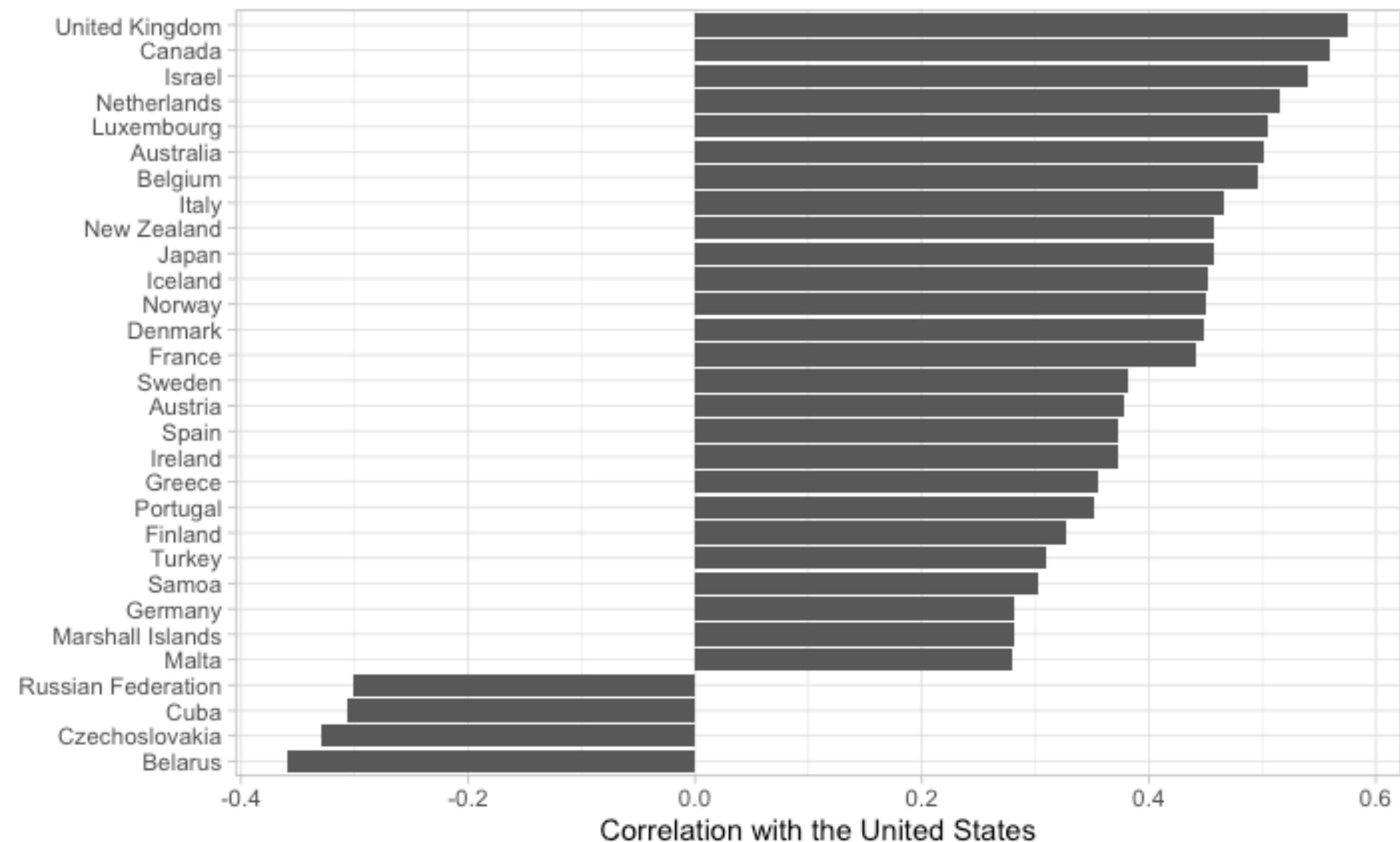
# Pairwise correlations with the United States

```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 == "United States of America")
```

```
# A tibble: 196 x 3
  item1           item2       correlation
  <chr>          <chr>        <dbl>
  1 United States of America United Kingdom 0.576
  2 United States of America Canada        0.559
  3 United States of America Israel        0.540
  4 United States of America Netherlands 0.515
  5 United States of America Luxembourg 0.505
  6 United States of America Australia   0.502
  7 United States of America Belgium    0.496
  8 United States of America Italy      0.467
  9 United States of America New Zealand 0.458
 10 United States of America Japan     0.458
# ... with 186 more rows
```

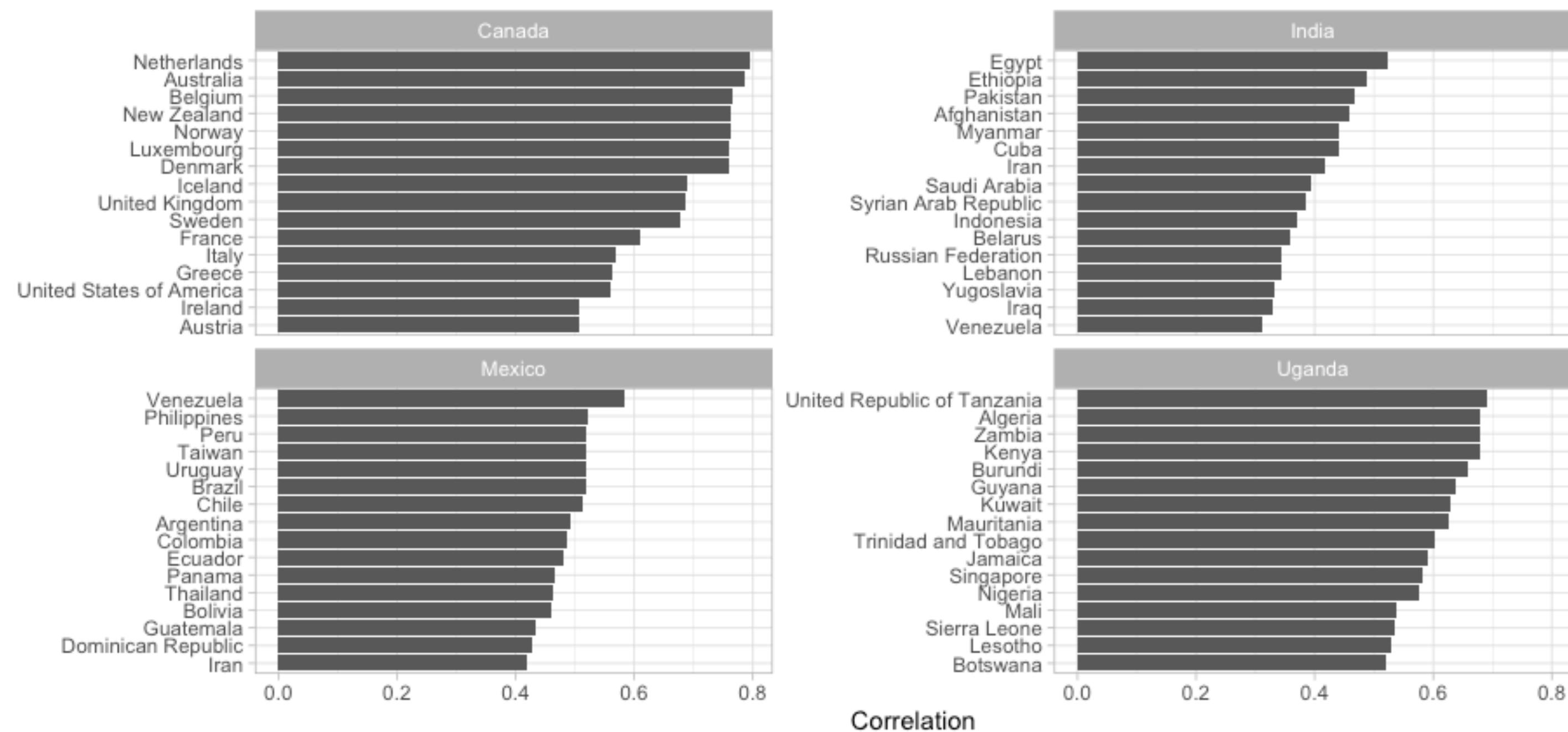
# Highest/lowest correlations with the United States

```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 == "United States of America") %>%
  top_n(30, abs(correlation)) %>%
  ggplot(aes(correlation, reorder(item2, correlation))) +
  geom_col() +
  labs(x = "Correlation with the United States", y = "")
```



# Highest correlations faceted by country

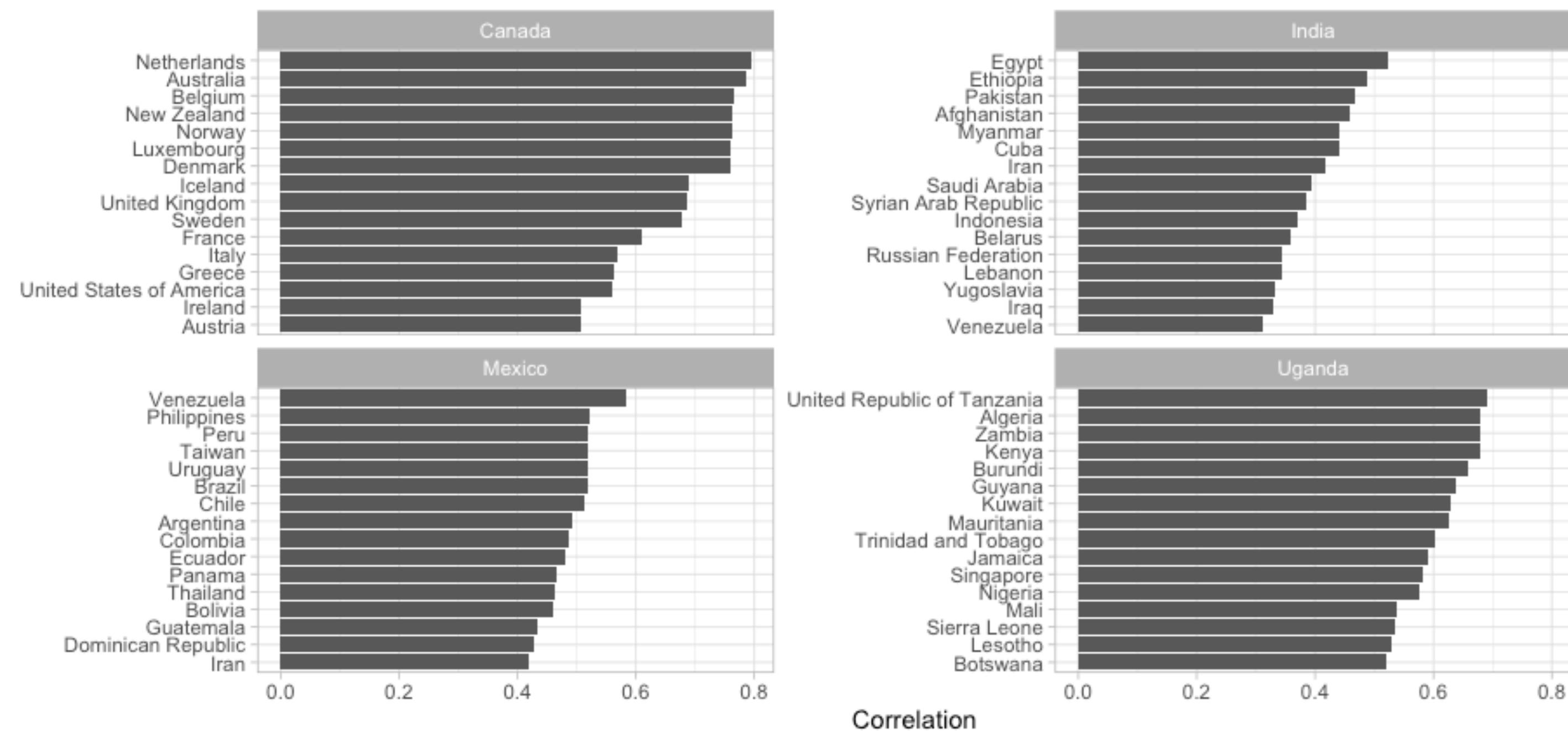
```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 %in% c("Uganda", "India", "Canada", "Mexico")) %>%
  group_by(item1) %>%
  top_n(16, abs(correlation)) %>%
  mutate(item2 = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(correlation, item2)) +
  geom_col() +
  facet_wrap(~ item1, scales = "free_y") +
  scale_y_reordered() +
  labs(x = "Correlation", y = "")
```



# Highest correlations faceted by country

widyr

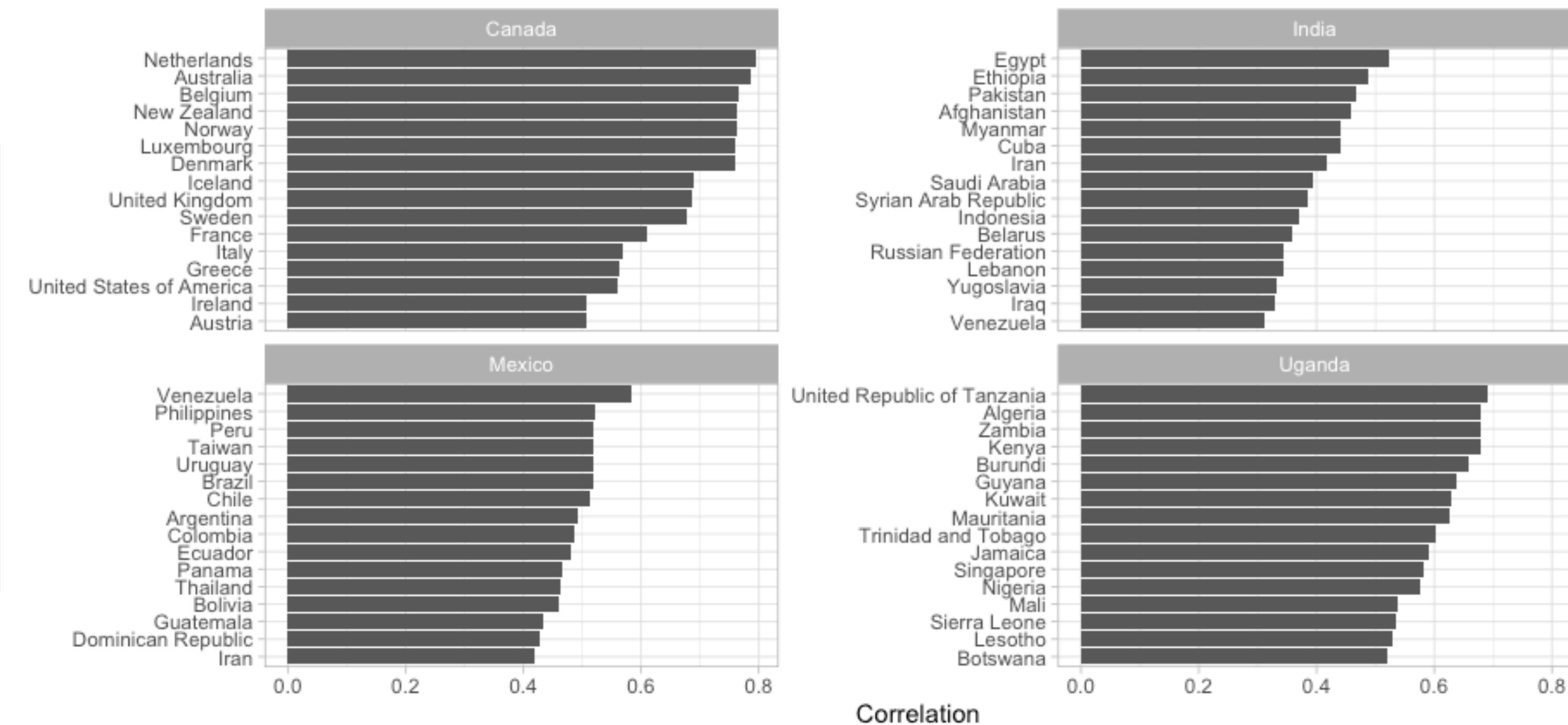
```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 %in% c("Uganda", "India", "Canada", "Mexico")) %>%
  group_by(item1) %>%
  top_n(16, abs(correlation)) %>%
  mutate(item2 = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(correlation, item2)) +
  geom_col() +
  facet_wrap(~ item1, scales = "free_y") +
  scale_y_reordered() +
  labs(x = "Correlation", y = "")
```



# Highest correlations faceted by country

dplyr

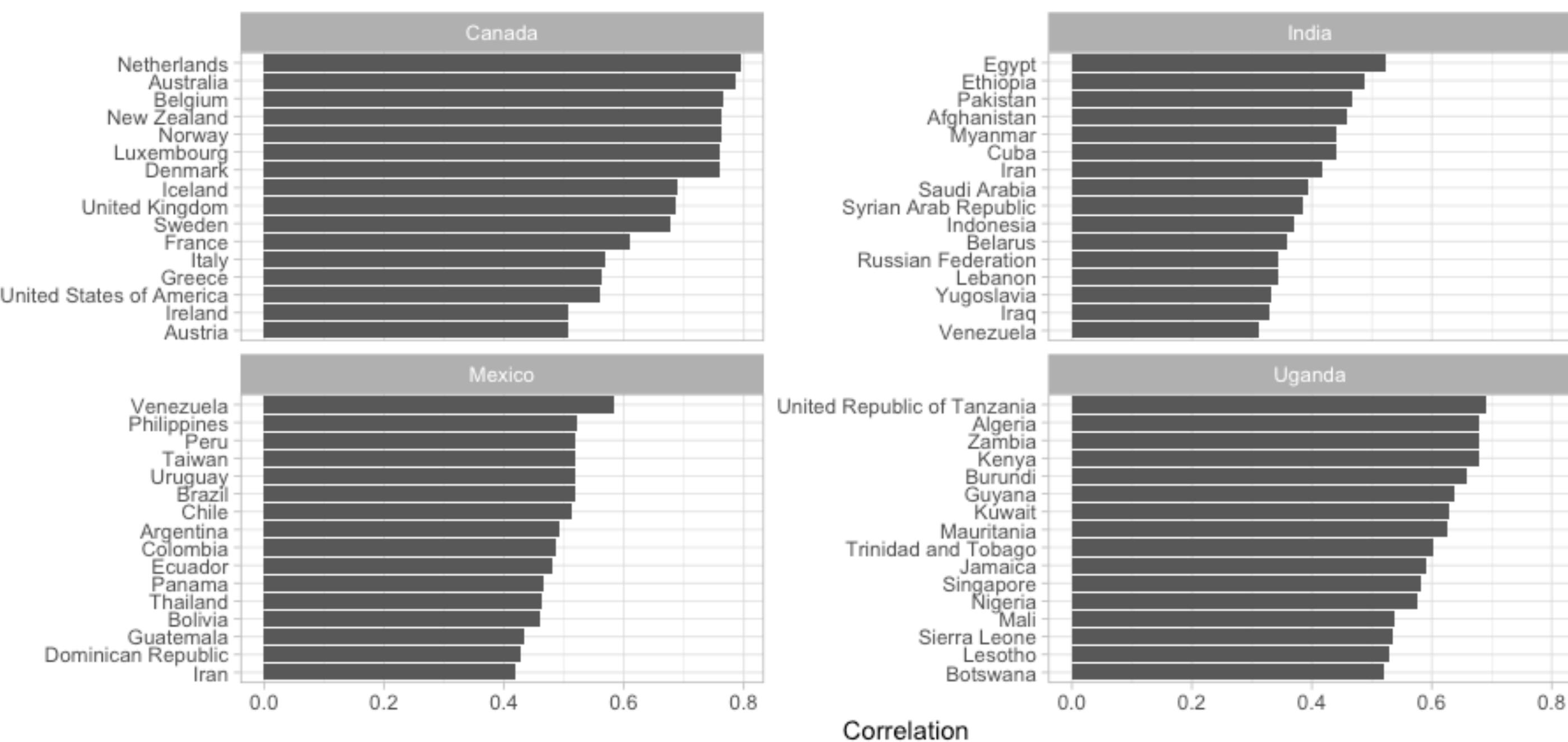
```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 %in% c("Uganda", "India", "Canada", "Mexico")) %>%
  group_by(item1) %>%
  top_n(16, abs(correlation)) %>%
  mutate(item2 = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(correlation, item2)) +
  geom_col() +
  facet_wrap(~ item1, scales = "free_y") +
  scale_y_reordered() +
  labs(x = "Correlation", y = "")
```



# Highest correlations faceted by country

```
votes %>%
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%
  filter(item1 %in% c("Uganda", "India", "Canada", "Mexico")) %>%
  group_by(item1) %>%
  top_n(16, abs(correlation)) %>%
  mutate(item2 = reorder_within(item2, correlation, item1)) %>%
  ggplot(aes(correlation, item2)) +
  geom_col() +
  facet_wrap(~ item1, scales = "free_y") +
  scale_y_reordered() +
  labs(x = "Correlation", y = "")
```

ggplot2



**Pairwise example:  
Word co-occurrence**

# Hacker News titles

Y Hacker News new   past   comments   ask   show   jobs   submit					
1. ▲ Fungus at Chernobyl absorbs nuclear radiation via radiosynthesis (technologynetworks.com)					
76 points by atlasshorts 2 hours ago   hide   22 comments					
2. ▲ J Notation as a Tool of Thought (hillelwayne.com)					
57 points by janvdbberg 4 hours ago   hide   23 comments					
3. ▲ Write Your Own Virtual Machine (justinmeiners.github.io)					
91 points by ChankeyPathak 5 hours ago   hide   9 comments					
4. ▲ Mozilla's Uncertain Future (civilityandtruth.com)					
137 points by jonathankoren 4 hours ago   hide   111 comments					
5. ▲ India announces plan to connect 600k villages with optical fiber in 1000 days (indianexpress.com)					
66 points by ra7 2 hours ago   hide   18 comments					
6. ▲ A review of Bel, Eve, and a silly VR rant (gist.github.com)					
22 points by lemming 3 hours ago   hide   discuss					
7. ▲ OpenVMS on x86 (vmssoftware.com)					
28 points by gjvc 3 hours ago   hide   16 comments					
8. ▲ Amazon's ML University is making its online courses available to the public (amazon.science)					
7 points by karxxm 2 hours ago   hide   discuss					
9. ▲ Using an old BlackBerry as a portable SSH or Telnet terminal (rqsall.com)					
32 points by todacerdoti 4 hours ago   hide   17 comments					
10. ▲ It's strange what people put up with in C# (gist.github.com)					
11 points by dustinmoris 1 hour ago   hide   2 comments					
11. ▲ "The Edge of Chaos" (2017) (bactra.org)					
7 points by meanie 1 hour ago   hide   3 comments					
12. ▲ Factorio 1.0 (factorio.com)					
1721 points by Akronymus 1 day ago   hide   561 comments					
13. ▲ Ghost.org deleted my website (postapathy.substack.com)					
156 points by davidbarker 2 hours ago   hide   136 comments					
14. ▲ Precise Higher-Order Meshing of Curved 2D Domains (uos.de)					
24 points by wowsig 6 hours ago   hide   1 comment					
15. ▲ PyIDM – Python open-source alternative to Internet Download Manager (github.com)					
76 points by URfejk 10 hours ago   hide   15 comments					
16. ▲ Welders set off Beirut blast while securing explosives (maritime-executive.com)					
566 points by tafda 17 hours ago   hide   474 comments					
17. ▲ Duality of Vector Spaces (2017) (solmaz.io)					
31 points by hosolmaz 6 hours ago   hide   9 comments					
18. ▲ Brain Oriented Programming (tobeva.com)					
47 points by pbw 6 hours ago   hide   32 comments					
19. ▲ Launch HN: Tella (YC S20) – Collaborative video editing in the browser					
178 points by 9ranty 19 hours ago   hide   74 comments					
20. ▲ Dear Google Cloud: Your Deprecation Policy Is Killing You (medium.com)					
241 points by bigain 7 hours ago   hide   119 comments					

# A tibble: 99,996 x 3

post_id	date	title
<int>	<date>	<chr>
1	2019-01-01	Learn the Rules Like a Pro, So You Can ...
2	2019-01-01	Upgrading the Nginx Executable on the F...
3	2019-01-01	Trendism and cognitive stagnation
4	2019-01-01	DNS Records Checker
5	2019-01-01	UX Designer's guide to effective retros...
6	2019-01-01	Nevralgiile faciale tratamente naturiste
7	2019-01-01	Online tutoring app Byju touches \$3.8B ...
8	2019-01-01	How to Play PUBG on Pc Using This Simpl...
9	2019-01-01	Simya Koleji Türkiye Geneli Bursluluk S...
10	2019-01-01	At the twilight of Moore's Law

# ... with 99,986 more rows

Adapted from [Training, Evaluating, and Interpreting Topic Models](#) by Julia Silge

# Tokenizing Hacker News titles with tidytext

```
hacker_news_words <- hacker_news_text %>%
  unnest_tokens(word, title) %>%
  anti_join(stop_words, by = "word") %>%
  filter(!str_detect(word, "[0-9]+")) %>%
  add_count(word, name = "word_total") %>%
  filter(word_total >= 250)
```

```
# A tibble: 120,106 x 3
  post_id date      word
  <int>   <date>    <chr>
1       1 2019-01-01 learn
2       1 2019-01-01 pro
3       5 2019-01-01 guide
4       7 2019-01-01 online
5       7 2019-01-01 app
6       8 2019-01-01 play
7       8 2019-01-01 simple
8      10 2019-01-01 law
9      15 2019-01-01 data
10     16 2019-01-01 design
# ... with 120,096 more rows
```



# Pairwise co-occurrences of words

```
hacker_news_words %>%
  pairwise_cor(word, post_id, sort = TRUE)
```

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\bullet}n_{0\bullet}n_{\bullet 0}n_{\bullet 1}}}$$

Phi coefficient

```
# A tibble: 51,302 x 3
  item1    item2    correlation
  <chr>    <chr>    <dbl>
1 machine  learning 0.505
2 learning machine 0.505
3 media    social   0.493
4 social   media   0.493
5 networks neural  0.472
6 neural   networks 0.472
7 climate  change  0.443
8 change   climate 0.443
9 react    native  0.356
10 native   react  0.356
# ... with 51,292 more rows
```

# Pairwise co-occurrences of words

```
hacker_news_words %>%  
  pairwise_cor(word, post_id, sort = TRUE) %>%  
  filter(item1 == "data")
```

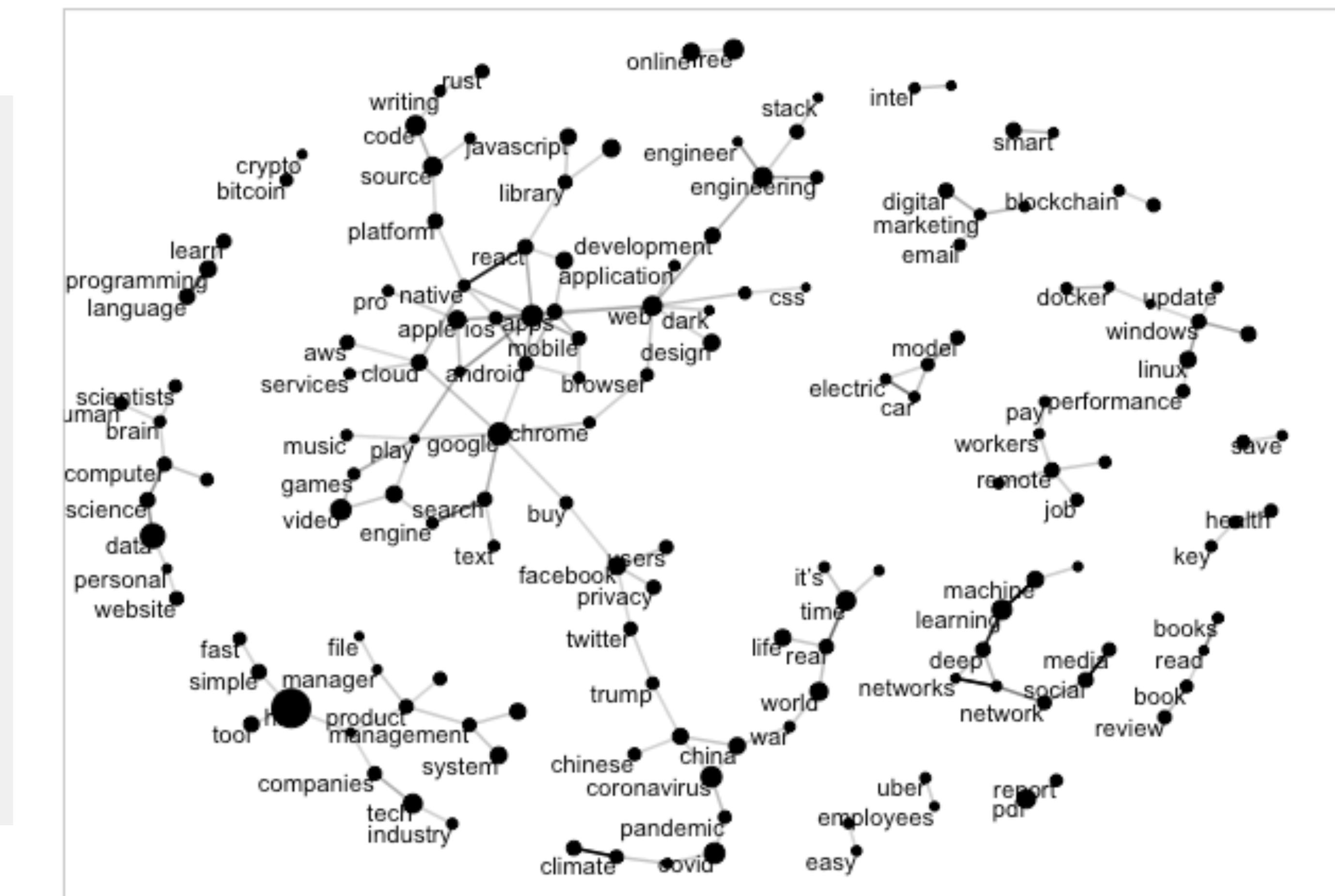
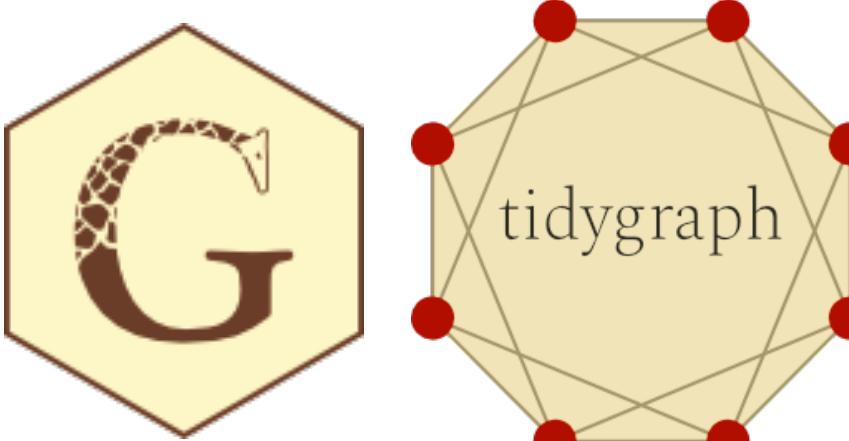
	item1	item2	correlation
	<chr>	<chr>	<dbl>
1	data	science	0.140
2	data	personal	0.0377
3	data	scientists	0.0351
4	data	user	0.0329
5	data	access	0.0294
6	data	analysis	0.0291
7	data	privacy	0.0264
8	data	machine	0.0177
9	data	cloud	0.0140
10	data	learning	0.0138
# ... with 216 more rows			

# Network plots with tidy graph + ggraph

```
library(ggraph)
library(tidygraph)

word_counts <- hacker_news_words %>%
  count(word, sort = TRUE)

hacker_news_words %>%
  pairwise_cor(word, post_id, sort = TRUE) %>%
  head(300) %>%
  as_tbl_graph() %>%
  inner_join(word_counts, by = c(name = "word")) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation)) +
  geom_node_point(aes(size = n)) +
  geom_node_text(aes(label = name), check_overlap = TRUE,
                vjust = 1, hjust = 1, size = 3) +
  theme(legend.position = "none")
```

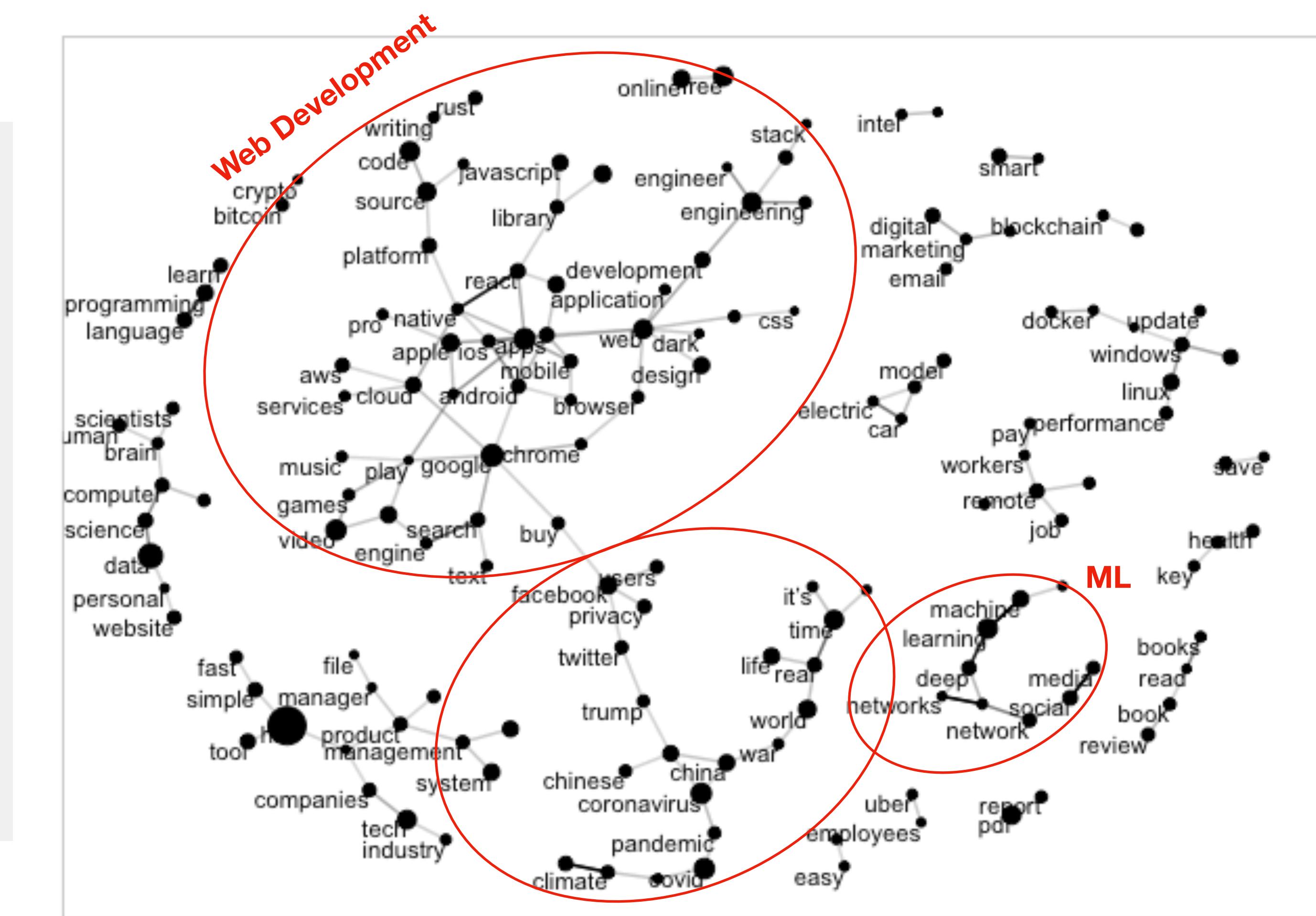
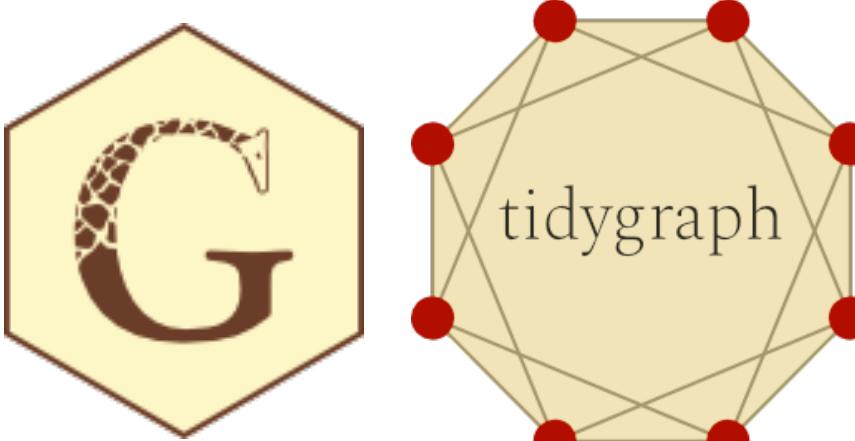


# Network plots with tidy graph + ggraph

```
library(ggraph)
library(tidygraph)

word_counts <- hacker_news_words %>%
  count(word, sort = TRUE)

hacker_news_words %>%
  pairwise_cor(word, post_id, sort = TRUE) %>%
  head(300) %>%
  as_tbl_graph() %>%
  inner_join(word_counts, by = c(name = "word")) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation)) +
  geom_node_point(aes(size = n)) +
  geom_node_text(aes(label = name), check_overlap = TRUE,
                vjust = 1, hjust = 1, size = 3) +
  theme(legend.position = "none")
```

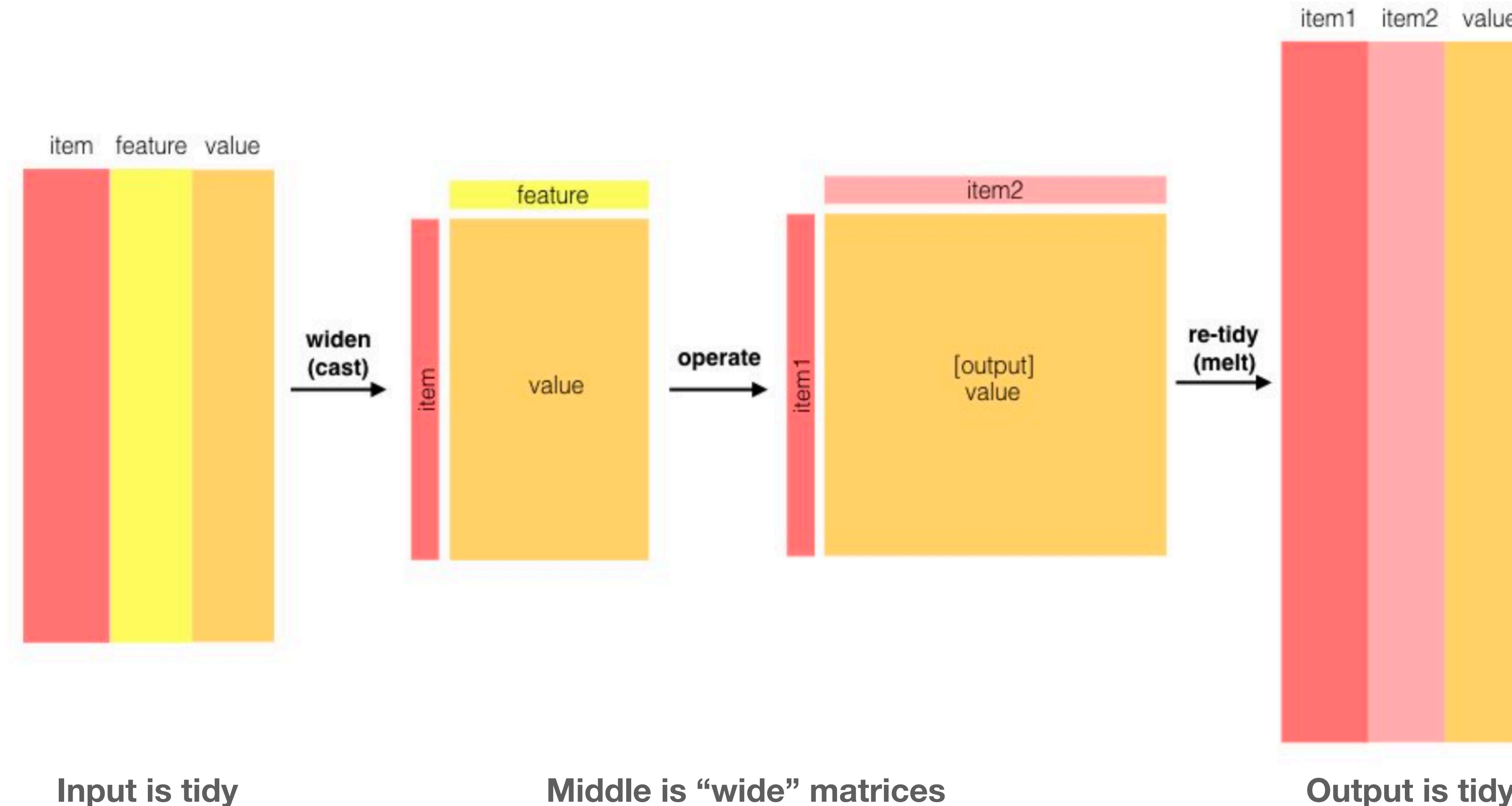


# Other pairwise operations in `widyr`

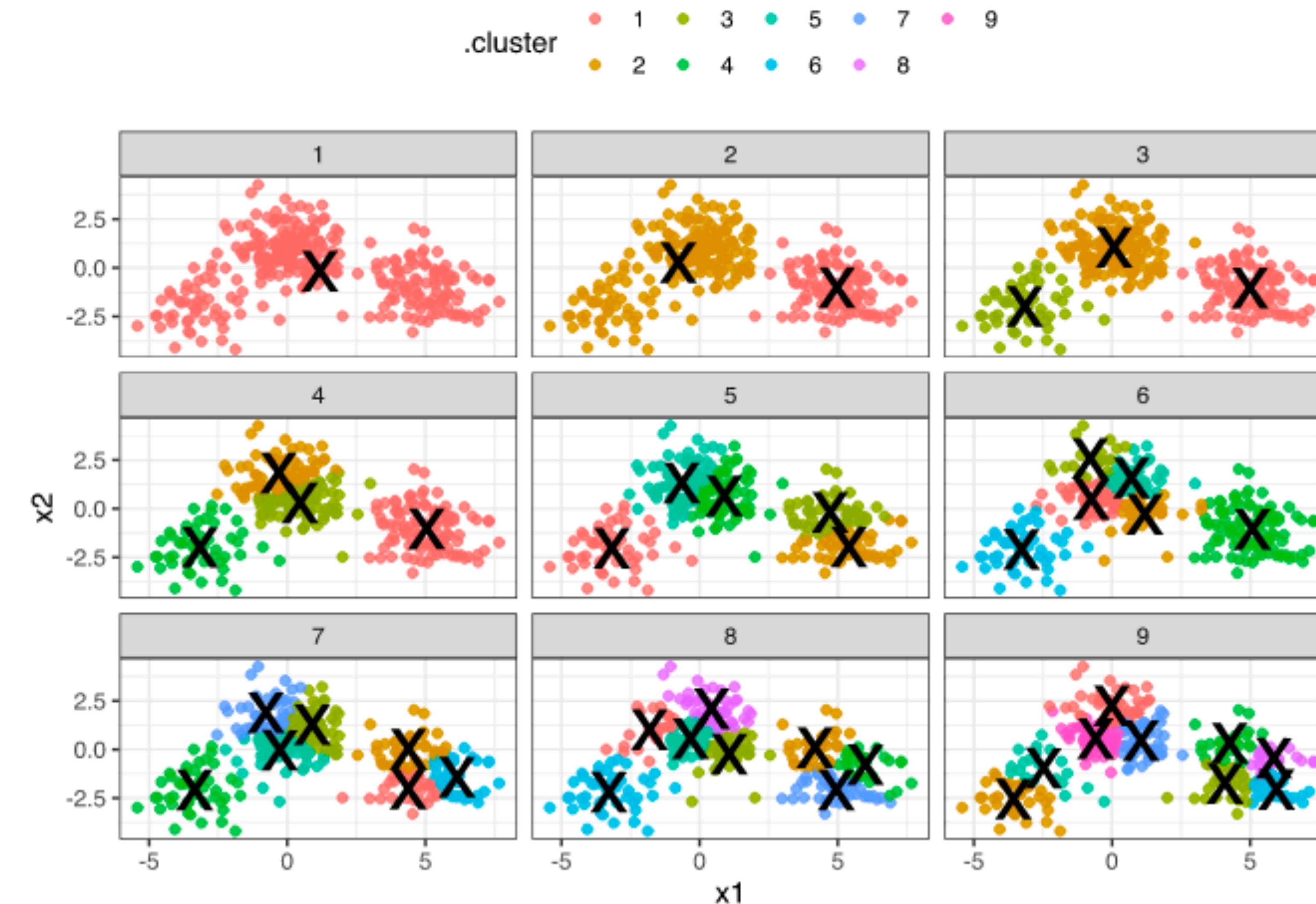
- `pairwise_count` How often do these two items appear together?
- `pairwise_dist` Euclidean/Manhattan/etc distance
- `pairwise_similarity` Cosine similarity
- `pairwise_pmi` Pairwise mutual information
- `pairwise_delta` Calculate Burrows delta (for authorship attribution)

**Widely example:  
clustering + dimensionality reduction**

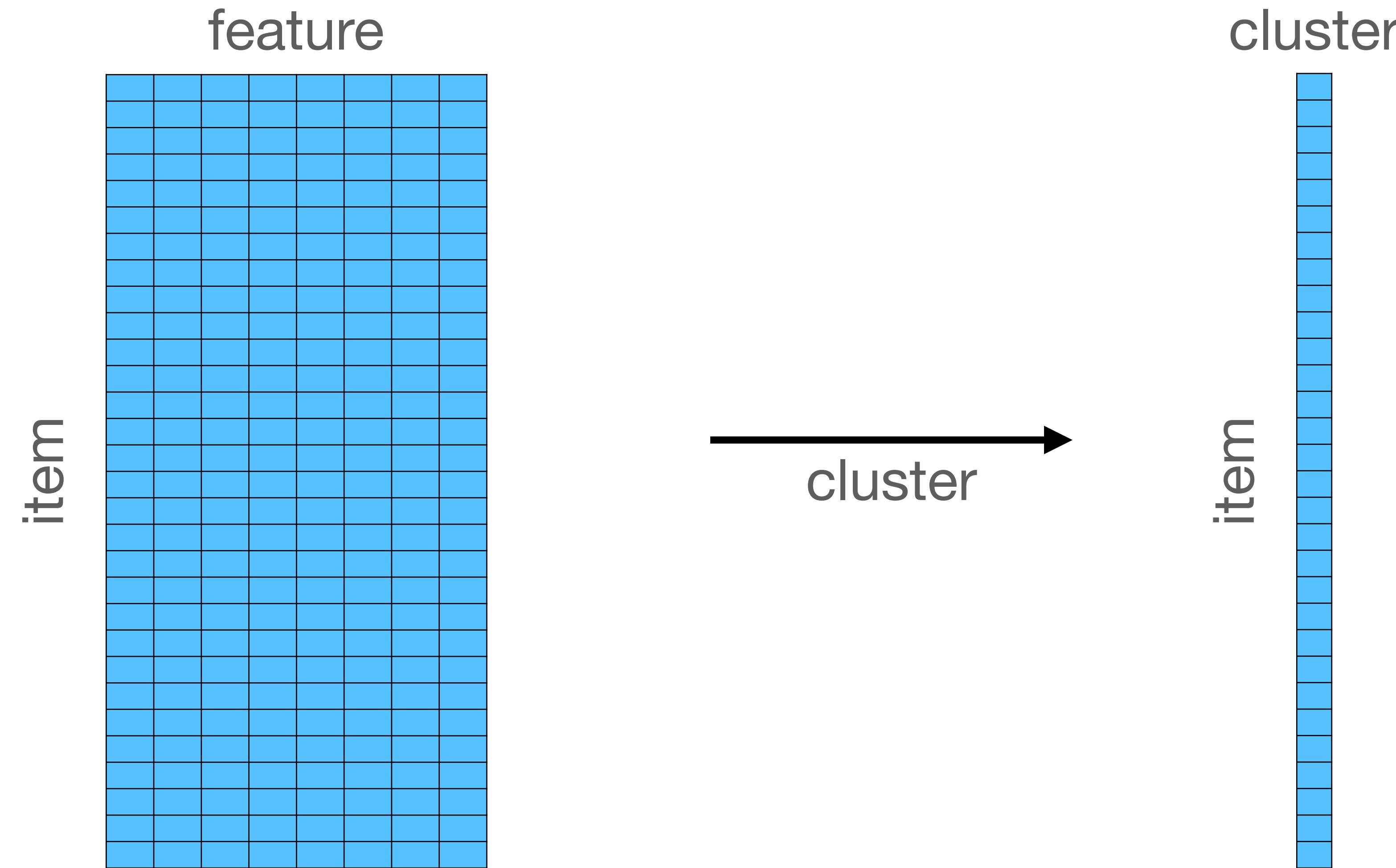
# The widen-operate-retidy pattern is very flexible



# K-means is a classic approach to clustering



# Clustering is an example of a “wide” operation



# widely\_kmeans performs clustering on tidy data

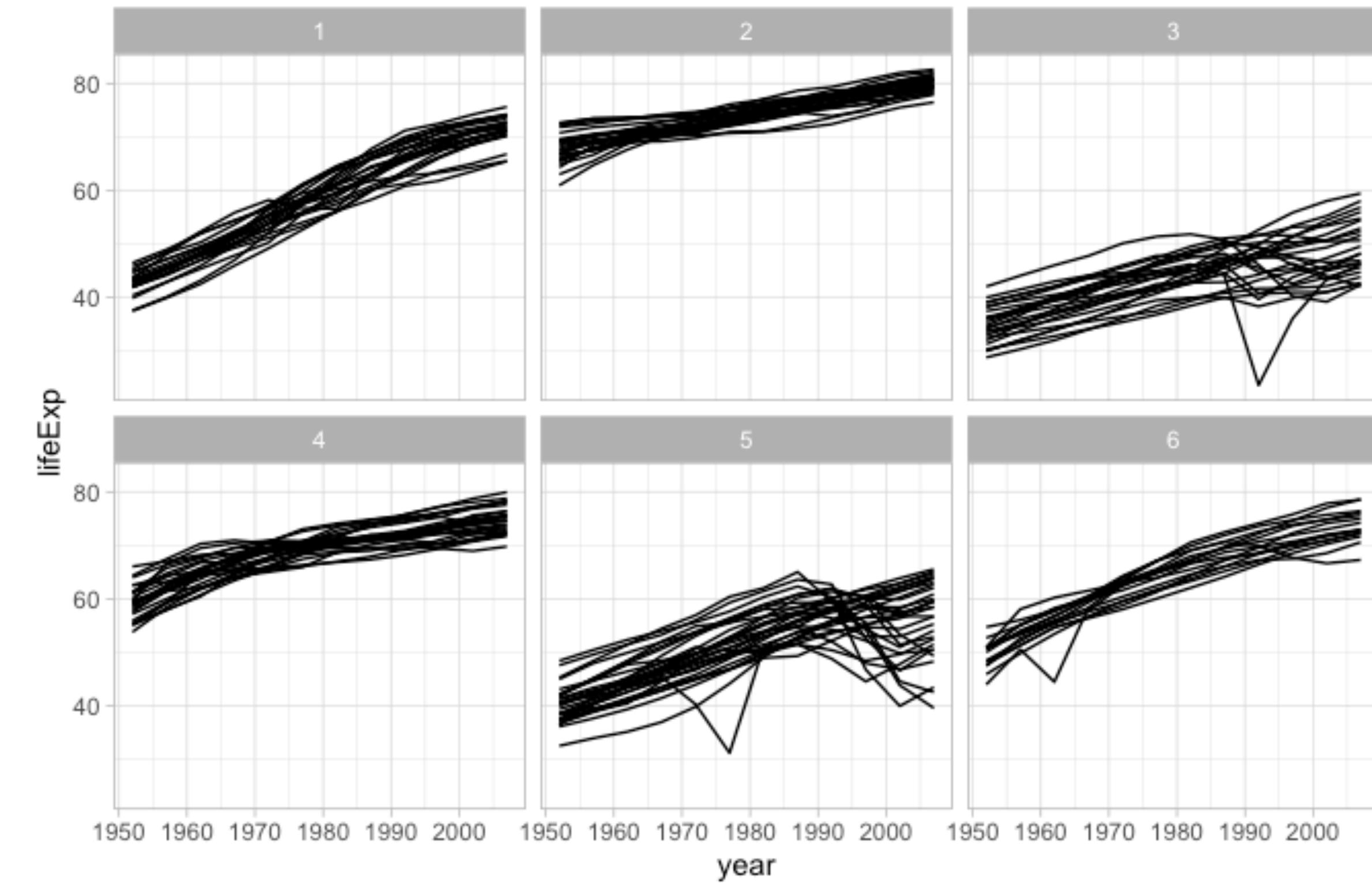
```
gapminder %>%  
  widely_kmeans(country, year, lifeExp, k = 6)
```

```
# A tibble: 142 x 2  
  country     cluster  
  <fct>       <fct>  
  1 Algeria    1  
  2 Egypt      1  
  3 El Salvador 1  
  4 Guatemala  1  
  5 Honduras   1  
  6 Indonesia  1  
  7 Iran       1  
  8 Jordan     1  
  9 Libya      1  
 10 Mongolia   1  
 # ... with 132 more rows
```

# widely\_kmeans performs clustering on tidy data

```
clusters <- gapminder %>%
  widely_kmeans(country, year, lifeExp, k = 6)

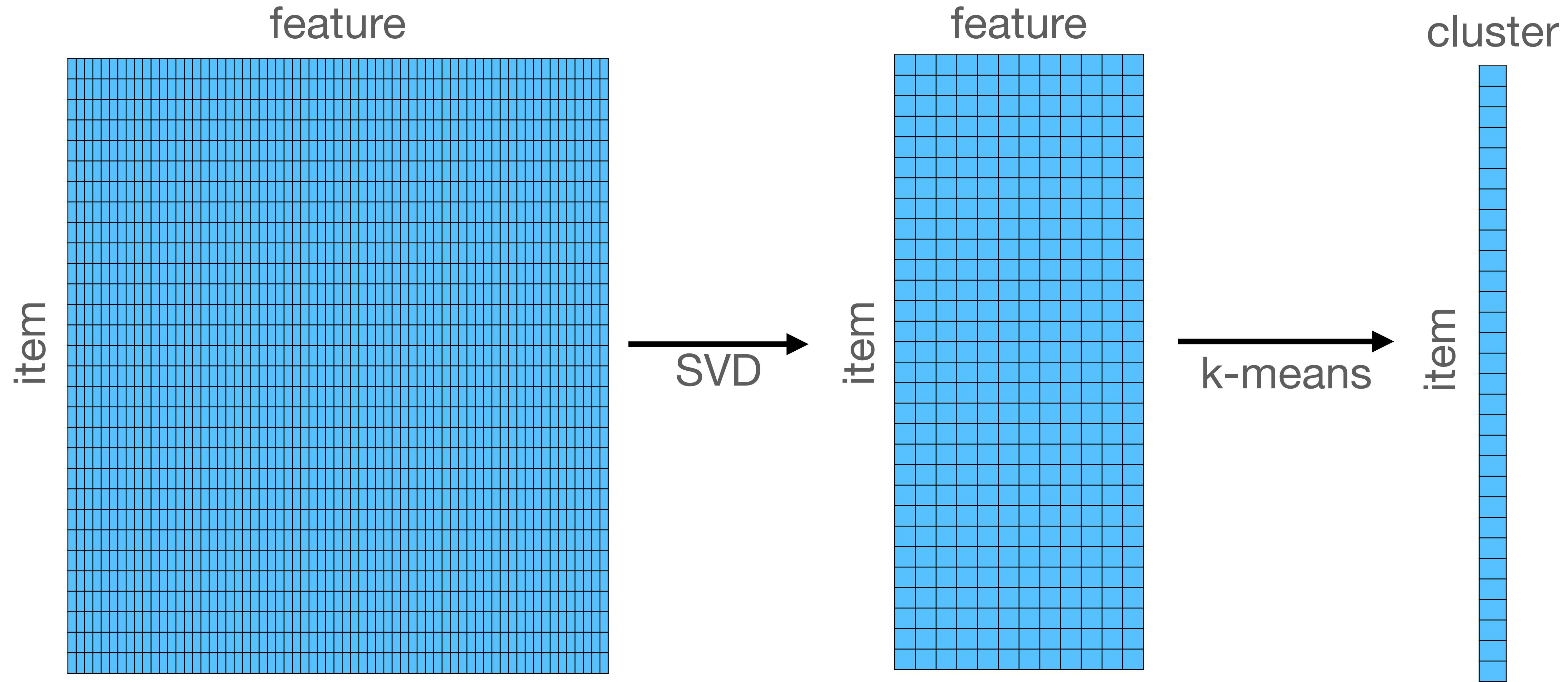
gapminder %>%
  inner_join(clusters, by = "country") %>%
  ggplot(aes(year, lifeExp, group = country)) +
  geom_line() +
  facet_wrap(~ cluster)
```



# widyr (development) offers three `widely_` functions

- `widely_kmeans` K-means clustering
- `widely_hclust` Hierarchical clustering on distances
- `widely_svd` Singular value decomposition for dimensionality reduction

# Dimensionality reduction + clustering



# Dimensionality reduction + clustering

```
# A tibble: 733,404 x 4
  rcid country
  <int> <chr>
1 3 United States of America
2 3 Canada
3 3 Cuba
4 3 Haiti
5 3 Dominican Republic
6 3 Mexico
7 3 Guatemala
8 3 Honduras
9 3 El Salvador
10 3 Nicaragua
# ... with 733,394 more rows
```

```
votes %>%
  widely_svd(country, rcid, vote, nv = 16) %>%
  widely_kmeans(country, dimension, value, k = 6)
```

	country_code	vote
1	US	1
2	CA	-1
3	CU	1
4	HT	1
5	DO	1
6	MX	1
7	GT	1
8	HN	1
9	SV	1
10	NI	1



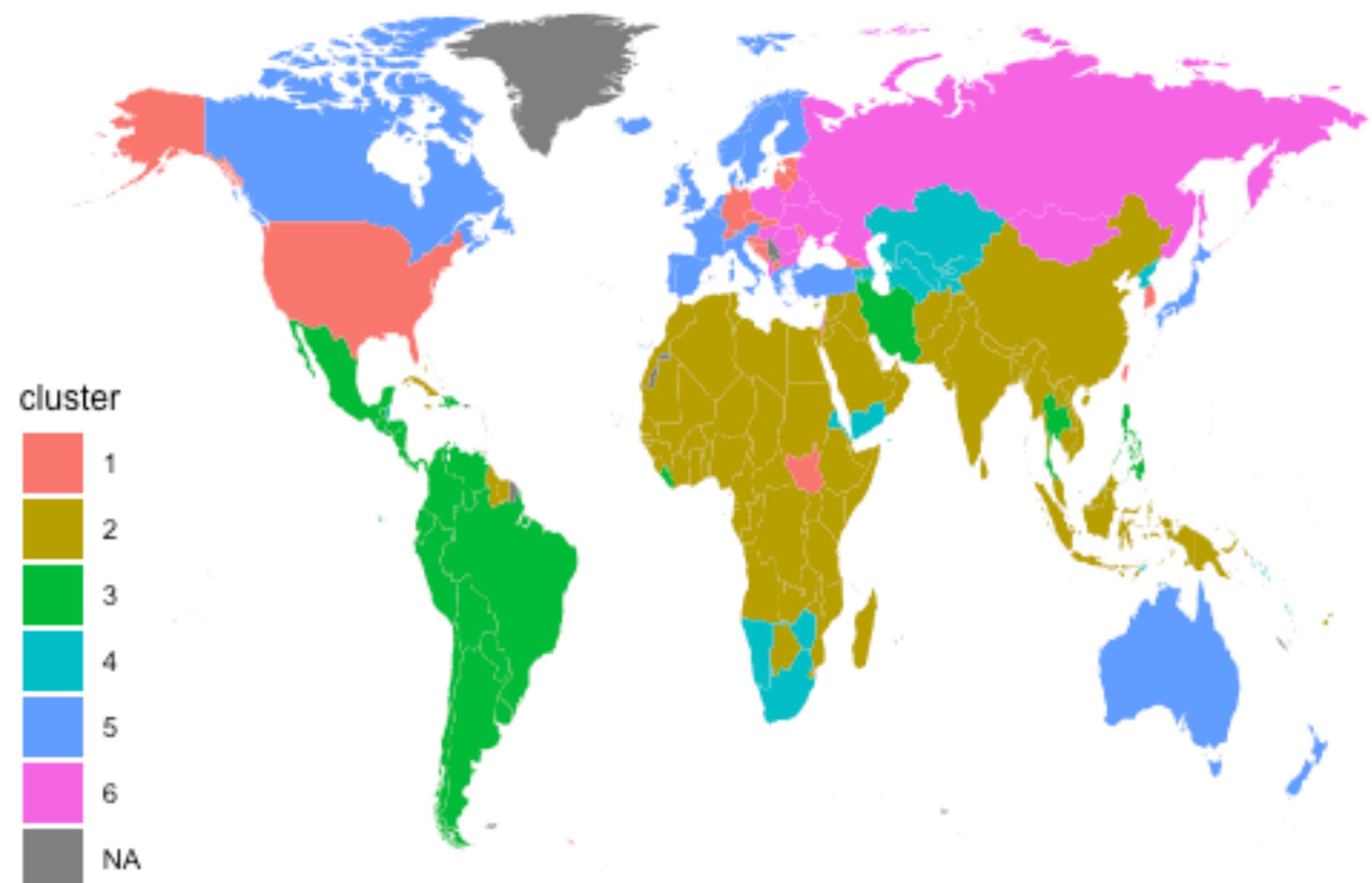
	country	cluster
1	Algeria	1
2	Bahrain	1
3	Barbados	1
4	Bhutan	1
5	Botswana	1
6	Burundi	1
7	China	1
8	Equatorial Guinea	1
9	Fiji	1
10	Gambia	1
# ... with 187 more rows		

# Describing voting blocs through clustering

```
library(maps)
library(fuzzyjoin)

map_clusters <- votes %>%
  widely_svd(country_code, rcid, vote, nv = 24) %>%
  widely_kmeans(country_code, dimension, value, k = 6) %>%
  inner_join(iso3166, by = c(country_code = "a2"))

map_data("world") %>%
  filter(region != "Antarctica") %>%
  regex_left_join(map_clusters, by = c("region" = "mapname")) %>%
  ggplot(aes(long, lat, group = group, fill = cluster)) +
  geom_polygon() +
  ggthemes::theme_map()
```

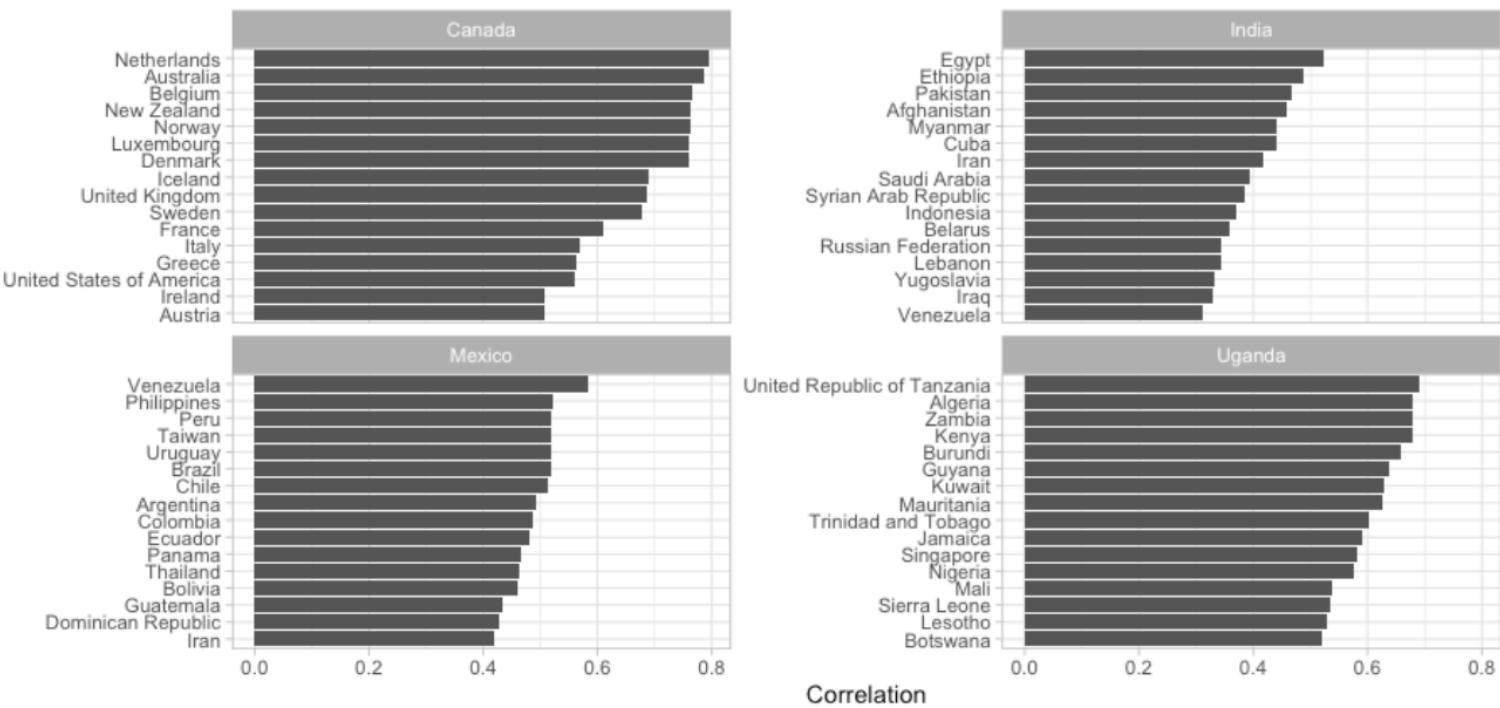


# Conclusion

**“No matter how complex and polished the individual operations are, it is often the quality of the glue that most directly determines the power of the system.”**

**-Hal Abelson**

```
Votes %>%  
  pairwise_cor(country, rcid, vote, sort = TRUE) %>%  
  filter(item1 %in% c("Uganda", "India", "Canada", "Mexico")) %>%  
  group_by(item1) %>%  
  top_n(16, abs(correlation)) %>%  
  mutate(item2 = reorder_within(item2, correlation, item1)) %>%  
  ggplot(aes(correlation, item2)) +  
  geom_col() +  
  facet_wrap(~ item1, scales = "free_y") +  
  scale_y_reordered() +  
  labs(x = "Correlation", y = "")
```



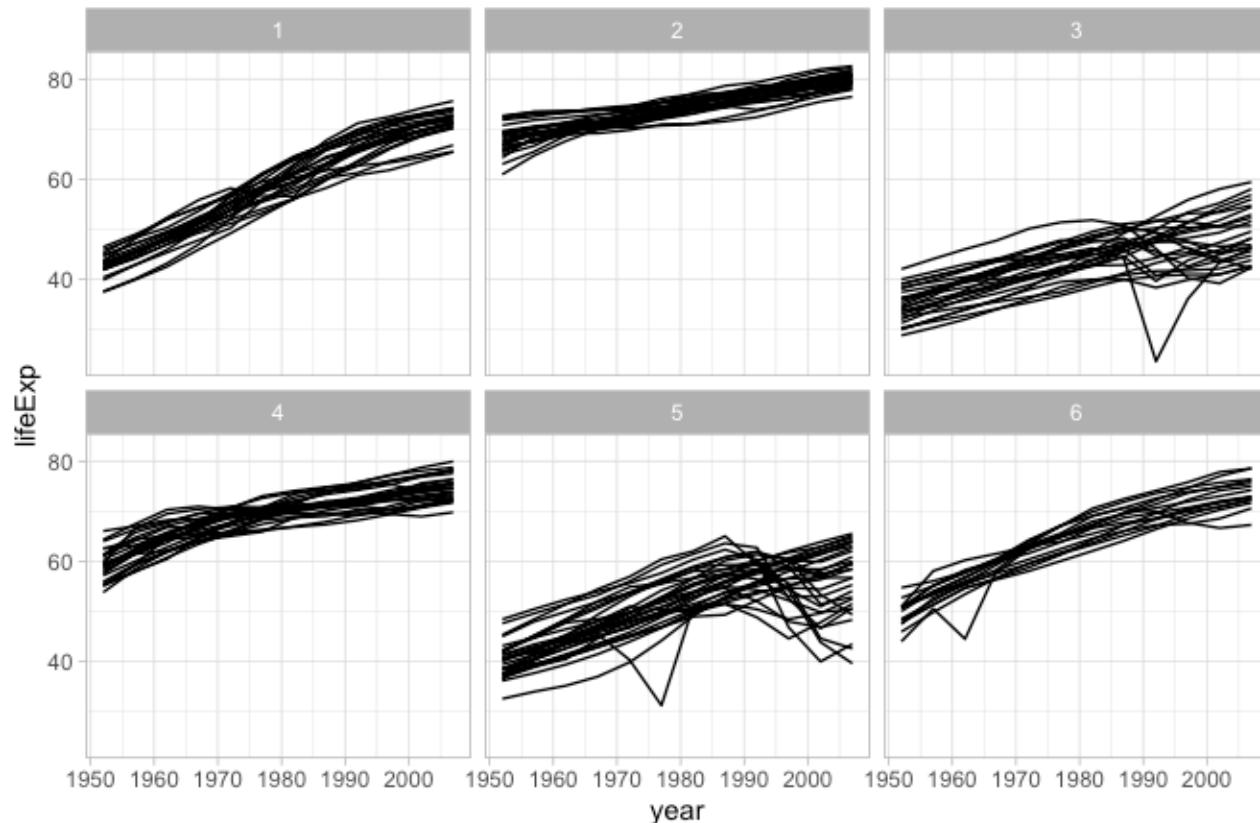
```
library(ggraph)
library(tidygraph)

word_counts <- hacker_news_words %>%
  count(word, sort = TRUE)

hacker_news_words %>%
  pairwise_cor(word, post_id, sort = TRUE) %>%
  head(300) %>%
  as_tbl_graph() %>%
  inner_join(word_counts, by = c(name = "word")) %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation)) +
  geom_node_point(aes(size = n)) +
  geom_node_text(aes(label = name), check_overlap = TRUE,
                vjust = 1, hjust = 1, size = 3) +
  theme(legend.position = "none")
```

```
clusters <- gapminder %>%
  widely_kmeans(country, year, lifeExp, k = 6)

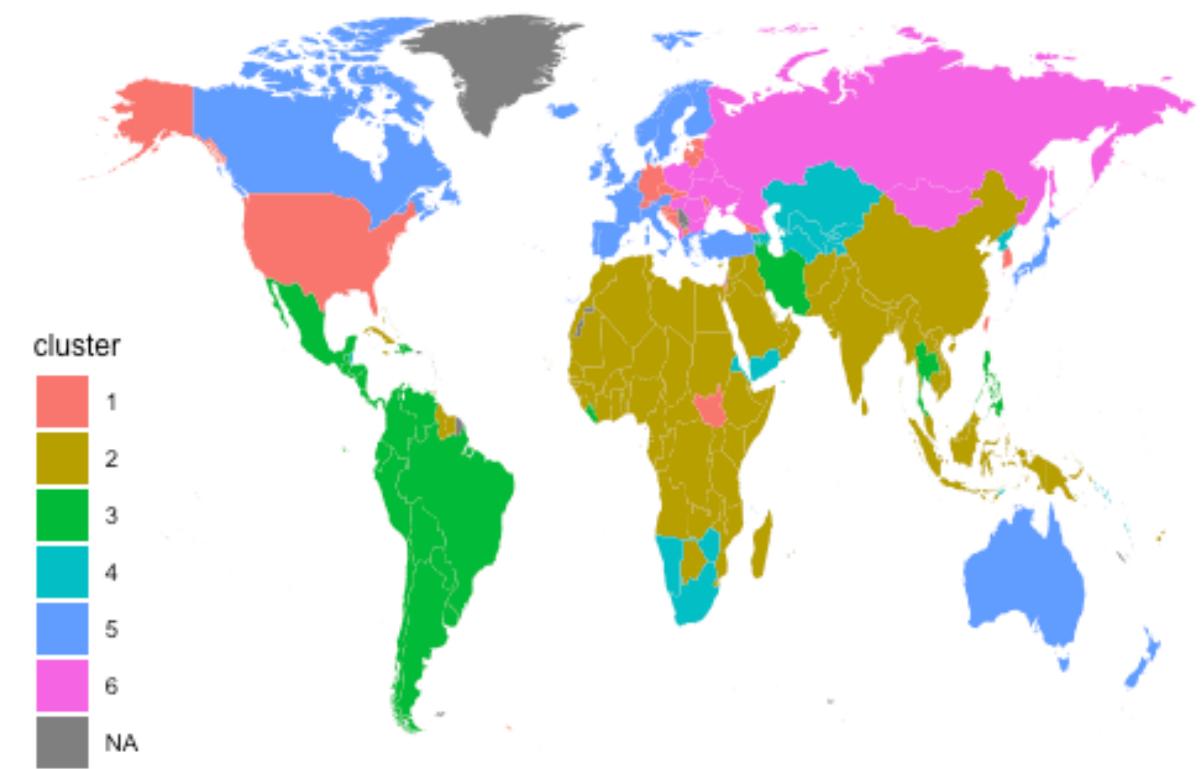
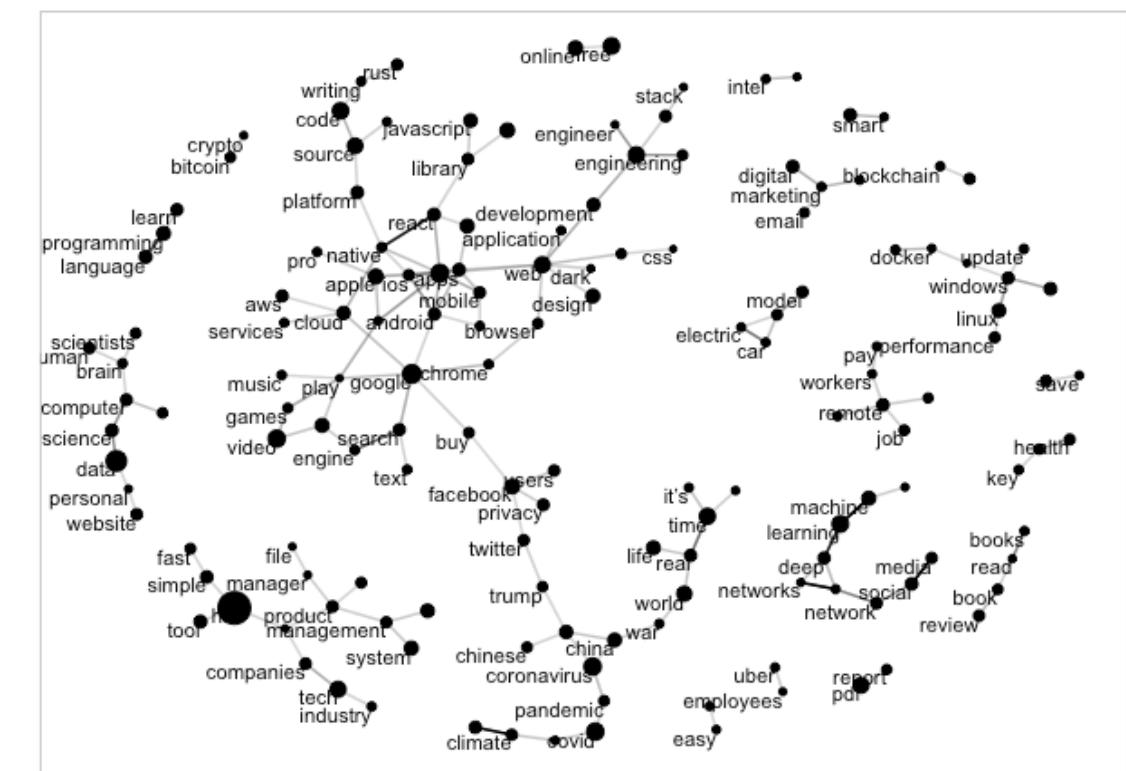
gapminder %>%
  inner_join(clusters, by = "country") %>%
  ggplot(aes(year, lifeExp, group = country)) +
  geom_line() +
  facet_wrap(~ cluster)
```



```
library(maps)
library(fuzzyjoin)

map_clusters <- votes %>%
  widely_svd(country_code, rcid, vote, nv = 24) %>%
  widely_kmeans(country_code, dimension, value, k = 6) %>%
  inner_join(iso3166, by = c(country_code = "a2"))

map_data("world") %>%
  filter(region != "Antarctica") %>%
  regex_left_join(map_clusters, by = c("region" = "mapname")) %>%
  ggplot(aes(long, lat, group = group, fill = cluster)) +
  geom_polygon() +
  ggthemes::theme_map()
```



Once “wide” operations are atomic actions, you can do a lot with a little code

# Thank you

- Lander Analytics
- Jared Lander
- Amada Echeverria



[@drob](http://www.varianceexplained.org)

[www.varianceexplained.org](http://www.varianceexplained.org)

VARIANCE EXPLAINED      ABOUT ME    POSTS    LEARN R    TEXT MINING IN R    INTRODUCTION TO EMPIRICAL BAYES

This is the homepage and blog of David Robinson, Chief Data Scientist at DataCamp. For more about me, [see here](#).

  
**David Robinson**  
Chief Data Scientist at DataCamp, works in R and Python.  
[Email](#)   [Twitter](#)   [Github](#)   [Stack Overflow](#)

**Recent Posts**

---

**The 'knight on an infinite chessboard' puzzle: efficient simulation in R** *December 10, 2018*  
A simulation of a probabilistic puzzle from the Riddler column on FiveThirtyEight.

**Exploring college major and income: a live data analysis in R** *October 16, 2018*  
A live screencast of an exploratory data analysis from the Tidy Tuesday series. This one explores college major and income data from 538.

**Who wrote the anti-Trump New York Times op-ed? Using tidytext to find document similarity** *September 06, 2018*  
An analysis of an anonymous op-ed in the New York Times, using document similarity metrics to match it to Twitter accounts.

**Scientific debt** *May 10, 2018*  
Introducing an analogy to 'technical debt' for data scientists.

Subscribe  
Your email