Data Analytics
CS301
Relational Data

Fall 2018

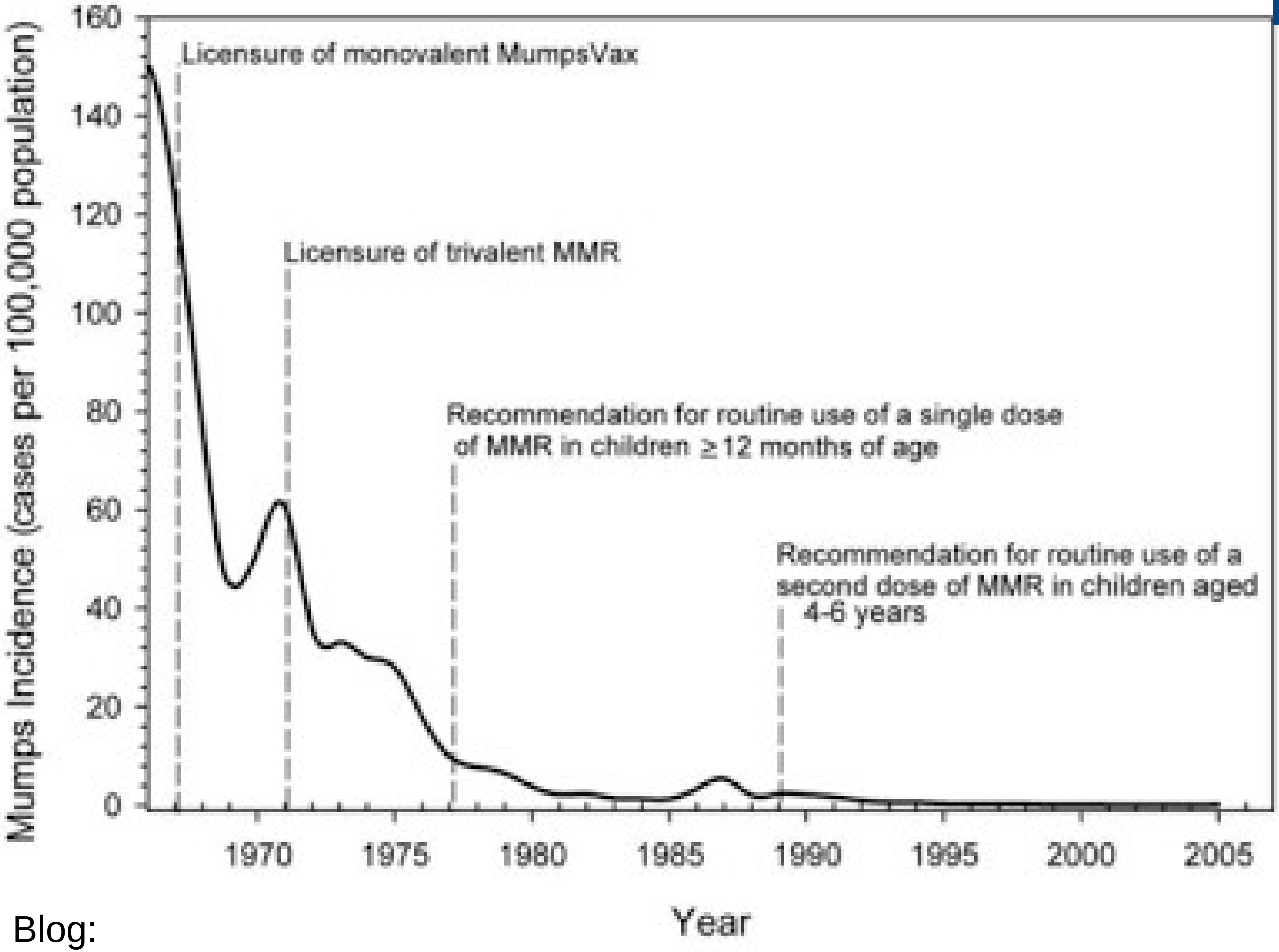
Oliver Bonham-Carter

Let's Talk About Lab 4 For A Moment...

- How do you know if something to prevent sickness is working?
- Are the Vaccines working?
  - Are there fewer people with Measles, mumps, Hepatitis B (and other illnesses) as a result of receiving vaccines in 1966?

- History of Vaccines: https://www.historyofvaccines.org/timeline

What Do Others Say About Vaccines?



Blog:
http://ruleof6ix.fieldofscience.com/2011/10/vaccines-can-you-predict-how-well.html

What Do Others Say About Vaccines?

Comparison of 20th Century Annual Morbidity & Current Morbidity

Disease	20th Century Annual Morbidity*	2010 Reported Cases†	% Decrease
Smallpox	29,005	0	100%
Diphtheria	21,053	0	100%
Pertussis	200,752	21,291	89%
Tetanus	580	8	99%
Polio (paralytic)	16,316	0	100%
Measles	530,217	61	>99%
Mumps	162,344	2,528	98%
Rubella	47,745	6	>99%
CRS	152	0	100%
Haemophilus influenzae (<5 years of age)	20,000 (est.)	270 (16 serotype b and 254 unknown serotype)	99%

Sources:

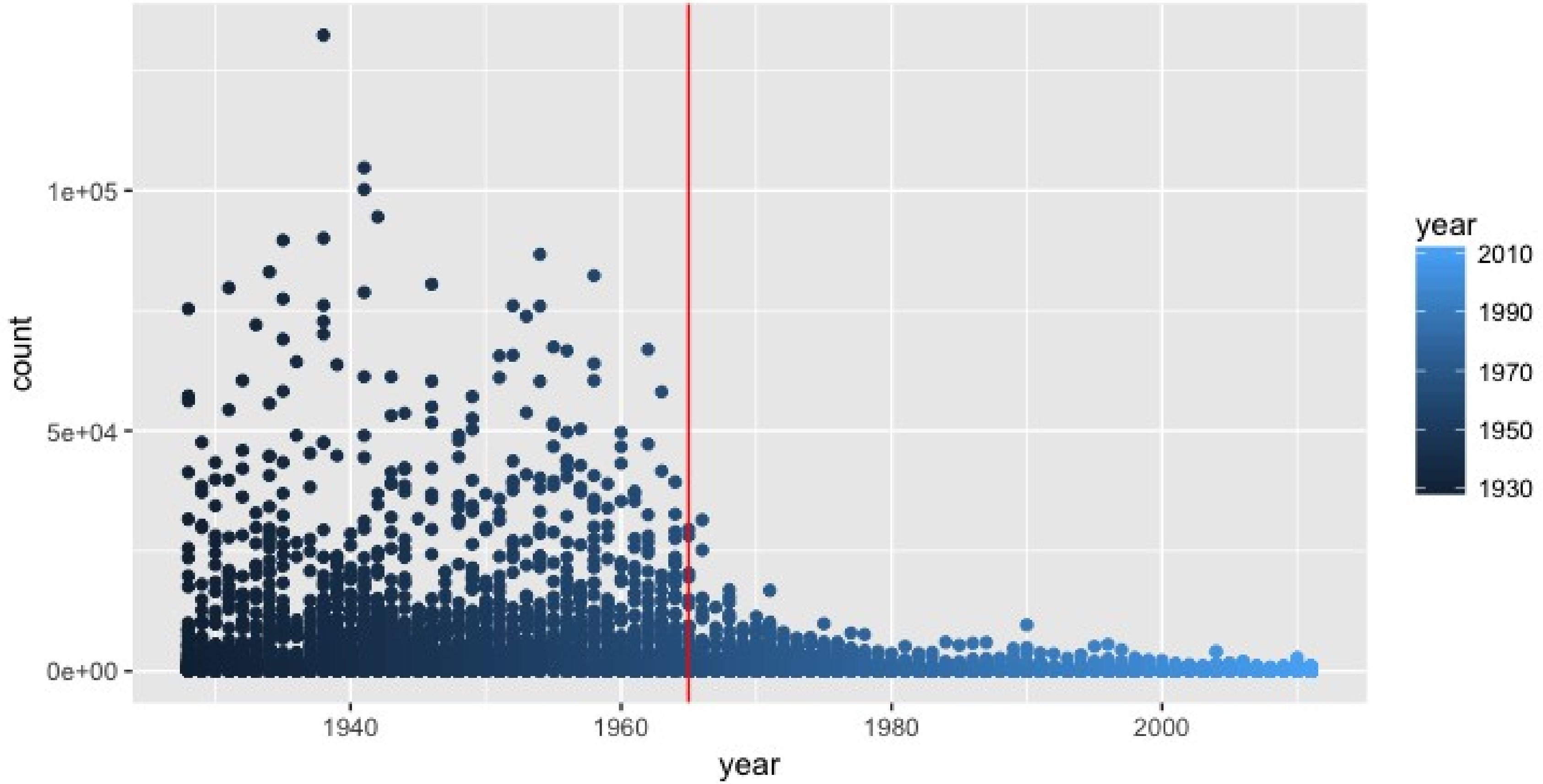
* *JAMA*. 2007;298(18):2155-2163

† CDC. *MMWR* January 7, 2011;59(52);1704-1716. (Provisional *MMWR* week 52 data)

- Vox Article: https://www.vox.com/health-care/2014/10/13/6967317/vaccines-work-this-chart-proves-it

What Does Our Data Say About (All) Vaccines of Data?

```
library(tidyverse)

library(dslabs)

library(dplyr)

ggplot(data = us_contagious_diseases) +  geom_point(mapping = aes(x = year,
y = count, color = year)) + geom_vline(xintercept = 1965, color = "red")
```



Cases of Illness

Lab Results

- #1) Use the us contagious disease and dplyr tools to create an object that **stores only the Measles data**, **includes a per 100,000 people rate**, and removes Alaska and Hawaii. **Note that there is a weeks reporting column. Take that into account when computing the rate.**

```
#Add the rate column to the data:
dat_measles_rate <-
filter(us_contagious_diseases, disease ==
"Measles") %>% mutate(rate = (count/population)
* 100000 * (weeks_reporting/52))

# Note: the rate could be one of several
possible calculations to work with the data.
```

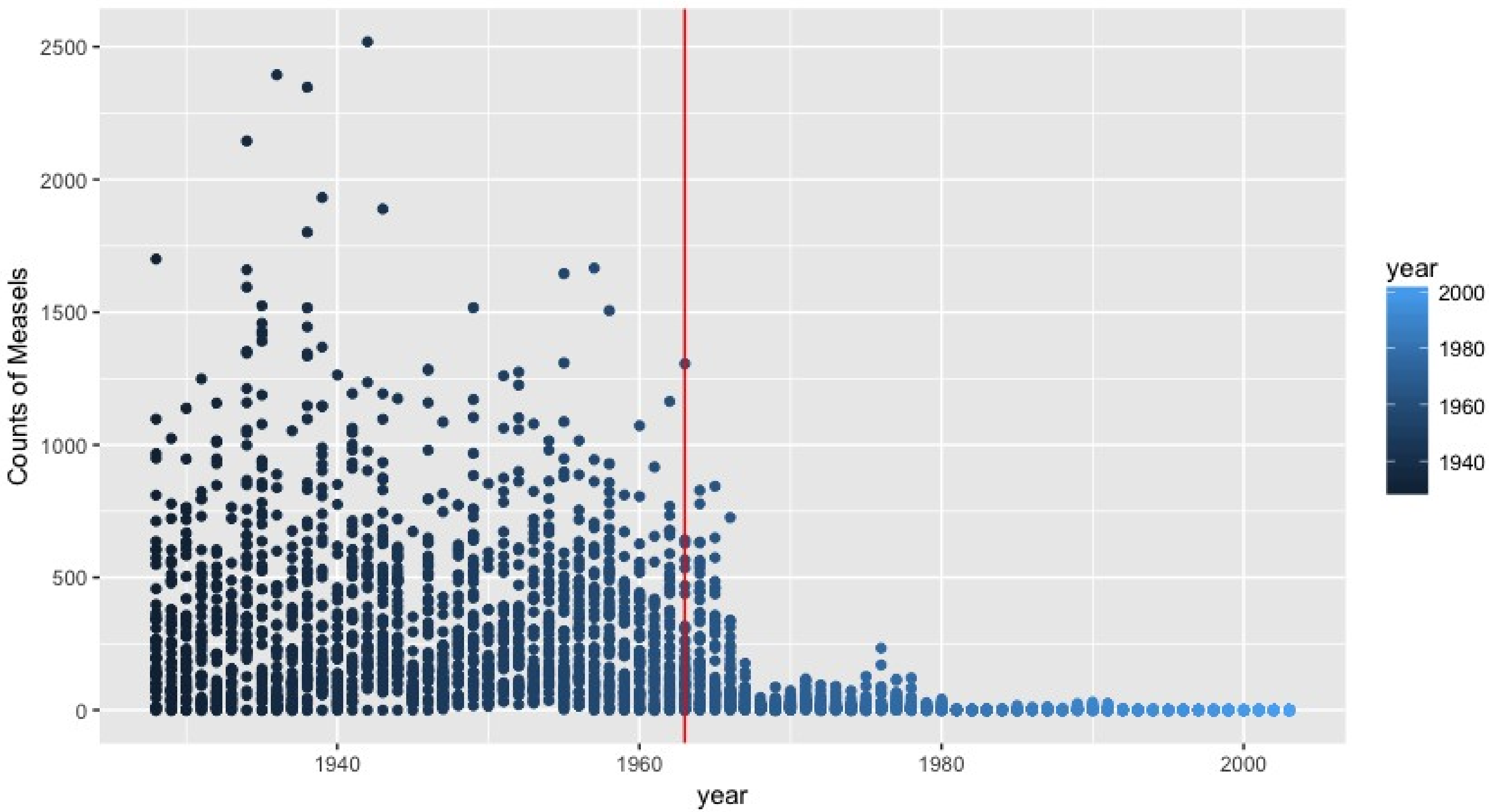
Trim Out Data of Two States: Alaska and Hawaii

```
#Remove the two states (Alaska and Hawaii)

dat_measles_rate_lessTwoStates <-
filter(dat_measles_rate, state != "Alaska",
state != "Hawaii")

View(dat_measles_rate_lessTwoStates)

# Plot the results across 48 states

ggplot(data = dat_measles_rate_lessTwoStates,
mapping = aes(x = year, y = rate, color =
year)) + geom_point() + geom_vline(xintercept =
1963, color = "red") + labs(y = "Counts of
Measels")
```

Plot Across 48 States

```
ggplot(data = dat_measles_rate_lessTwoStates,
mapping = aes(x = year, y = rate, color = year)) +
geom_point() + geom_vline(xintercept = 1963, color
= "red") + labs(y = "Counts of Measels")
```

Code shown
on previous slide



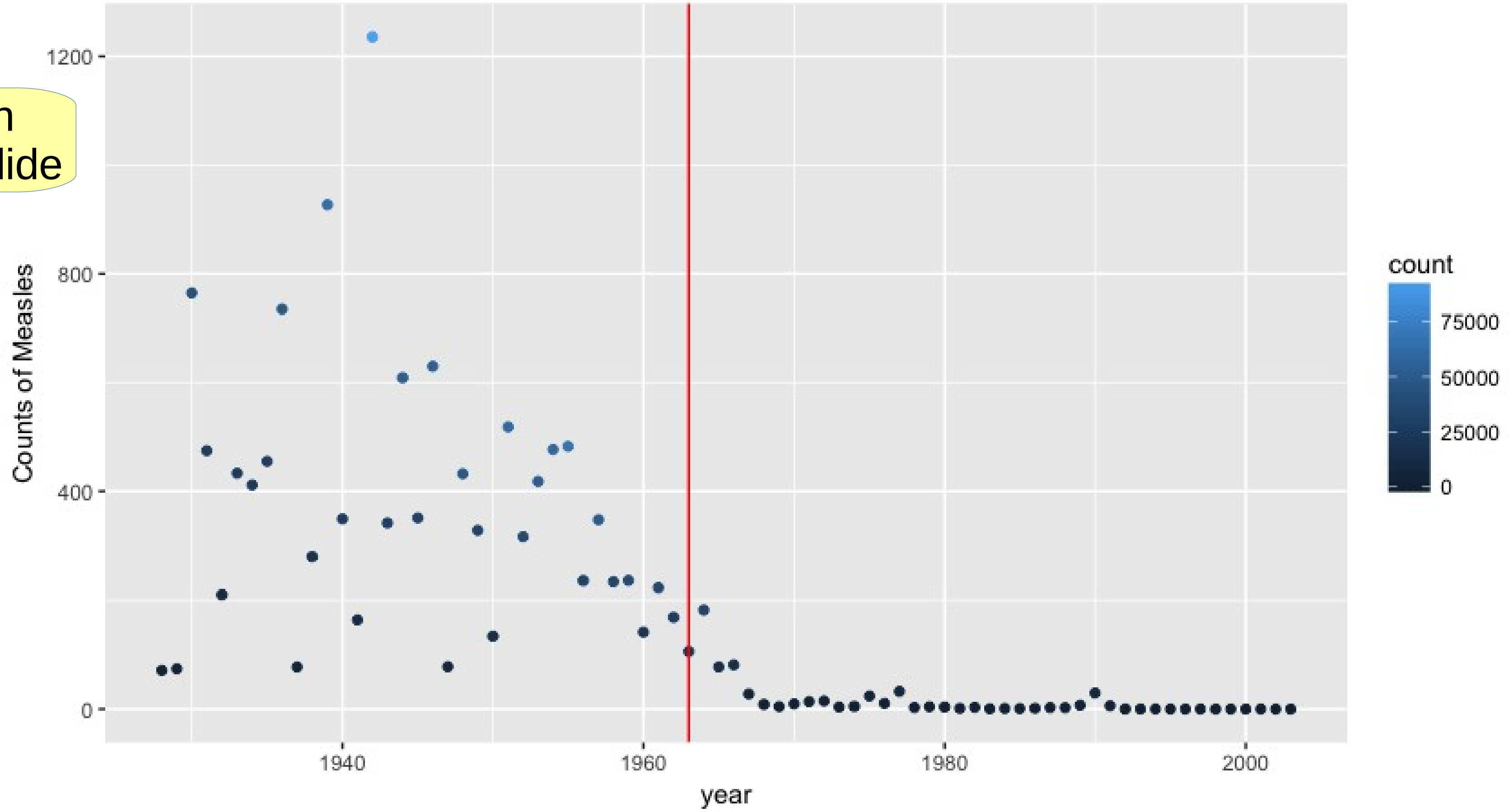
Focus On California

```
# Create table to focus on California

dat_caliFocus <-
filter(dat_measles_rate_lessTwoStates,
state == "California")

View(dat_caliFocus)

ggplot(data = dat_caliFocus, mapping =
aes(x = year, y = rate, color = count)) +
geom_point() + geom_vline(xintercept =
1963, color = "red") + labs(y = "Counts of
Measles")
```

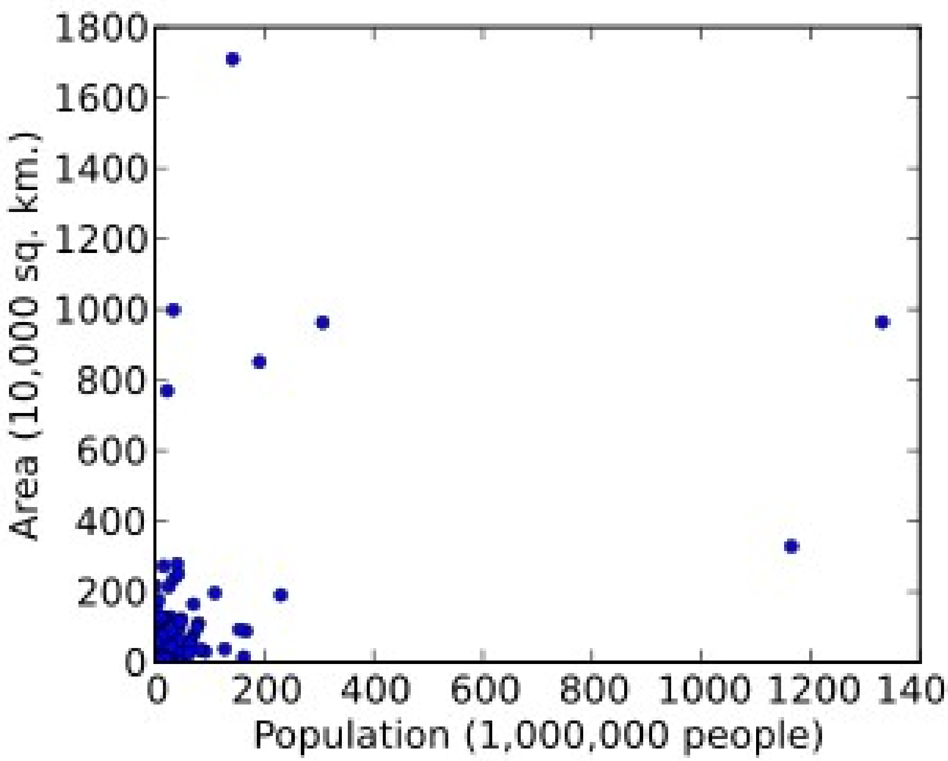
Data From California, Only

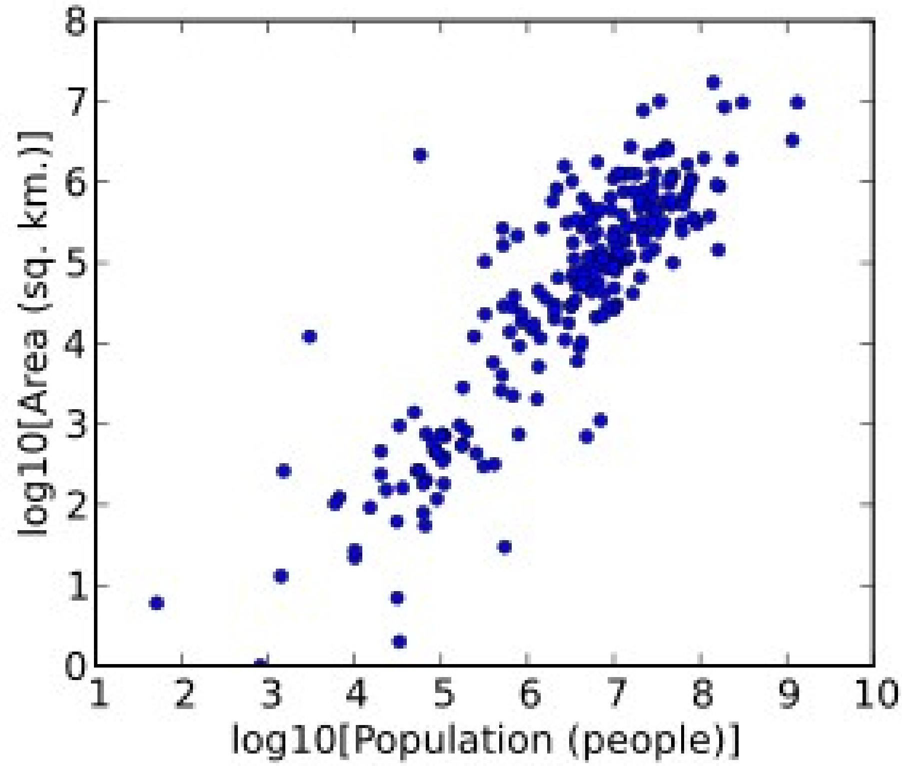
```
ggplot(data = dat_caliFocus, mapping = aes(x = year, y
= rate, color = count)) + geom_point() +
geom_vline(xintercept = 1963, color = "red")
+ labs(y = "Counts of Measles")
```

Code shown
on previous slide



Transformations Help to Fit the Data





Not transformed

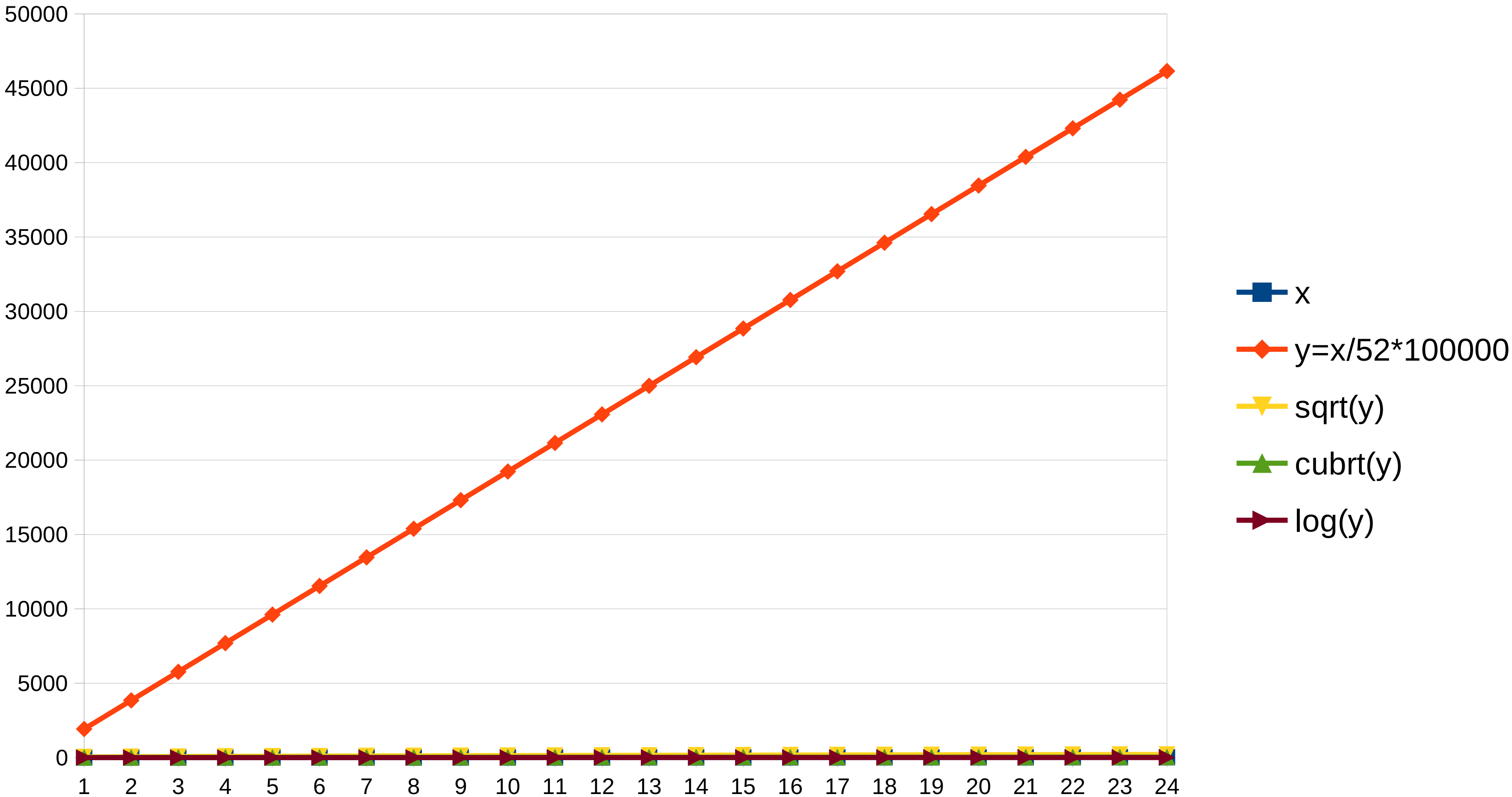
Transformed (using logs)

Transformations Help to Fit the Data

- The square root, ***x to x^(1/2) = sqrt(x)***, is a transformation with a moderate effect on distribution shape.
- This approach is weaker than the logarithm and the cube root transformations in its ability to influence the distribution shape.
- Used for reducing right skewness
- Has the advantage that it can be applied to zero values.
- Commonly applied to counted data, especially if the values are mostly rather small

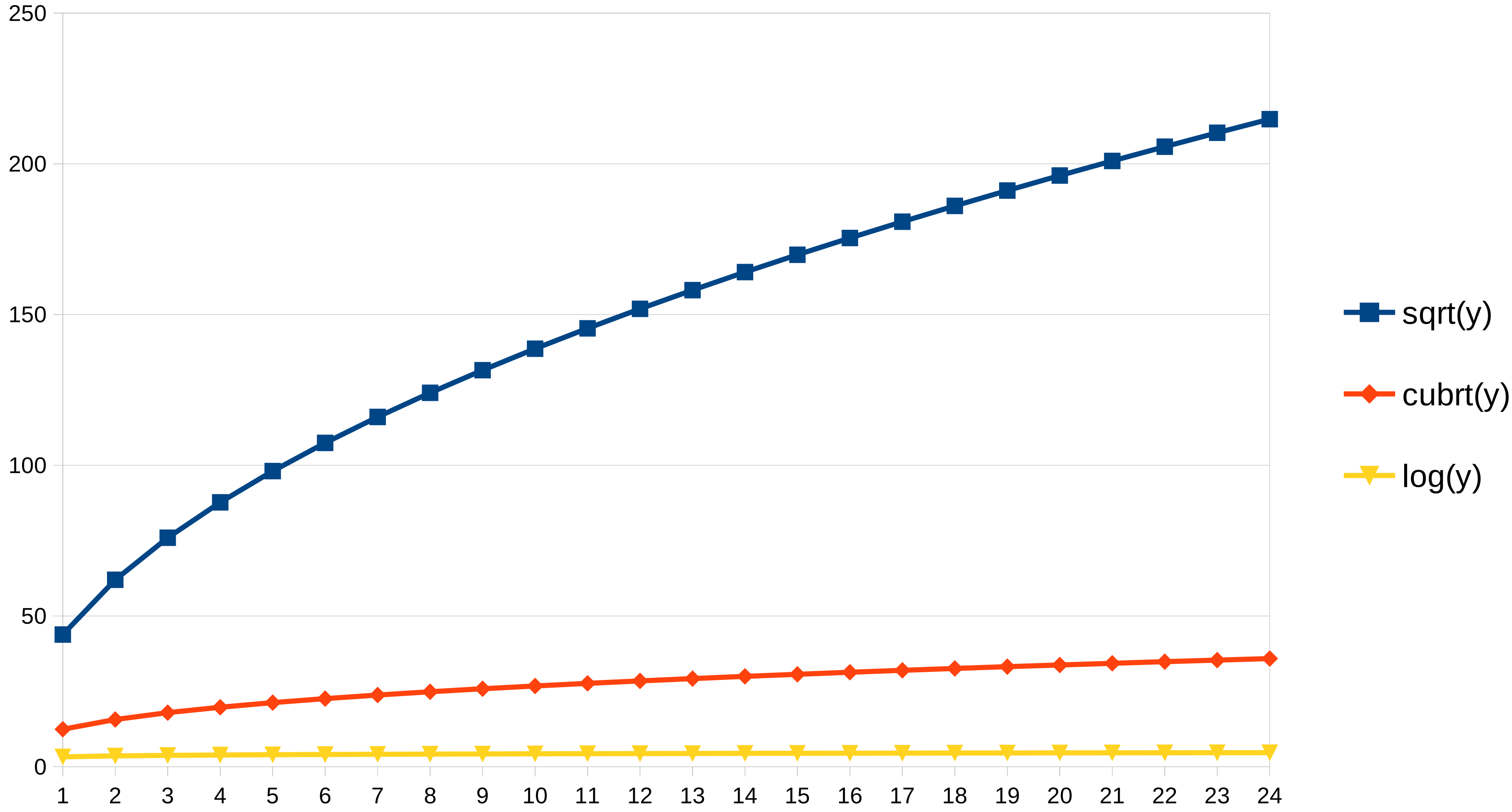
http://fmwww.bc.edu/repec/bocode/t/transint.html

Effects of Transformations on Values



x	y=x/52*10000	sqrt(y)	cubrt(y)	log(y)
1	1923.076923	43.85290097	12.43556587	3.283996656
2	3846.153846	62.01736729	15.6678312	3.585026652
3	5769.230769	75.95545253	17.93518953	3.761117911
4	7692.307692	87.70580193	19.74023034	3.886056648
5	9615.384615	98.05806757	21.26451851	3.982966661
6	11538.46154	107.4172311	22.59692282	4.062147907

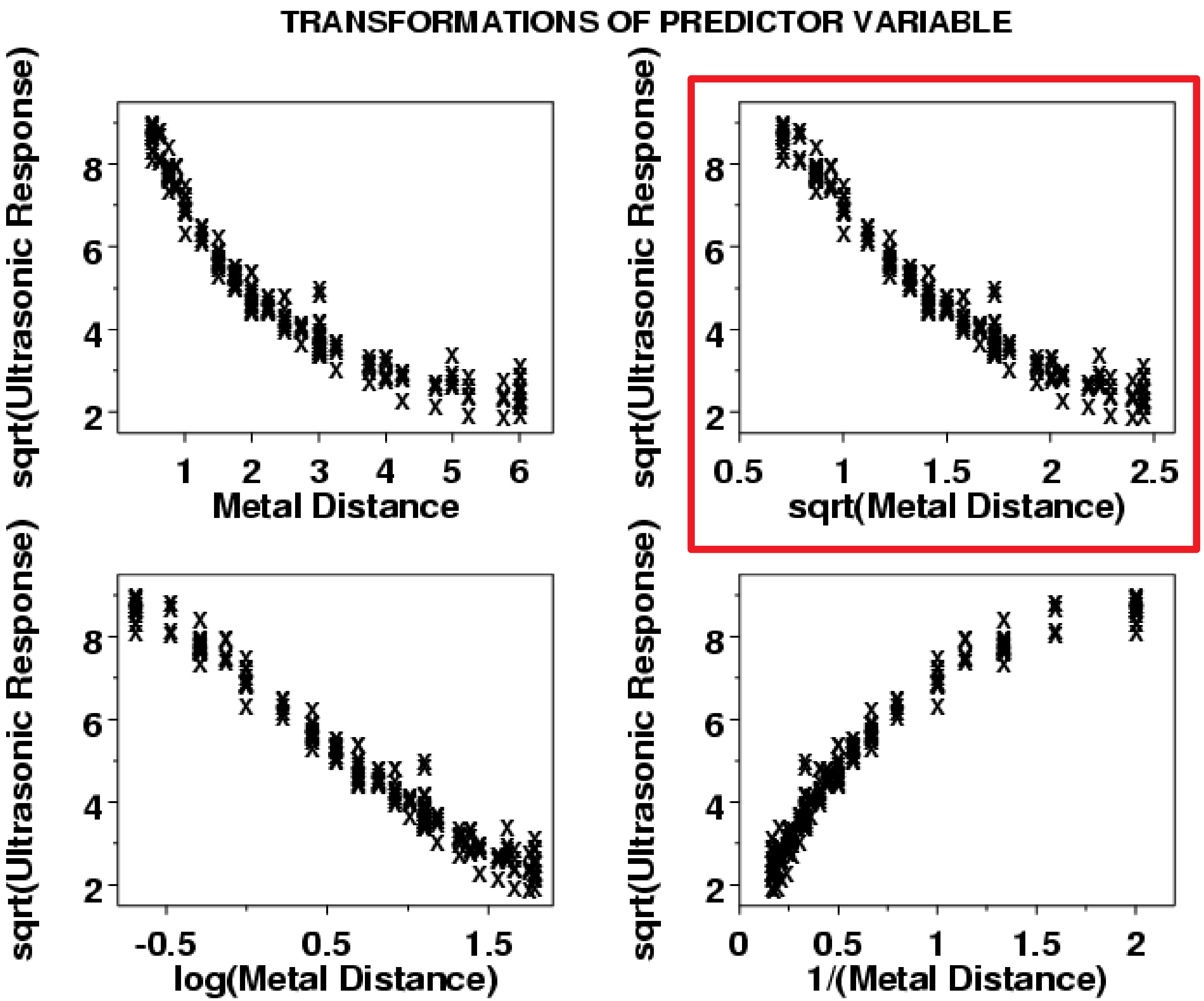
Effects of Transformations on Values Zoom-in



Transformations Help to Fit the Data

- Reduce the Y into a smaller space to see trends.
- Places all points on a similar playing ground
- P <- (x,y)
- Trans(p) <-

  (x, sqrt(y))

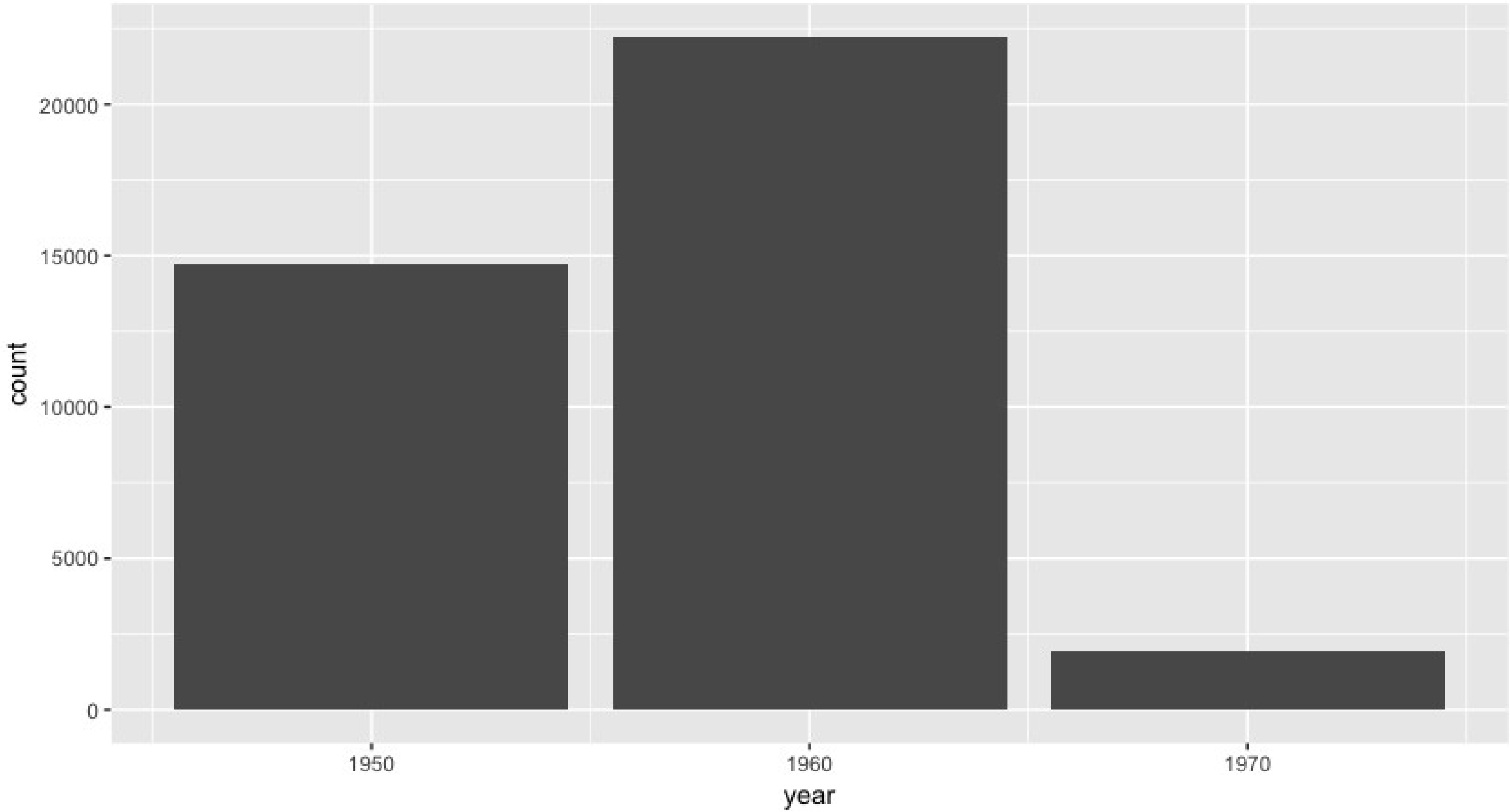


The 1950’s, 1960’s and 1970’s
Without Transformation

```
#plot three bars to see what happened
in the 1950’s, 1960’s and 1970’s.

ggplot(data = dat_caliFocus %>%
filter(year == 1950 | year == 1960 |
year == 1970)) + geom_bar(mapping =
aes(x = year, y = count), stat =
"identity")
```

Back to the vaccines lab...

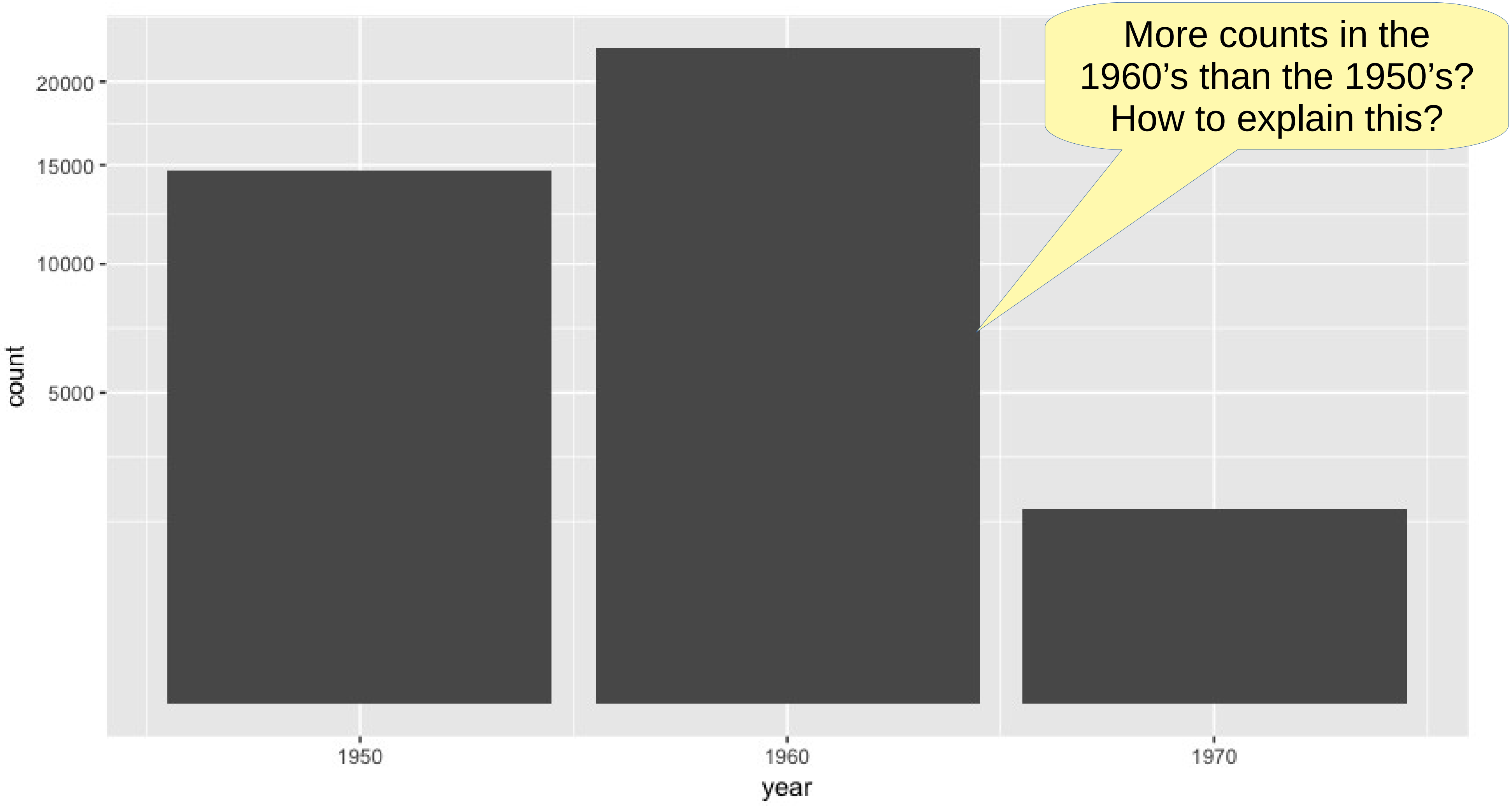
The 1950’s, 1960’s and 1970’s
Without Transformation



The 1950's, 1960's and 1970's
With Sqrt() Transformation

```
#plot three bars to see what happened in the 1950's, 1960's and 1970's.

ggplot(data = dat_caliFocus %>% filter(year == 1950 | year == 1960 | year == 1970)) + geom_bar(mapping = aes(x = year, y = sqrt(count)), stat = "identity")
```

The 1950's, 1960's and 1970's
With Sqrt() Transformation

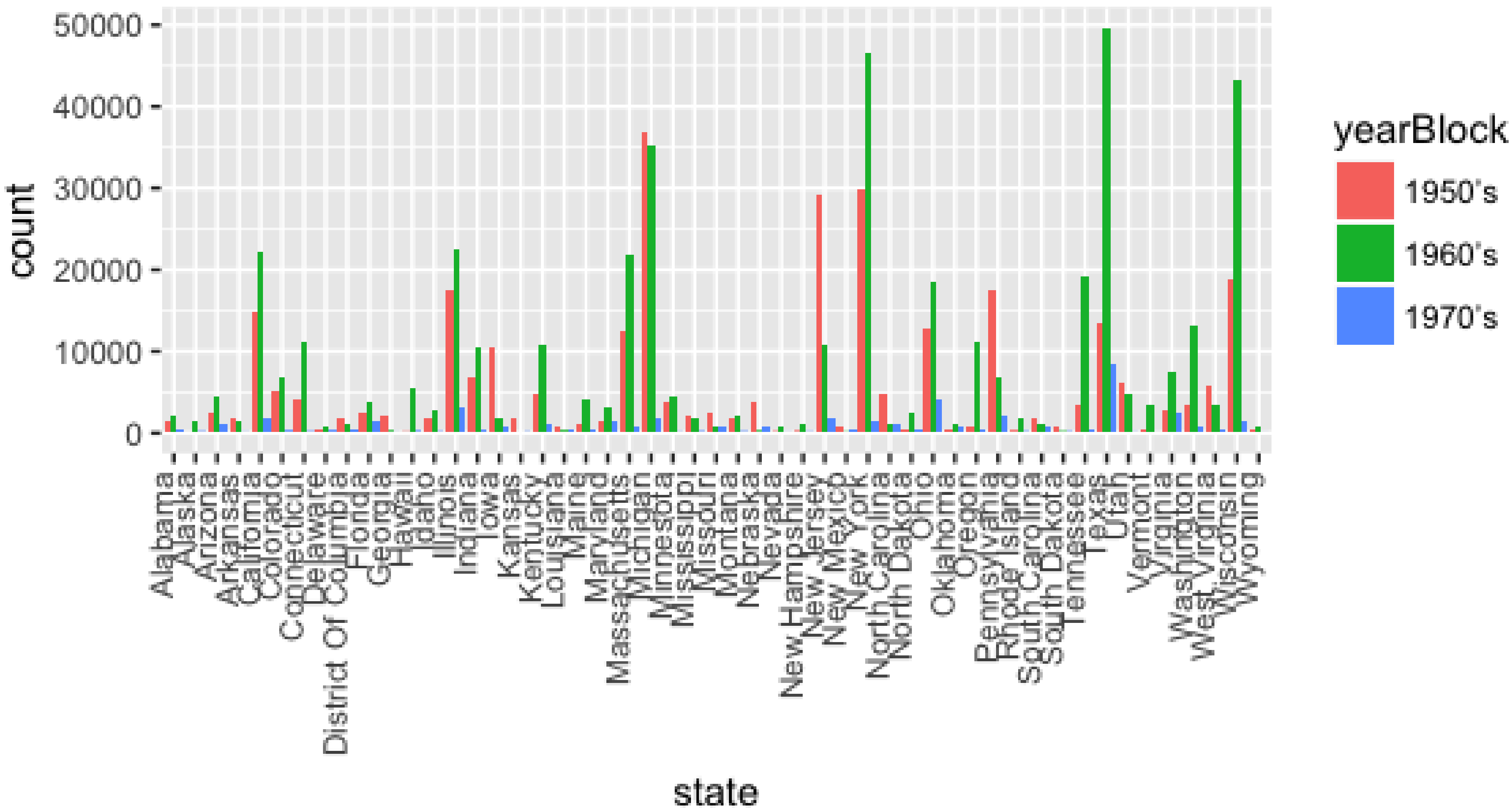


The 1950’s, 1960’s and 1970’s
Without Transformation

```
library(tidyverse)

library(dslabs)

library(dplyr)

dat <- filter(us_contagious_diseases, disease == "Measles") %>% mutate(rate =
(count/population) * 100000 * (weeks_reporting/52))

# Filter out all data except in the years 1950, 1960, and 1970

dat_measles_rate_lessTwoStates <- dat %>% filter(year == 1950 | year == 1960 | year == 1970)

#create some “block”, containers to hold the data for each year.

dat_measles_rate_lessTwoStates$yearBlock[dat_measles_rate_lessTwoStates$year == 1950]
<-"1950’s"

dat_measles_rate_lessTwoStates$yearBlock[dat_measles_rate_lessTwoStates$year == 1960]
<-"1960’s"

dat_measles_rate_lessTwoStates$yearBlock[dat_measles_rate_lessTwoStates$year == 1970]
<-"1970’s"

#Without transformation, Multi-bar per state,

ggplot(data = dat_measles_rate_lessTwoStates) + geom_bar(mapping = aes(x = state, y = count,
fill = yearBlock), position = "dodge", stat = "identity") + theme(axis.text.x =
element_text(angle = 90, hjust = 1, vjust=-0.01))
```

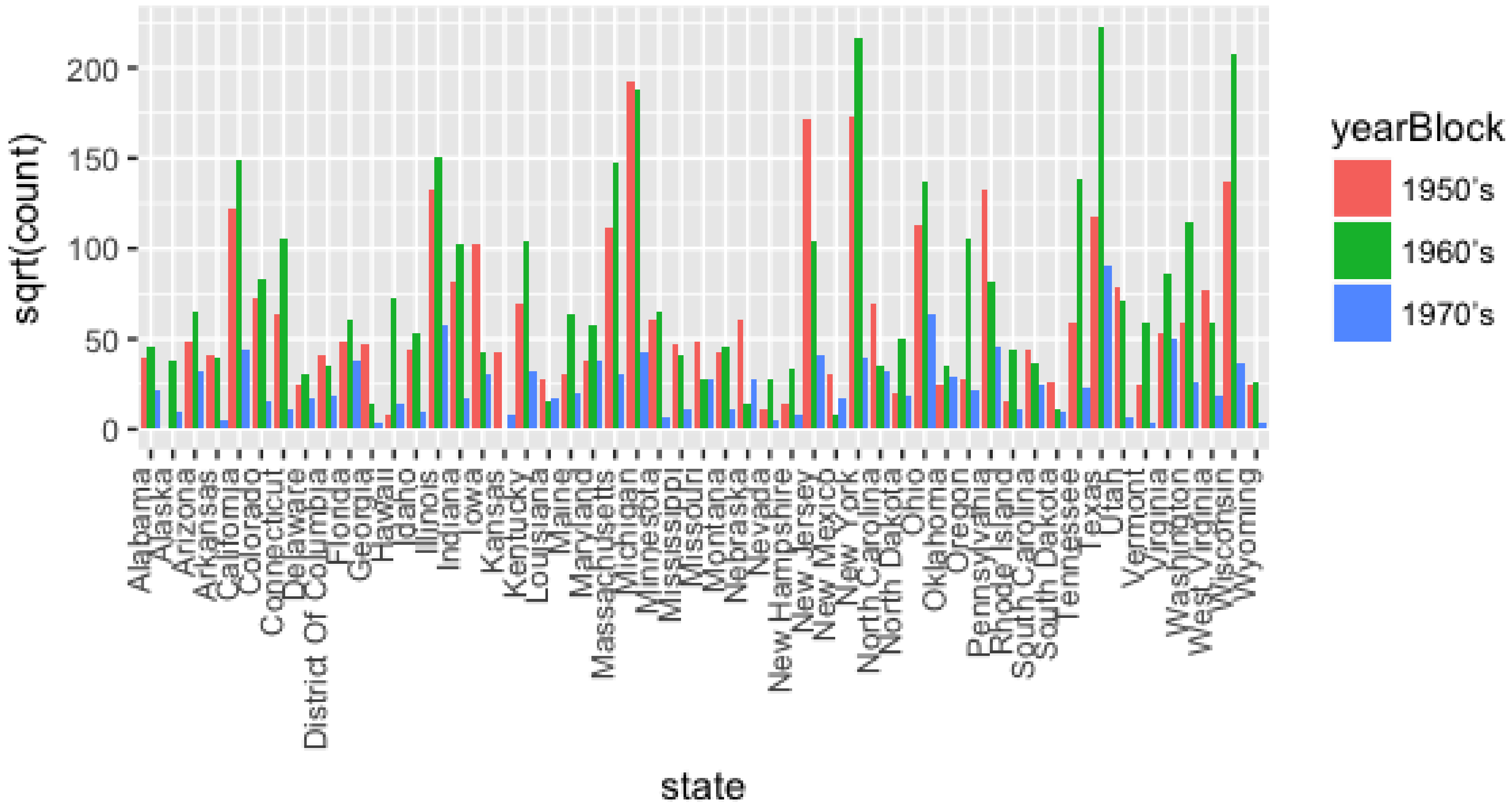
The 1950’s, 1960’s and 1970’s
Without Transformation

```
ggplot(data = dat_measles_rate_lessTwoStates) + geom_bar(mapping = aes(x
= state, y = count, fill = yearBlock), position = "dodge", stat =
"identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1,
vjust=-0.01))
```



The 1950's, 1960's and 1970's
with sqrt() Transformation

```
ggplot(data = dat_measles_rate_lessTwoStates) + geom_bar(mapping = aes(x
= state, y = sqrt(count), fill = yearBlock), position = "dodge", stat =
"identity") + theme(axis.text.x = element_text(angle = 90, hjust = 1,
vjust=-0.01))
```



Urban Versus Rural

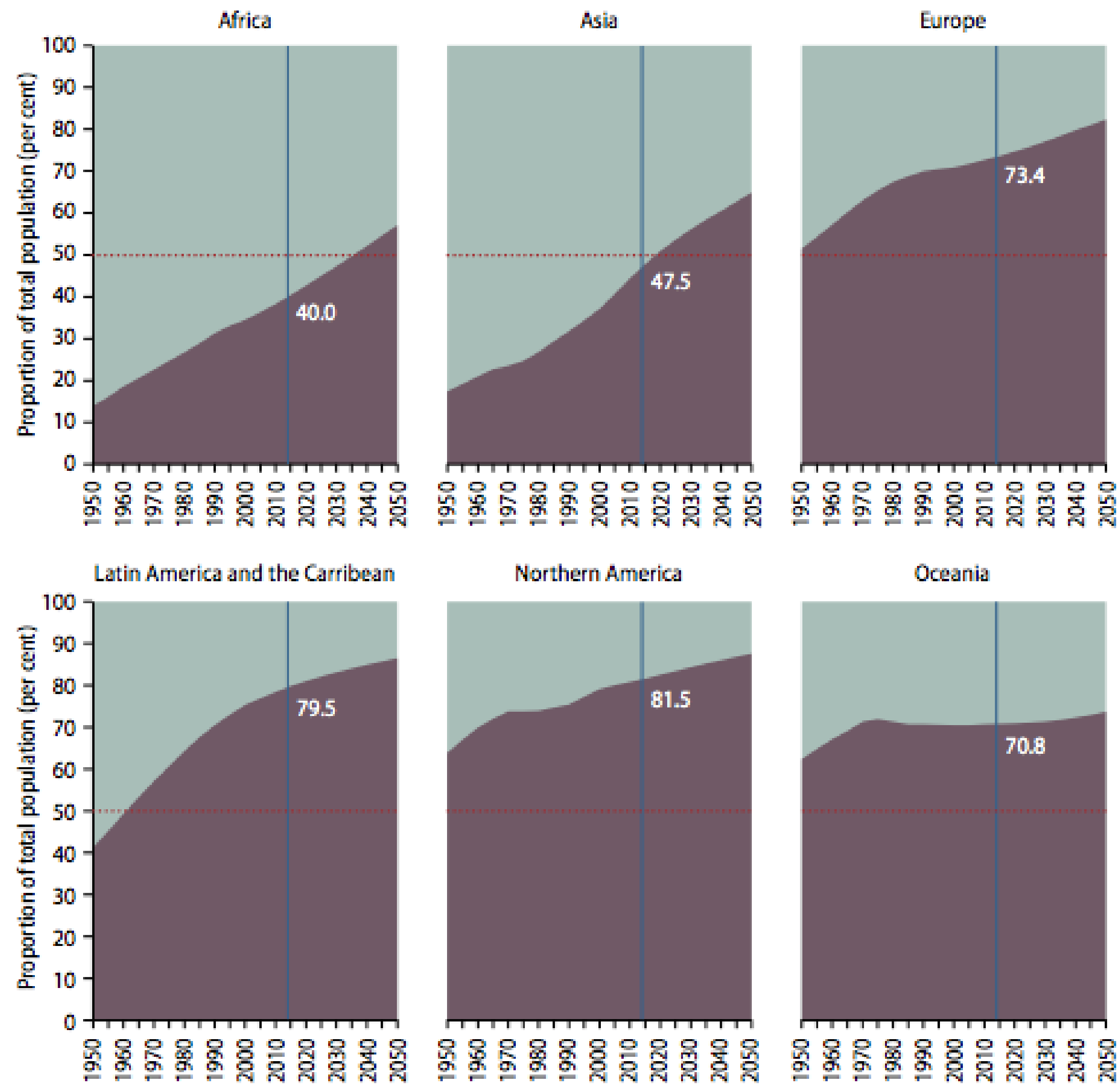
A possible Explanation for the 1950's

Urbanization has occurred in all major areas, yet Africa and Asia remain mostly rural

- **Urban**: City dwelling
- **Rural**: Country dwelling
- Vaccinations in Rural Areas:
  - Were there fewer people available in from whom to contract viruses?
  - Less opportunity to see others in country?
- Country areas: you are breathing your neighbour's breath.

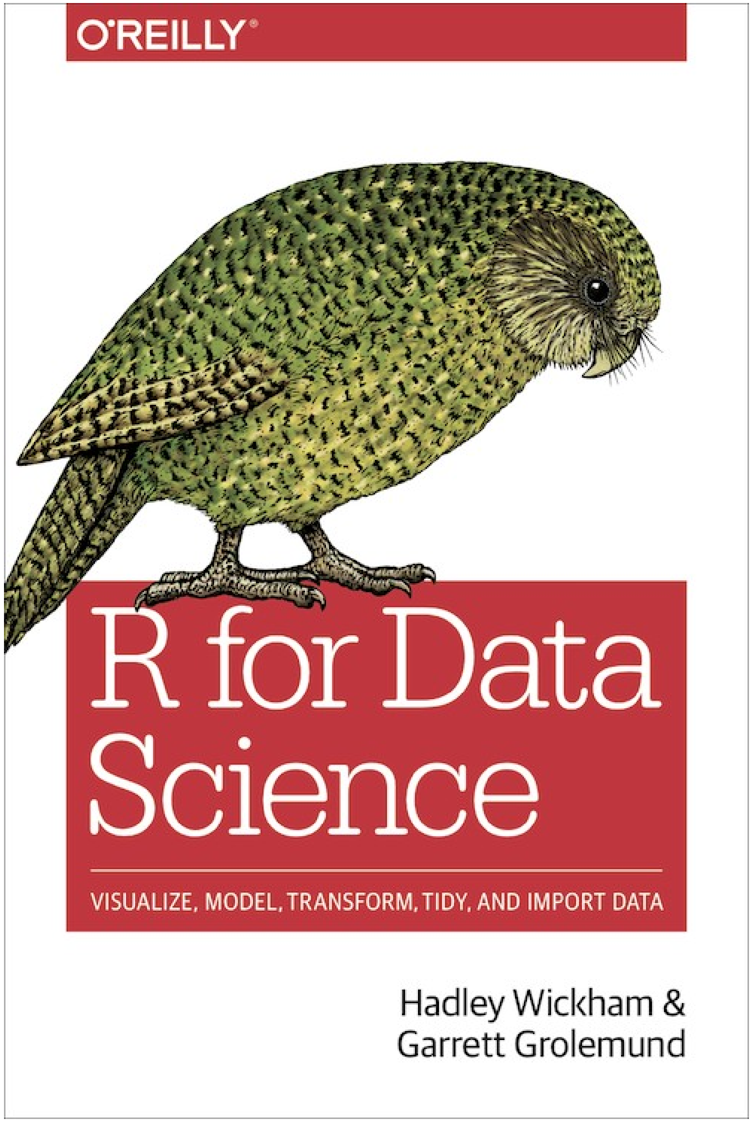
Figure 3.

Urban and rural population as proportion of total population, by major areas, 1950–2050



https://esa.un.org/unpd/wup/publications/files/wup2014-highlights.Pdf

Where in the Web?
Where in the Book?



- Note the chapter differences!
- Book:
  - Chap 10
- Web:
  - Chap 13
- Relational Data

Relational Databases

- A database table is similar to those that we have been using already.

attributes (or columns)

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

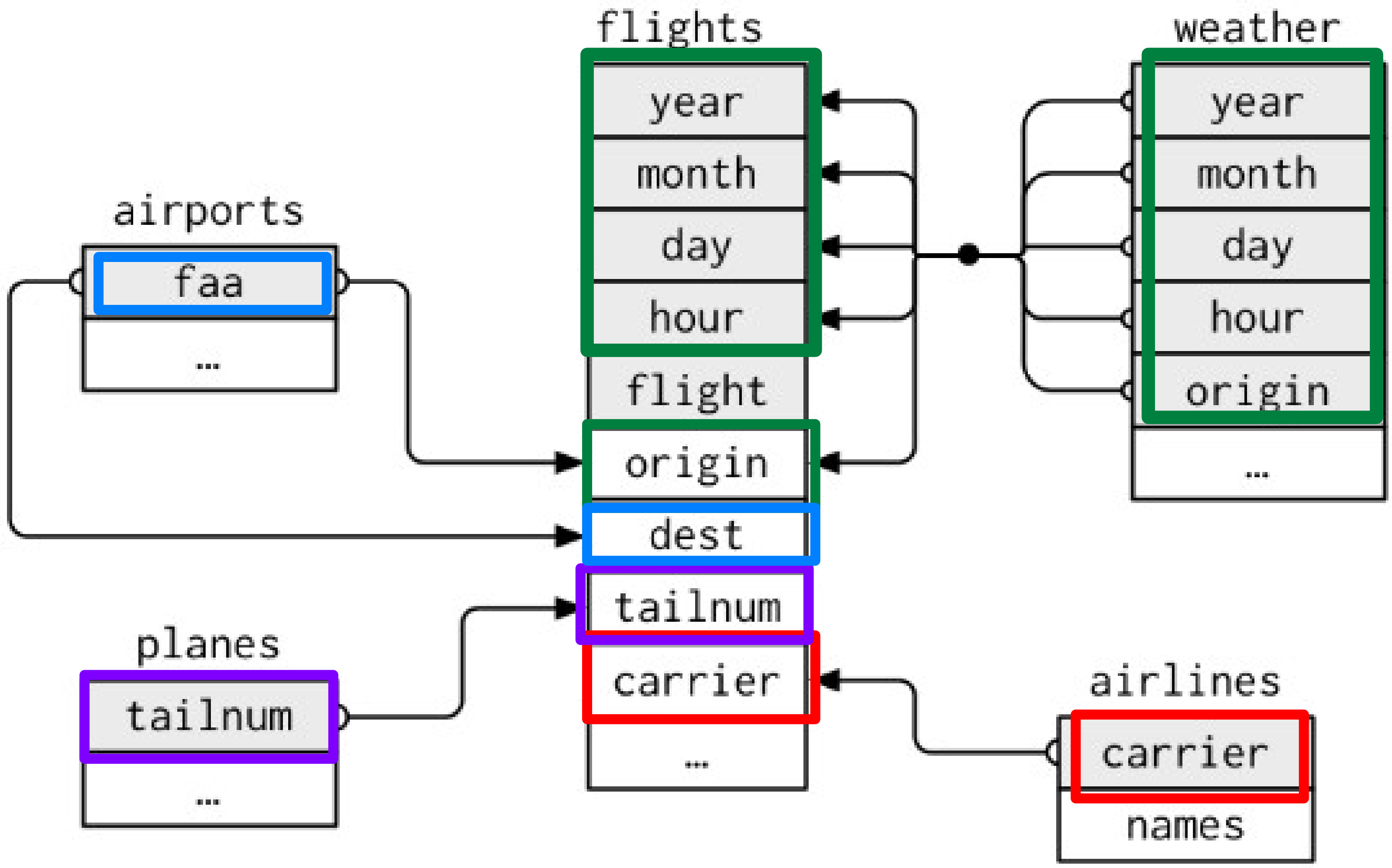
tuples (or rows)

Let's Look at Some Tables

```
library(tidyverse)
library(nycflights13)
#show built-in tables
View(airlines)
View(airports)
View(planes)
View(weather)
```

Relational Databases

- The data of these built-in tables is “connected” in the sty



Relational Databases

- **Primary Keys:** Unique identifier for each row of the table.
  - Ex: planes$tailnum
- **Foreign Keys:** Unique identifier for row in another table.
  - Ex: flights$tailnum
  - Is a foreign key since it exists in the flights table and matches a flight to a unique plane.

Checking For Your Keys

```
# If something is unique: there is only one of it. Here each tailnum entry is unique

planes %>% count(tailnum)

# Try setting up a test to see if there are any more than one of an entry (necessary to be a primary key)

planes %>% count(tailnum) %>% filter(n > 1)

# A key could be a combination of things

weather %>% count(year, month, day, hour, origin) %>%  filter(n > 1)

flights %>% count(year, month, day, flight) %>%  filter(n > 1)
```

Find Some Keys!

- **Baby-name Data**
- First: **install.packages("babynames")**
- **library(babynames) and tidyverse too!**
- Then find the primary keys in,
  babynames:babynames

- **Baseball data:**
- First: **install.packages("Lahman")**
- **library(Lahman)**
- Then find the primary keys in,
  Lahman::Batting

Possible Solutions: Find Some Keys!

- **Baby-name Data**
- `babynames::babynames %>% count(name, year, sex)`
  - `Note: filter the added counter column (nn)`
- `babynames::babynames %>% count(name, year, sex) %>% filter(nn >1)`
- **Baseball data:**
- `Lahman::Batting %>% group_by(playerID, yearID, stint) %>% filter(n() > 1) %>% nrow()`

Babynames Data
On a Side-Note

```
library(stringr)
library(babynames),
View(babynames)
#Want to Select all names beginning with “A”
#Use regular expressions!
   #Names beginning A : “^A”
str_subset(c(babynames$name),"^A")
```

	year	sex	name	n	prop
1	1880	F	Mary	7065	0.072384329
2	1880	F	Anna	2604	0.026679234
3	1880	F	Emma	2003	0.020521700
4	1880	F	Elizabeth	1939	0.019865989
5	1880	F	Minnie	1746	0.017888611
6	1880	F	Margaret	1578	0.016167370

Names Beginning With ‘A’

```
str_subset(c(babynames$name),"^A")
```

	year	sex	name	n	prop
1	1880	F	Mary	7065	0.072384329
2	1880	F	Anna	2604	0.026679234
3	1880	F	Emma	2003	0.020521700
4	1880	F	Elizabeth	1939	0.019865989
5	1880	F	Minnie	1746	0.017888611
6	1880	F	Margaret	1578	0.016167370

```
> str_subset(c(babynames$name),"^A")
 [1] "Anna"      "Alice"     "Annie"     "Ada"       "Agnes"     "Alma"      "Addie"     "Amanda"
 [9] "Amelia"    "Amy"       "Augusta"   "Anne"      "Ann"       "Allie"     "Alta"      "Alberta"
[17] "Abbie"     "Adelaide"  "Adeline"   "Adele"     "Angie"     "Artie"     "Alvina"    "Annette"
[25] "Adella"    "Alpha"     "Angeline"  "Adah"      "Adaline"   "Almeda"    "Aurelia"   "Antoinette"
[33] "Adelia"    "Annetta"   "Antonia"   "Alida"     "Alva"      "Agatha"    "America"   "Anita"
[41] "Arminta"   "Adda"      "Avis"      "Aimee"     "Annabel"   "Ava"       "Abigail"   "Aline"
[49] "Altha"     "Anastasia" "Adela"     "Althea"    "Amalia"    "Amber"     "Angelina"  "Annabelle"
[57] "Anner"     "Arie"      "Adline"    "Almira"    "Alvena"    "Arizona"   "Albertina" "Albina"
[65] "Alyce"     "Amie"      "Angela"    "Annis"     "Abby"      "Aileen"    "Alba"      "Alda"
```

Names Beginning With ‘Amel’

```
str_subset(c(babynames$name),"^Amel")
```

```
> str_subset(c(babynames$name),"^Ameli")
 [1] "Amelia"   "Amelia"   "Amelia"   "Amelia"   "Amelia"   "Amelia"
 [8] "Amelie"   "Amelia"   "Amelie"   "Amelia"   "Amelia"   "Amelia"
[15] "Amelie"   "Amelia"   "Amelia"   "Amelia"   "Amelia"   "Amelia"
[22] "Amelia"   "Amelia"   "Amelia"   "Amelie"   "Amelia"   "Amelia"
[29] "Amelie"   "Amelia"   "Amelie"   "Amelia"   "Amelie"   "Amelia"
[36] "Amelie"   "Amelia"   "Amelie"   "Amelia"   "Amelie"   "Amelia"
[43] "Amelie"   "Amelia"   "Amelia"   "Amelio"   "Amelia"   "Amelie"
[50] "Amelia"   "Amelie"   "Amelio"   "Amelia"   "Amelie"   "Amelio"
[57] "Amelie"   "Amelio"   "Amelia"   "Amelie"   "Amelio"   "Amelia"
[64] "Amelio"   "Amelia"   "Amelie"   "Amelita"  "Amelio"   "Amelia"
[71] "Amelita"  "Amelio"   "Amelia"   "Amelie"   "Amelio"   "Amelia"
[78] "Amelita"  "Amelio"   "Amelia"   "Amelita"  "Amelio"   "Amelia"
```

Reduce the List of Names Beginning With Chars

```
unique(str_subset(c(babynames$name),"^Ameli"))
```

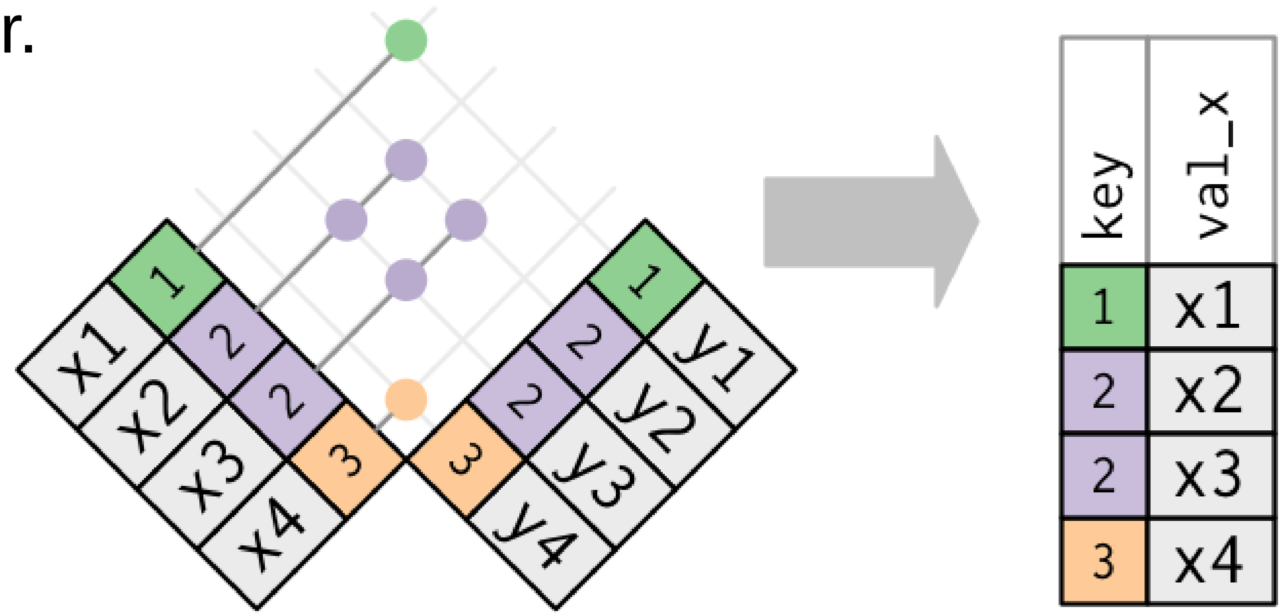
```
> unique(str_subset(c(babynames$name),"^Ameli"))
 [1] "Amelia"      "Amelie"      "Amelio"      "Amelita"     "Amelinda"    "Ameliah"     "Ameli"
 [8] "Amelina"     "Ameliya"     "Ameliana"    "Ameliyah"    "Amelianna"   "Ameliagrace" "Ameliarose"
[15] "Ameline"
```

```
unique(str_subset(c(babynames$name),"^Oli"))
```

```
> unique(str_subset(c(babynames$name),"^Oli"))
 [1] "Olive"       "Olivia"      "Olie"        "Oliver"      "Olin"        "Olivine"     "Olinda"
 [8] "Oline"       "Oliva"       "Olivette"    "Olia"        "Olita"       "Olimpia"     "Olivene"
[15] "Olis"        "Olida"       "Olindo"      "Olice"       "Olivet"      "Olivea"      "Olif"
[22] "Olivett"     "Olivier"     "Olivama"     "Olivio"      "Oliverio"    "Olivera"     "Olisa"
[29] "Olinka"      "Olisha"      "Olibia"      "Olina"       "Olicia"      "Oliviah"     "Oliwia"
[36] "Olivya"      "Oliviagrace" "Oliviana"    "Oliviya"     "Oliviarose"  "Oliana"      "Oliveah"
[43] "Oliviyah"    "Olivianna"   "Oliviamarie" "Oli"         "Oliyah"      "Olisaemeka"  "Oliviaann"
[50] "Olivyah"     "Olijah"      "Olianna"     "Olivija"     "Oliber"      "Oliverjames"
```

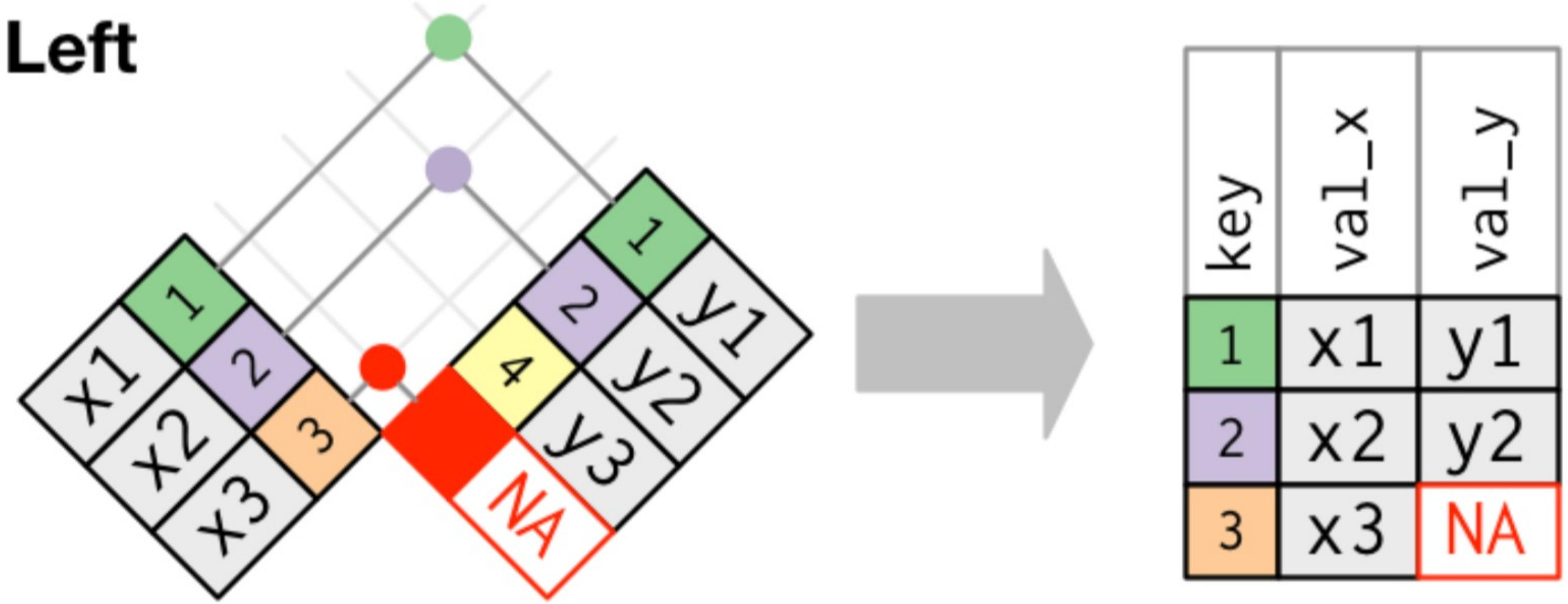
Mutating Joins

- A *mutating join* allows to combine variables from two tables (into one).
- How works:
  - Matches observations by particular keys
  - Copies entries across variables from one table to the other.



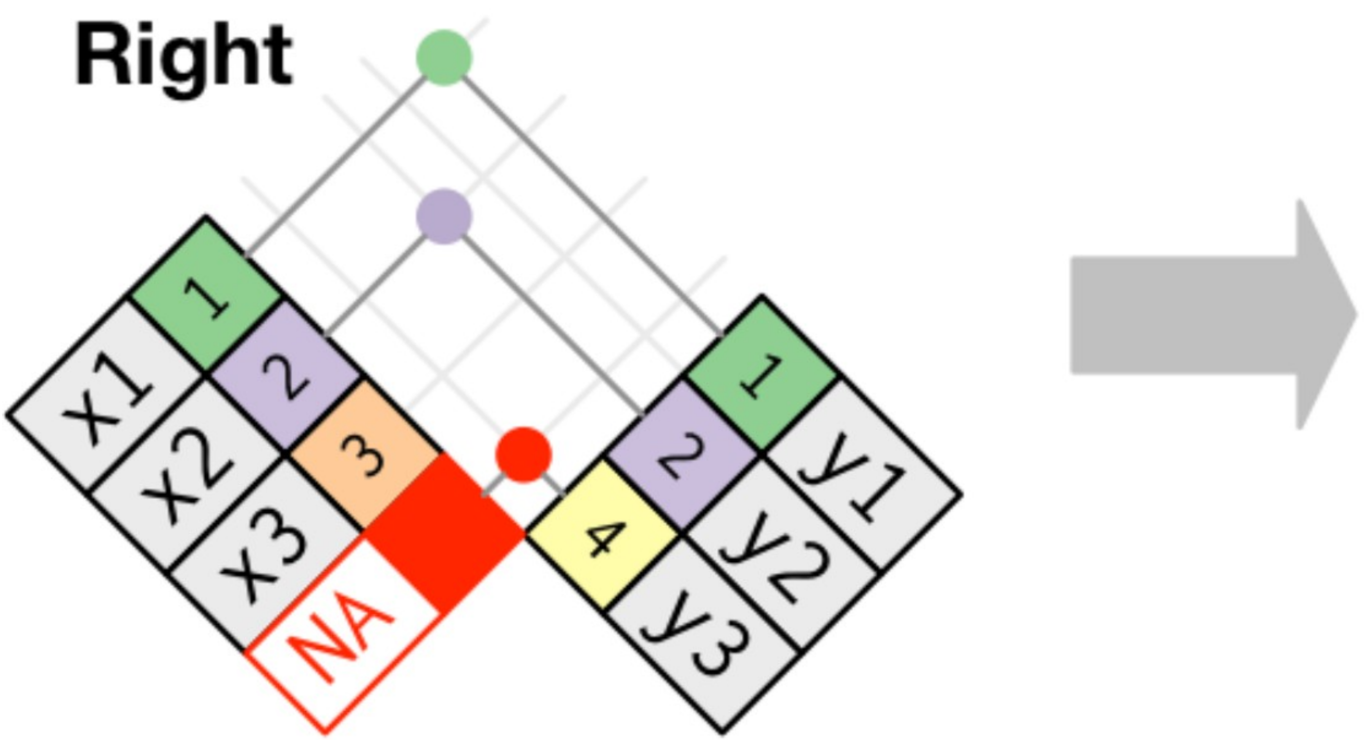
Left Join

- The ***left*** side with the ***x***'s is used to determine a column.
- Missing data (***y***3) from the right-side is shown to be missing.



Right Join

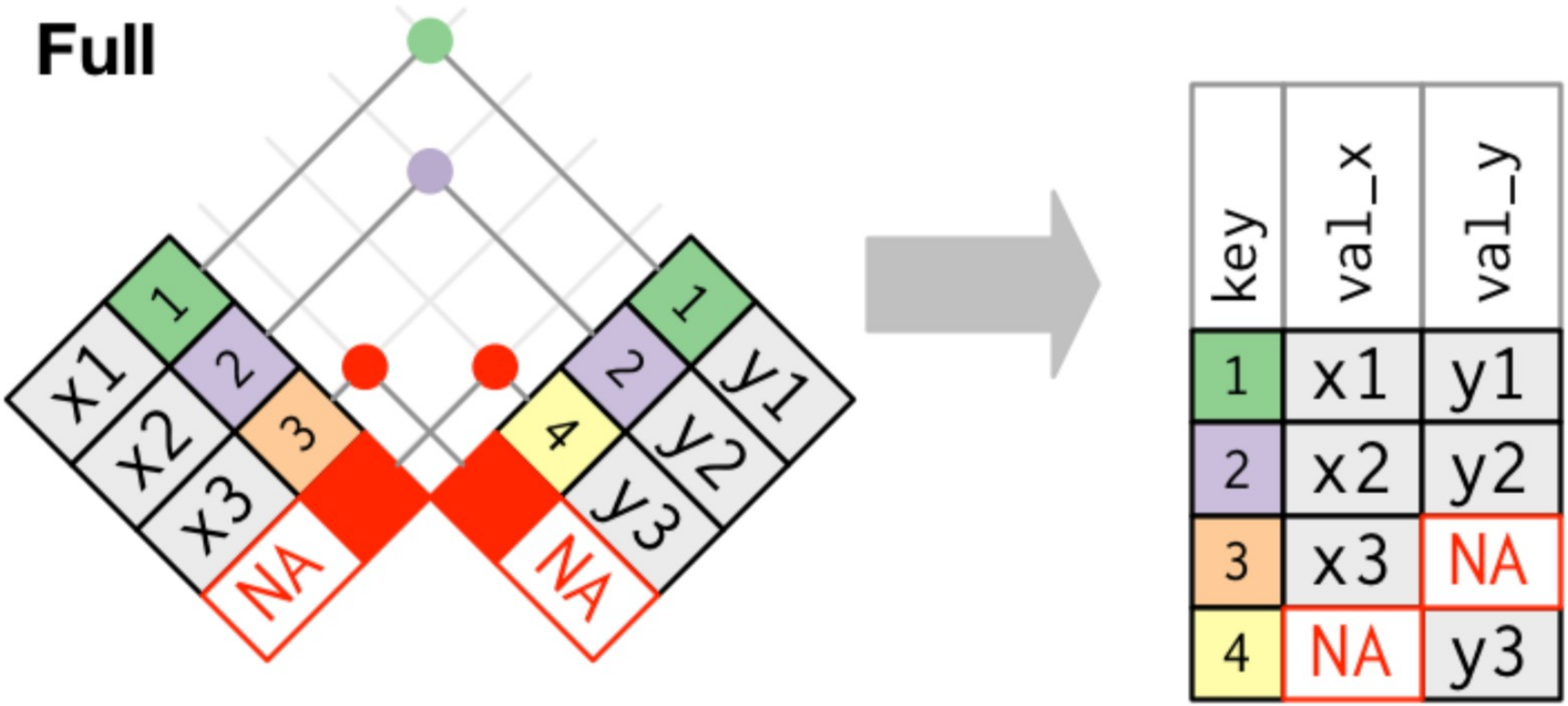
- The ***right*** side with the ***y***’s is used to determine a column.
- Missing data (***x***3) from the left-side is shown to be missing.



key	val_x	val_y
1	x1	y1
2	x2	y2
4	NA	y3

Full Join

- The ***left and right*** sides are used to determine a column.
- Missing data from either the left-side or the right-side is shown to be missing.



Show Me Some Joins

```
# Create two small tables to experiment on
x <- tribble(~key, ~val_x, 1, "x1", 2, "x2", 3, "x3")
y <- tribble(~key, ~val_y, 1, "y1", 2, "y2", 4, "y3")

# Note where the location of missing entries
lj <- left_join(x, y, by="key")
rj <- right_join(x, y, by="key")
fj <- full_join(x, y, by="key")
```

Mutating Joins

```
View(flights)

# First we must reduce the data set to flights2 using the select() feature


flights2 <- flights %>%
select(year:day, hour, origin, dest,
tailnum, carrier)

View(flights2)
```

The Go-Bigger Demo: Left Joins

```
# The left side with the x’s is used to determine a column.

# To add the full airline name to the flights2 data table, we combine the airlines and flights2 data frames with left_join()

# First, remove the origin and dest columns

flights3 <- flights2 %>% select(-origin, -dest) %>%  left_join(airlines, by = "carrier")
```

The Theory: Left Joins

	year	month	day	hour	origin	dest	tailnum	carrier
1	2013	1	1	5	EWR	IAH	N14228	UA
2	2013	1	1	5	LGA	IAH	N24211	UA
3	2013	1	1	5	JFK	MIA	N619AA	AA
4	2013	1	1	5	JFK	BQN	N804JB	B6
5	2013	1	1	6	LGA	ATL	N668DN	DL
6	2013	1	1	5	EWR	ORD	N39463	UA

Flight2 (left)

	carrier	name
1	9E	Endeavor Air Inc.
2	AA	American Airlines Inc.
3	AS	Alaska Airlines Inc.
4	B6	JetBlue Airways
5	DL	Delta Air Lines Inc.
6	EV	ExpressJet Airlines Inc.

Airlines (right)

	year	month	day	hour	tailnum	carrier	name
1	2013	1	1	5	N14228	UA	United Air Lines Inc.
2	2013	1	1	5	N24211	UA	United Air Lines Inc.
3	2013	1	1	5	N619AA	AA	American Airlines Inc.
4	2013	1	1	5	N804JB	B6	JetBlue Airways
5	2013	1	1	6	N668DN	DL	Delta Air Lines Inc.
6	2013	1	1	5	N39463	UA	United Air Lines Inc.

Flight3 (left join)

Connecting Keys: Left Joins

	year	month	day	hour	origin	dest	tailnum	carrier
1	2013	1	1	5	EWR	IAH	N14228	UA
2	2013	1	1	5	LGA	IAH	N24211	UA
3	2013	1	1	5	JFK	MIA	N619AA	AA
4	2013	1	1	5	JFK	BQN	N804JB	B6
5	2013	1	1	6	LGA	ATL	N668DN	DL
6	2013	1	1	5	EWR	ORD	N39463	UA

Flight2

	carrier	name
1	9E	Endeavor Air Inc.
2	AA	American Airlines Inc.
3	AS	Alaska Airlines Inc.
4	B6	JetBlue Airways
5	DL	Delta Air Lines Inc.
6	EV	ExpressJet Airlines Inc.

airlines

	year	month	day	hour	tailnum	carrier	name
1	2013	1	1	5	N14228	UA	United Air Lines Inc.
2	2013	1	1	5	N24211	UA	United Air Lines Inc.
3	2013	1	1	5	N619AA	AA	American Airlines Inc.
4	2013	1	1	5	N804JB	B6	JetBlue Airways
5	2013	1	1	6	N668DN	DL	Delta Air Lines Inc.
6	2013	1	1	5	N39463	UA	United Air Lines Inc.

Flight3

Another Way To Left Join

```
# Another way to make a left join is to
use the mutate() method

flights2 %>%  select(-origin, -dest) %>%
mutate(name = airlines$name[match(carrier,
airlines$carrier)])
```

Verify that this alternative way works to produce the same table (flight3) as before.

The Theory: Right Joins

	year	month	day	hour	origin	dest	tailnum	carrier
1	2013	1	1	5	EWR	IAH	N14228	UA
2	2013	1	1	5	LGA	IAH	N24211	UA
3	2013	1	1	5	JFK	MIA	N619AA	AA
4	2013	1	1	5	JFK	BQN	N804JB	B6
5	2013	1	1	6	LGA	ATL	N668DN	DL
6	2013	1	1	5	EWR	ORD	N39463	UA

Flight2 (left)

	carrier	name
1	9E	Endeavor Air Inc.
2	AA	American Airlines Inc.
3	AS	Alaska Airlines Inc.
4	B6	JetBlue Airways
5	DL	Delta Air Lines Inc.
6	EV	ExpressJet Airlines Inc.

Airlines (right)

	carrier	name	year	month	day	hour	tailnum
1	UA	United Air Lines Inc.	2013	1	1	5	N14228
2	UA	United Air Lines Inc.	2013	1	1	5	N24211
3	AA	American Airlines Inc.	2013	1	1	5	N619AA
4	B6	JetBlue Airways	2013	1	1	5	N804JB
5	DL	Delta Air Lines Inc.	2013	1	1	6	N668DN
6	UA	United Air Lines Inc.	2013	1	1	5	N39463
7	B6	JetBlue Airways	2013	1	1	6	N516JB
8	EV	ExpressJet Airlines Inc.	2013	1	1	6	N829AS

Flight3 (right join)