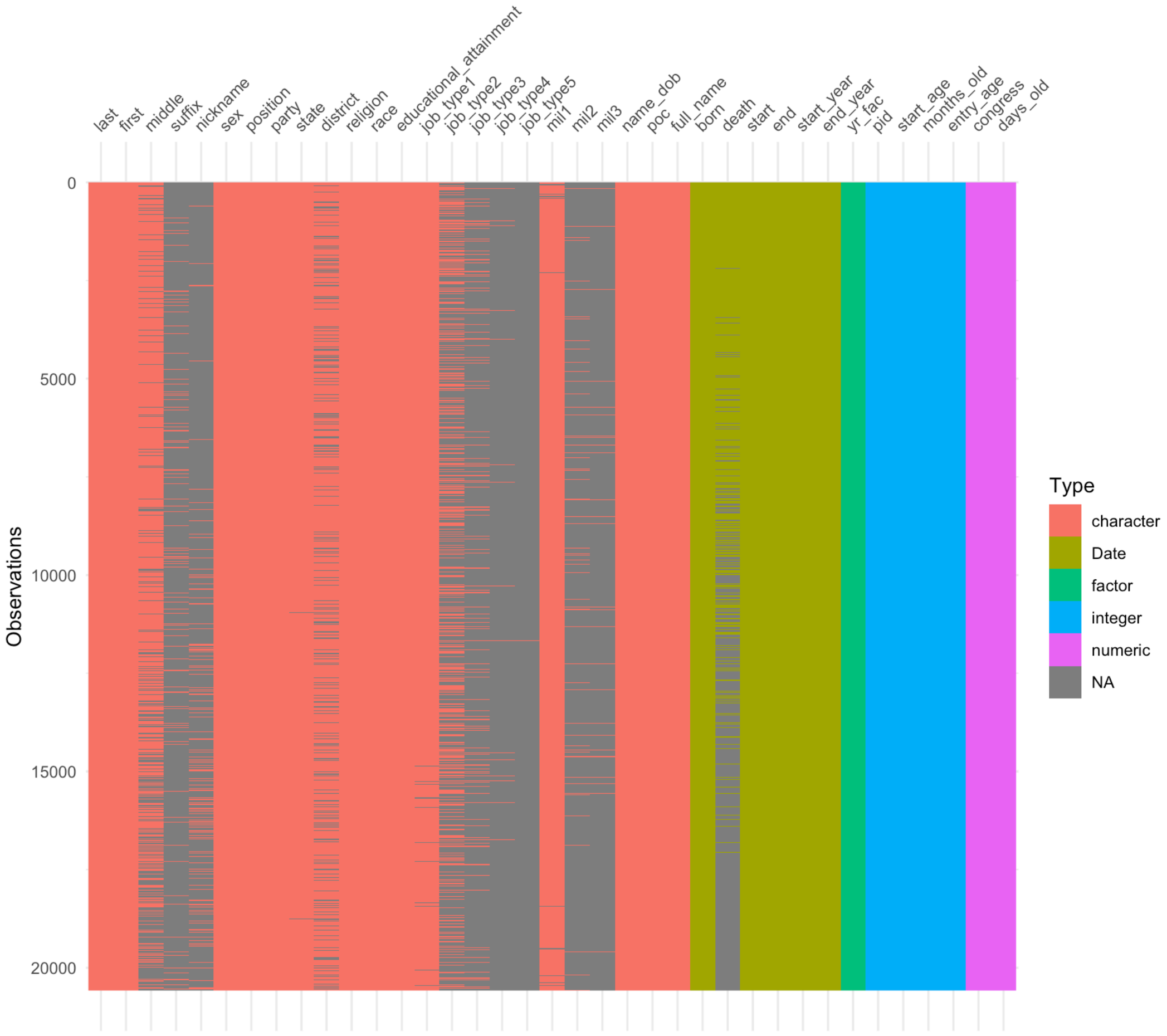
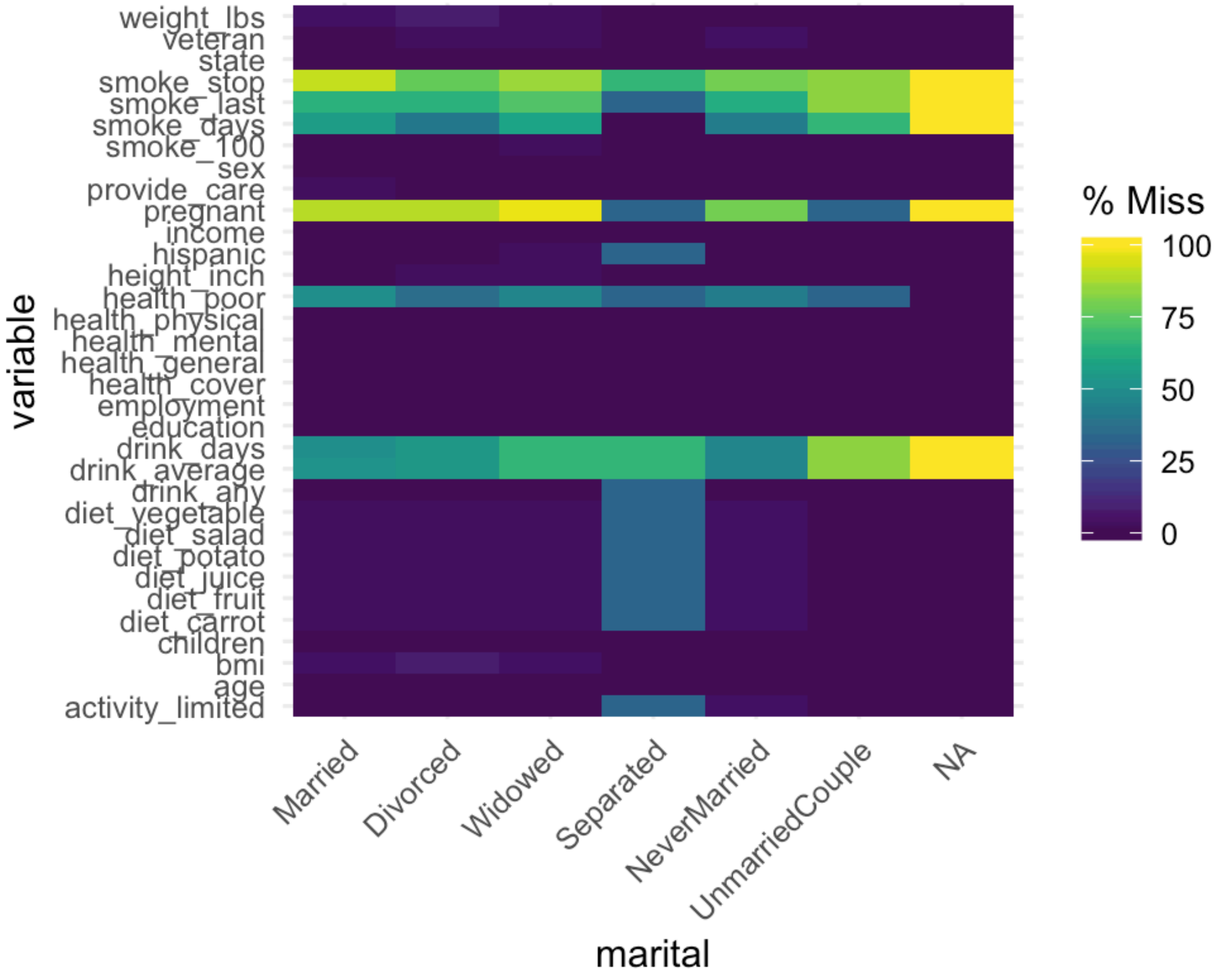
~ / > _

More on dplyr

~/> previously …


last
first
middle
suffix
nickname
sex
position
party
state
district
religion
race
educational_attainment
job_type1
job_type2
job_type3
job_type4
job_type5
mil1
mil2
mil3
name_dob
poc
full_name
born
death
start
end
start_year
end_year
yr_fac
pid
start_age
months_old
entry_age
congress
days_old
0
5000
10000
15000
20000
Observations
Type
character
Date
factor
integer
numeric
NA

```
gg_miss_fct(x = riskfactors, fct = marital)
```



```r
quick_na <- function(x, vals = c(9, 10, 11, 97, 99)) {
      x[x %in% vals] <- NA
      x
}

num_vec <- c(1:12, 97, 97, 99, NA)

num_vec
#>  [1]  1  2  3  4  5  6  7  8  9 10 11 12 97 97 99 NA

quick_na(num_vec)
#>  [1]  1  2  3  4  5  6  7  8 NA NA NA 12 NA NA NA NA
```

~ /> _

Working with dplyr

~ /> _

Standard verbs

Group the data at the level we want, such as “Religion by Region” or “Authors by Publications by Year”.

`group_by()`

Filter or ***Select*** pieces of the data. This gets us the subset of the table we want to work on.

`filter()` rows
`select()` columns

Mutate the data by creating new variables at the *current* level of grouping. Mutating adds new columns to the table.

`mutate()`

Summarize or aggregate the grouped data. This creates new variables at a higher level of grouping. For example we might calculate means with mean() or counts with n(). This results in a smaller, summary table, which we might do more things with if we want.

`summarize()`

~ /> _

Scoped verbs

Scoped Verbs

`action_all()` Take action on all variables

`action_if()` Take action on a subset of variables selected by a criterion

`action_at()` Take action on a subset of variables selected by their names

`action` can be **`mutate summarize filter`**

Useful scope-setters

```
is.character()

is.factor()

is.numeric()

is.logical()

is.integer()

is.ordered()

lubridate::is.Date()
```

Useful scoping helpers

```
starts_with()

ends_with()

contains()

one_of()

matches()

vars()

everything()
```

Examples

```
organdata %>%
  group_by(world) %>%
  summarize_if(is.numeric, mean, na.rm = TRUE) %>%
  select(world, donors, pubhealth, roads) %>%
  select_all(tools::toTitleCase)
```

```
# A tibble: 4 x 4
  World        Donors Pubhealth Roads
  <chr>         <dbl>     <dbl> <dbl>
1 NA             28.1      5.45 161.
2 Corporatist    16.8      6.40 132.
3 Liberal        15.6      5.75 111.
4 SocDem         14.8      6.54  82.7
```

Examples

```
organdata %>%
   group_by(country) %>%
   summarize_if(is.numeric,
                funs(avg = mean, sd = sd),
                na.rm = TRUE) %>%
select(country, donors_avg,
         donors_sd, roads_avg, roads_sd) %>%
arrange(desc(donors_avg))
```

Examples

A tibble: 17 x 5

	country	donors_avg	donors_sd	roads_avg	roads_sd
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	Spain	28.1	4.96	161.	35.3
2	Austria	23.5	2.42	150.	30.3
3	Belgium	21.9	1.94	155.	20.6
4	United States	20.0	1.33	155.	8.35
5	Ireland	19.8	2.48	118.	10.8
6	Finland	18.4	1.53	93.6	19.0
7	France	16.8	1.60	156.	20.1
8	Norway	15.4	1.11	70.0	6.68
9	Switzerland	14.2	1.71	96.4	21.7
10	Canada	14.0	0.751	109.	17.7
11	Netherlands	13.7	1.55	76.1	9.93
12	United Kingdom	13.5	0.775	67.9	10.5
13	Sweden	13.1	1.75	72.3	13.2
14	Denmark	13.1	1.47	102.	12.4
15	Germany	13.0	0.611	113.	25.9
16	Italy	11.1	4.28	122.	10.2
17	Australia	10.6	1.14	105.	14.3

~ /> _

Scoping and Mapping

map() and friends are the general case

```
out <- lm(donors ~ pop + gdp + roads, data = organdata)
```

```
summary(out)
```

```
Call:
lm(formula = donors ~ pop + gdp + roads, data = organdata)

Residuals:
    Min      1Q  Median      3Q     Max 
-13.423  -2.658  -0.080   1.963  15.864 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.506e+00  2.364e+00   1.906   0.0580 .  
pop         -1.153e-05  5.643e-06  -2.043   0.0423 *  
gdp          1.082e-04  7.527e-05   1.438   0.1521    
roads        8.988e-02  1.032e-02   8.710 1.14e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.325 on 200 degrees of freedom
  (34 observations deleted due to missingness)
Multiple R-squared:  0.2944,  Adjusted R-squared:  0.2838 
F-statistic: 27.81 on 3 and 200 DF,  p-value: 4.486e-15
```

map() and friends are the general case

```
> names(summary(out))
 [1] "call"          "terms"         "residuals"
"coefficients"  "aliased"
 [6] "sigma"         "df"            "r.squared"
"adj.r.squared" "fstatistic"
[11] "cov.unscaled"  "na.action"
```

map() and friends are the general case

```
organdata %>%
  split(.$world) %>%
  map(~ lm(donors ~ pop + gdp + roads, data = .)) %>%
  map(summary) %>%
  map_dbl("r.squared")
```

We'll see cleaner ways to do this shortly

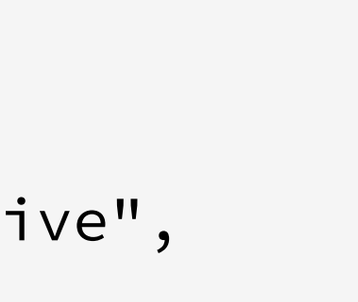
~ /> _

Zero Counts in dplyr

```
data %>%
    select(start_year, job_type1) %>%
    group_by(start_year, job_type1) %>%
    summarize(n = n()) %>%
    mutate(pct = (n/sum(n))*100)
```

```
  # A tibble: 689 x 4
  # Groups:   start_year [38]
     start_year job_type1                          n    pct
     <date>     <chr>                          <int>  <dbl>
   1 1945-01-03 NA                                 5  0.880
   2 1945-01-03 Acting/entertainer                11  1.94
   3 1945-01-03 Aeronautics                        2  0.352
   4 1945-01-03 Agriculture                       65 11.4
   5 1945-01-03 Business or banking              108 19.0
   6 1945-01-03 Clergy                             3  0.528
   7 1945-01-03 Congressional Aide                11  1.94
   8 1945-01-03 Construction/building trades       9  1.58
   9 1945-01-03 Education                         58 10.2
  10 1945-01-03 Engineering                        2  0.352
  # … with 679 more rows
```

```
df <- data %>%
    filter(position == "U.S. Representative",
           start > "1945-01-01") %>%
    group_by(pid) %>%
    nest() %>%
    mutate(data = map(data, ~ mutate(.x,
                  term_id = 1 + congress - first(congress)))) %>%
unnest() %>%
filter(term_id == 1,
       party %in% c("Democrat", "Republican"),
       start_year > int_to_year(2012)) %>%
group_by(start_year, party, sex) %>%
select(pid, start_year, party, sex)
```



This caused the difference in N you saw in class. Fixed here.

```
> df
# A tibble: 293 x 4
# Groups:   start_year, party, sex [14]
     pid start_year party      sex
   <int> <date>     <chr>      <chr>
 1  3160 2013-01-03 Republican M
 2  3161 2013-01-03 Democrat   F
 3  3162 2013-01-03 Democrat   M
 4  3163 2013-01-03 Republican M
 5  3164 2013-01-03 Democrat   M
 6  3165 2013-01-03 Republican M
 7  3166 2013-01-03 Republican M
 8  3167 2013-01-03 Democrat   F
 9  3168 2013-01-03 Republican M
10  3169 2013-01-03 Democrat   M
# … with 283 more rows
```

```r
df %>%
    group_by(start_year, party, sex) %>%
    summarize(N = n()) %>%
    mutate(freq = N / sum(N))
```

```
#> # A tibble: 14 x 5
#> # Groups:   start_year, party [8]
#>    start_year party      sex       N   freq
#>    <date>     <chr>      <chr> <int>  <dbl>
#>  1 2013-01-03 Democrat   F        21 0.362
#>  2 2013-01-03 Democrat   M        37 0.638
#>  3 2013-01-03 Republican F         8 0.101
#>  4 2013-01-03 Republican M        71 0.899
#>  5 2015-01-03 Democrat   M         1 1
#>  6 2015-01-03 Republican M         5 1
#>  7 2017-01-03 Democrat   F         6 0.24
#>  8 2017-01-03 Democrat   M        19 0.76
#>  9 2017-01-03 Republican F         2 0.0667
#> 10 2017-01-03 Republican M        28 0.933
#> 11 2019-01-03 Democrat   F        33 0.647
#> 12 2019-01-03 Democrat   M        18 0.353
#> 13 2019-01-03 Republican F         1 0.0323
#> 14 2019-01-03 Republican M        30 0.968
```

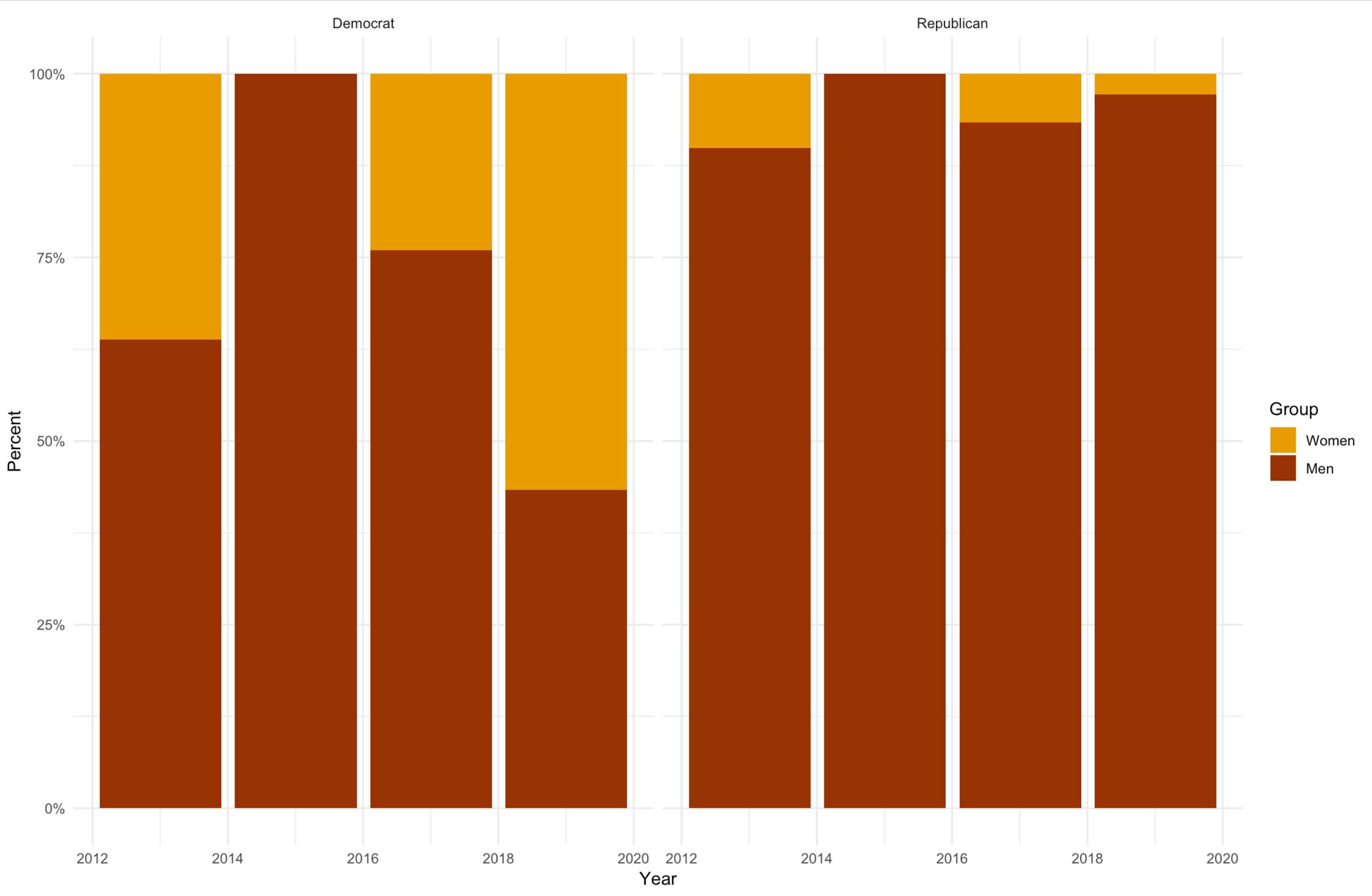
```r
## Hex colors for sex
sex_colors <- c("#E69F00", "#993300")

## Hex color codes for Dem Blue and Rep Red
party_colors <- c("#2E74C0", "#CB454A")
```

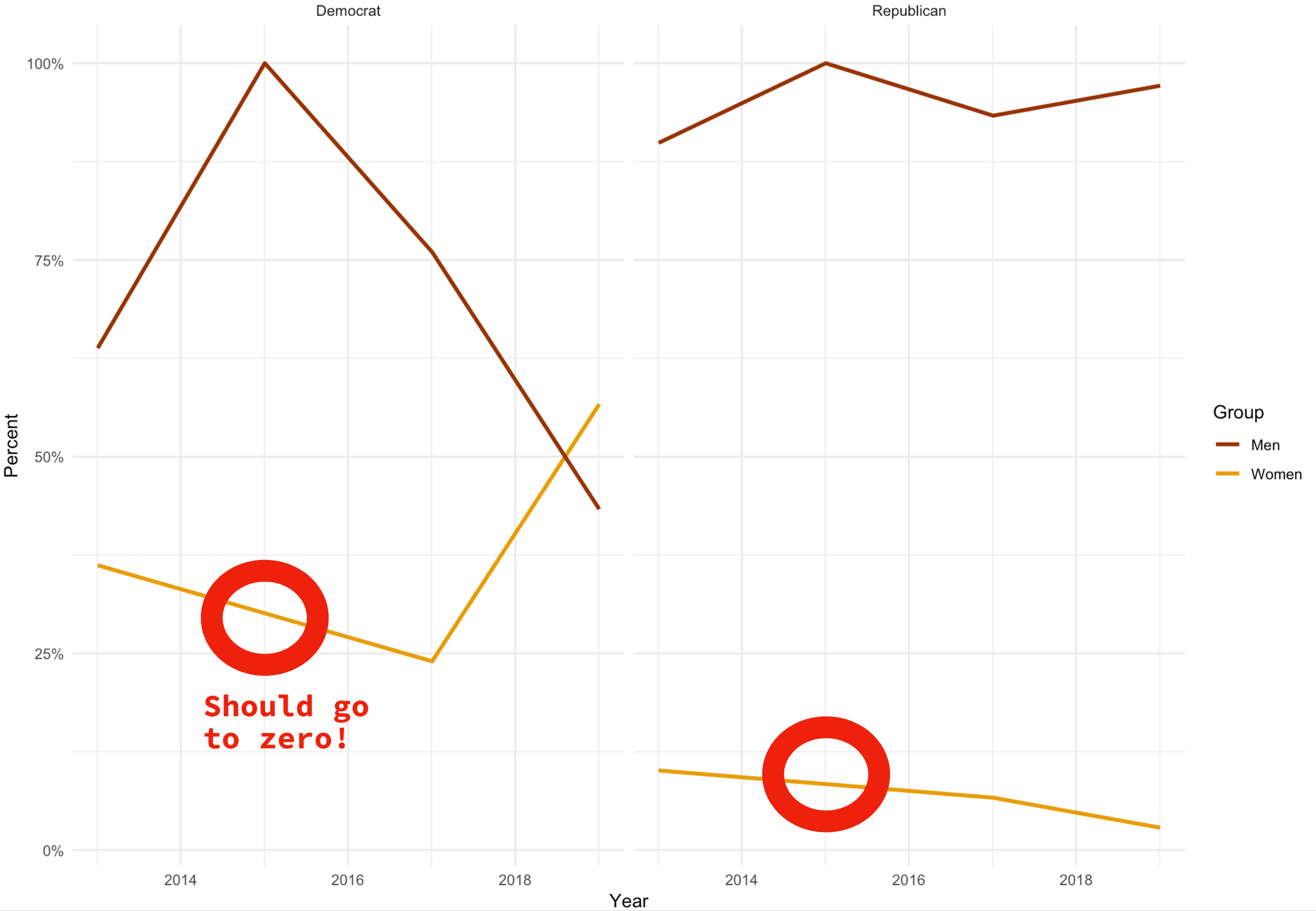
```r
## Group labels
mf_labs <- tibble(M = "Men", F = "Women")

theme_set(theme_minimal())
```

```r
df %>%
    group_by(start_year, party, sex) %>%
    summarize(N = n()) %>%
    mutate(freq = N / sum(N)) %>%
    ggplot(aes(x = start_year,
               y = freq,
               fill = sex)) +
    geom_col() +
    scale_y_continuous(labels = scales::percent) +
    scale_fill_manual(values = sex_colors, labels = c("Women", "Men")) +
    labs(x = "Year", y = "Percent", fill = "Group") +
    facet_wrap(~ party)
```


Democrat
Republican
100%
75%
50%
25%
0%
Percent
2012
2014
2016
2018
2020
Year
Group
Women
Men

```r
df %>%
    group_by(start_year, party, sex) %>%
    summarize(N = n()) %>%
    mutate(freq = N / sum(N)) %>%
    ggplot(aes(x = start_year,
               y = freq,
               color = sex)) +
    geom_line(size = 1.1) +
    scale_y_continuous(labels = scales::percent) +
    scale_color_manual(values = sex_colors, labels = c("Women", "Men")) +
    guides(color = guide_legend(reverse = TRUE)) +
    labs(x = "Year", y = "Percent", color = "Group") +
    facet_wrap(~ party)
```


Democrat
Republican
100%
75%
50%
25%
0%
Percent
2014
2016
2018
Year
Group
Men
Women
Should go
to zero!

```r
df_f <- df %>% modify_if(is.character, as.factor)

df_f %>%
    group_by(start_year, party, sex) %>%
    tally()
```

```
#> # A tibble: 16 x 4
#> # Groups:   start_year, party [8]
#>    start_year party      sex       n
#>    <date>     <fct>      <fct> <int>
#>  1 2013-01-03 Democrat   F        21
#>  2 2013-01-03 Democrat   M        37
#>  3 2013-01-03 Republican F         8
#>  4 2013-01-03 Republican M        71
#>  5 2015-01-03 Democrat   F         0
#>  6 2015-01-03 Democrat   M         1
#>  7 2015-01-03 Republican F         0
#>  8 2015-01-03 Republican M         5
```

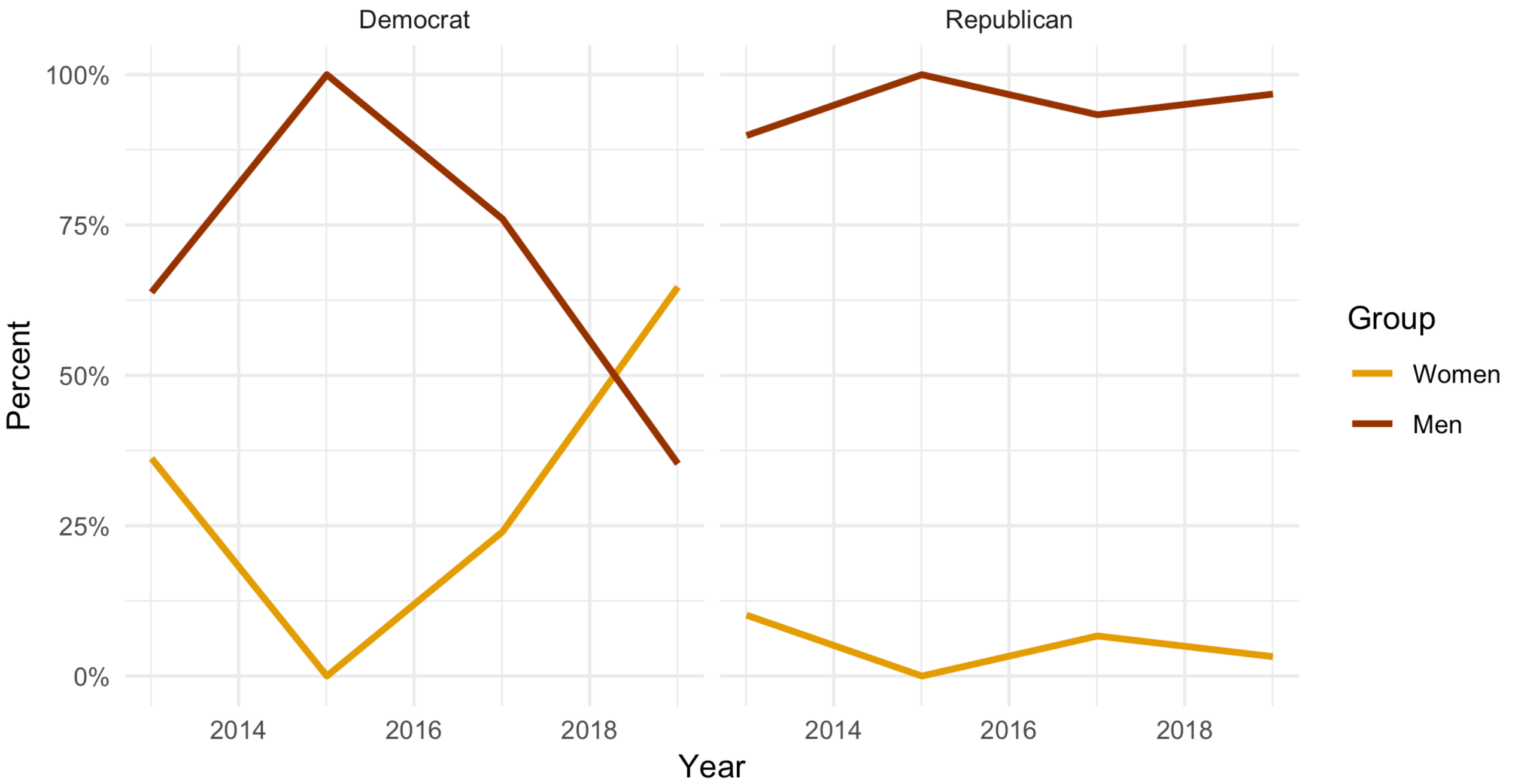
Option 1: Convert to Factor

```r
df %>%
    group_by(start_year, party, sex) %>%
    summarize(N = n()) %>%
    mutate(freq = N / sum(N)) %>%
    ungroup() %>%
    complete(start_year, party, sex,
             fill = list(N = 0, freq = 0))
```

```
#> # A tibble: 16 x 5
#>    start_year party      sex       N   freq
#>    <date>     <chr>      <chr> <dbl>  <dbl>
#>  1 2013-01-03 Democrat   F        21 0.362
#>  2 2013-01-03 Democrat   M        37 0.638
#>  3 2013-01-03 Republican F         8 0.101
#>  4 2013-01-03 Republican M        71 0.899
#>  5 2015-01-03 Democrat   F         0 0
#>  6 2015-01-03 Democrat   M         1 1
#>  7 2015-01-03 Republican F         0 0
#>  8 2015-01-03 Republican M         5 1
```

Option 2: ungroup() & complete()

```r
df_f %>%
    group_by(start_year, party, sex) %>%
    summarize(N = n()) %>%
    mutate(freq = N / sum(N)) %>%
    ggplot(aes(x = start_year,
               y = freq,
               color = sex)) +
    geom_line(size = 1.1) +
    scale_y_continuous(labels = scales::percent) +
    scale_color_manual(values = sex_colors, labels = c("Women", "Men")) +
    guides(color = guide_legend(reverse = TRUE)) +
    labs(x = "Year", y = "Percent", color = "Group") +
    facet_wrap(~ party)
```


Democrat
Republican
100%
75%
50%
25%
0%
Percent
2014
2016
2018
Year
Group
Women
Men

~ /> _

Spreading multiple values

```
edu
```

```
## # A tibble: 366 x 11
##    age   sex    year total elem4 elem8   hs3   hs4 coll3 coll4 median
##    <chr> <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>
##  1 25-34 Male   2016 21845   116   468  1427  6386  6015  7432     NA
##  2 25-34 Male   2015 21427   166   488  1584  6198  5920  7071     NA
##  3 25-34 Male   2014 21217   151   512  1611  6323  5910  6710     NA
##  4 25-34 Male   2013 20816   161   582  1747  6058  5749  6519     NA
##  5 25-34 Male   2012 20464   161   579  1707  6127  5619  6270     NA
##  6 25-34 Male   2011 20985   190   657  1791  6444  5750  6151     NA
##  7 25-34 Male   2010 20689   186   641  1866  6458  5587  5951     NA
##  8 25-34 Male   2009 20440   184   695  1806  6495  5508  5752     NA
##  9 25-34 Male   2008 20210   172   714  1874  6356  5277  5816     NA
## 10 25-34 Male   2007 20024   246   757  1930  6361  5137  5593     NA
## # ... with 356 more rows
```

```r
edu_t <- gather(data = edu,
                key = school,
                value = freq,
                elem4:coll4)

head(edu_t)
```

```
## # A tibble: 6 x 7
##   age   sex    year total median school  freq
##   <chr> <chr> <int> <int>  <dbl> <chr>  <dbl>
## 1 25-34 Male   2016 21845     NA elem4    116
## 2 25-34 Male   2015 21427     NA elem4    166
## 3 25-34 Male   2014 21217     NA elem4    151
## 4 25-34 Male   2013 20816     NA elem4    161
## 5 25-34 Male   2012 20464     NA elem4    161
## 6 25-34 Male   2011 20985     NA elem4    190
```

```r
tail(edu_t)
```

```
## # A tibble: 6 x 7
##   age   sex     year total median school  freq
##   <chr> <chr>  <int> <int>  <dbl> <chr>  <dbl>
## 1 55>   Female  1959 16263   8.30 coll4    688
## 2 55>   Female  1957 15581   8.20 coll4    630
## 3 55>   Female  1952 13662   7.90 coll4    628
## 4 55>   Female  1950 13150   8.40 coll4    436
## 5 55>   Female  1947 11810   7.60 coll4    343
## 6 55>   Female  1940  9777   8.30 coll4    219
```

```r
gen_cats <- function(x, N = 1000) {
    sample(x, N, replace = TRUE)
}

set.seed(101)
N <- 1000

income <- rnorm(N, 100, 50)

vars <- list(stratum = c(1:8),
             sex = c("M", "F"),
             race =  c("B", "W"),
             educ = c("HS", "BA"))

df <- as_tibble(map_dfc(vars, gen_cats))
df <- add_column(df, income)

df
```

```
#>  A tibble: 1,000 x 5
#>     stratum sex   race  educ  income
#>       <int> <chr> <chr> <chr>  <dbl>
#>  1       6 F     W     BA      83.7
#>  2       7 F     B     HS     128.
#>  3       1 F     B     BA      66.3
#>  4       2 M     B     BA     111.
#>  5       4 M     B     BA     116.
#>  6       6 F     W     BA     159.
#>  7       7 F     W     BA     131.
#>  8       6 M     W     HS      94.4
#>  9       4 F     W     HS     146.
#> 10       7 M     B     BA      88.8
#> # ... with 990 more rows
```

```r
## Simple tidy summary
tv_wide1 <- df %>% group_by(sex, race, stratum, educ) %>%
    summarize(mean_inc = mean(income), N = n())

tv_wide1
# A tibble: 64 x 6
# Groups:   sex, race, stratum [?]
   sex   race  stratum educ  mean_inc     N
   <chr> <chr>   <int> <chr>    <dbl> <int>
 1 F     B           1 BA       102.     16
 2 F     B           1 HS       116.     23
 3 F     B           2 BA        93.0    12
 4 F     B           2 HS        90.5    15
 5 F     B           3 BA       114.     18
 6 F     B           3 HS       104.     11
 7 F     B           4 BA        90.5    13
 8 F     B           4 HS       103.     15
 9 F     B           5 BA       111.     12
10 F     B           5 HS        70.7    10
# ... with 54 more rows
```

```r
## 1. gather()
tv_wide2 <- df %>% group_by(sex, race, stratum, educ) %>%
    summarize(mean_inc = mean(income), N = n()) %>%
    gather(variable, value, -(sex:educ))

tv_wide2

# A tibble: 128 x 6
# Groups:   sex, race, stratum [32]
   sex   race  stratum educ  variable value
   <chr> <chr>   <int> <chr> <chr>    <dbl>
 1 F     B           1 BA    mean_inc 102.
 2 F     B           1 HS    mean_inc 116.
 3 F     B           2 BA    mean_inc  93.0
 4 F     B           2 HS    mean_inc  90.5
 5 F     B           3 BA    mean_inc 114.
 6 F     B           3 HS    mean_inc 104.
 7 F     B           4 BA    mean_inc  90.5
 8 F     B           4 HS    mean_inc 103.
 9 F     B           5 BA    mean_inc 111.
10 F     B           5 HS    mean_inc  70.7
# ... with 118 more rows
```

```r
## 2. unite()
tv_wide2 <- df %>% group_by(sex, race, stratum, educ) %>%
    summarize(mean_inc = mean(income), N = n()) %>%
    gather(variable, value, -(sex:educ)) %>%
    unite(temp, educ, variable)

tv_wide2

# A tibble: 128 x 5
# Groups:   sex, race, stratum [32]
   sex   race  stratum temp          value
   <chr> <chr>   <int> <chr>         <dbl>
 1 F     B           1 BA_mean_inc 102.
 2 F     B           1 HS_mean_inc 116.
 3 F     B           2 BA_mean_inc  93.0
 4 F     B           2 HS_mean_inc  90.5
 5 F     B           3 BA_mean_inc 114.
 6 F     B           3 HS_mean_inc 104.
 7 F     B           4 BA_mean_inc  90.5
 8 F     B           4 HS_mean_inc 103.
 9 F     B           5 BA_mean_inc 111.
10 F     B           5 HS_mean_inc  70.7
# ... with 118 more rows
```

```r
## 3. spread()
tv_wide2 <- df %>% group_by(sex, race, stratum, educ) %>%
    summarize(mean_inc = mean(income), N = n()) %>%
    gather(variable, value, -(sex:educ)) %>%
    unite(temp, educ, variable) %>%
    spread(temp, value)

tv_wide2

# A tibble: 32 x 7
# Groups:   sex, race, stratum [32]
   sex   race  stratum BA_mean_inc  BA_N HS_mean_inc  HS_N
   <chr> <chr>   <int>       <dbl> <dbl>       <dbl> <dbl>
 1 F     B           1        102.    16        116.    23
 2 F     B           2         93.0   12         90.5   15
 3 F     B           3        114.    18        104.    11
 4 F     B           4         90.5   13        103.    15
 5 F     B           5        111.    12         70.7   10
 6 F     B           6         97.8    9         88.7   17
 7 F     B           7         70.2   17         99.0   21
 8 F     B           8        116.    15        101.     8
 9 F     W           1         86.4   20         93.0    8
10 F     W           2        104.    14         93.4   12
# ... with 22 more rows
```

A function to do this

```
multi_spread <- function(df, key, value) {
    # quote key
    keyq <- rlang::enquo(key)
    # break value vector into quotes
    valueq <- rlang::enquo(value)
    s <- rlang::quos(!!valueq)
    df %>% gather(variable, value, !!!s) %>%
        unite(temp, !!keyq, variable) %>%
        spread(temp, value)
}
```

Final Version

```
## Final version
tv_wide3 <- df %>% group_by(sex, race, stratum, educ) %>%
    summarize(mean_inc = mean(income), N = n()) %>%
    multi_spread(educ, c(mean_inc, N))

tv_wide3

# A tibble: 32 x 7
# Groups:   sex, race, stratum [32]
   sex   race  stratum BA_mean_inc  BA_N HS_mean_inc  HS_N
   <chr> <chr>   <int>       <dbl> <dbl>       <dbl> <dbl>
 1 F     B           1        102.    16        116.    23
 2 F     B           2         93.0   12         90.5   15
 3 F     B           3        114.    18        104.    11
 4 F     B           4         90.5   13        103.    15
 5 F     B           5        111.    12         70.7   10
 6 F     B           6         97.8    9         88.7   17
 7 F     B           7         70.2   17         99.0   21
 8 F     B           8        116.    15        101.     8
 9 F     W           1         86.4   20         93.0     8
10 F     W           2        104.    14         93.4   12
# ... with 22 more rows
```

~ /> _

Nesting

Nesting as "super-grouping"

```
gapminder %>% filter(continent == "Europe",
                     year == 1977)
```

```
# A tibble: 30 x 6
   country                  continent  year lifeExp      pop gdpPercap
   <fct>                    <fct>     <int>   <dbl>    <int>     <dbl>
 1 Albania                  Europe     1977    68.9  2509048     3533.
 2 Austria                  Europe     1977    72.2  7568430    19749.
 3 Belgium                  Europe     1977    72.8  9821800    19118.
 4 Bosnia and Herzegovina   Europe     1977    69.9  4086000     3528.
 5 Bulgaria                 Europe     1977    70.8  8797022     7612.
 6 Croatia                  Europe     1977    70.6  4318673    11305.
 7 Czech Republic           Europe     1977    70.7 10161915    14800.
 8 Denmark                  Europe     1977    74.7  5088419    20423.
 9 Finland                  Europe     1977    72.5  4738902    15605.
10 France                   Europe     1977    73.8 53165019    18293.
# … with 20 more rows
```

Nesting as “super-grouping”

```
eu77 <- gapminder %>% filter(continent == "Europe", year == 1977)

fit <- lm(lifeExp ~ log(gdpPercap), data = eu77)
summary(fit)
```

```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap), data = eu77)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.496 -1.031  0.093  1.176  3.712
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      29.489      7.161    4.12  0.00031 ***
## log(gdpPercap)    4.488      0.756    5.94  2.2e-06 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.11 on 28 degrees of freedom
## Multiple R-squared:  0.557,  Adjusted R-squared:  0.541
## F-statistic: 35.2 on 1 and 28 DF,  p-value: 2.17e-06
```

Nesting as "super-grouping"

```
out_le <- gapminder %>%
    group_by(continent, year) %>%
    nest()

out_le
```

```
## # A tibble: 60 x 3
##    continent  year data
##    <fct>     <int> <list>
##  1 Asia       1952 <tibble [33 × 4]>
##  2 Asia       1957 <tibble [33 × 4]>
##  3 Asia       1962 <tibble [33 × 4]>
##  4 Asia       1967 <tibble [33 × 4]>
##  5 Asia       1972 <tibble [33 × 4]>
##  6 Asia       1977 <tibble [33 × 4]>
##  7 Asia       1982 <tibble [33 × 4]>
##  8 Asia       1987 <tibble [33 × 4]>
##  9 Asia       1992 <tibble [33 × 4]>
## 10 Asia       1997 <tibble [33 × 4]>
## # ... with 50 more rows
```

Nesting as “super-grouping”

```
out_le %>% filter(continent == "Europe" & year == 1977) %>%
           unnest()
```

```
## # A tibble: 30 x 6
##    continent  year country                lifeExp    pop gdpPercap
##    <fct>     <int> <fct>                    <dbl>  <int>     <dbl>
##  1 Europe     1977 Albania                  68.9 2.51e⁶      3533
##  2 Europe     1977 Austria                  72.2 7.57e⁶     19749
##  3 Europe     1977 Belgium                  72.8 9.82e⁶     19118
##  4 Europe     1977 Bosnia and Her…          69.9 4.09e⁶      3528
##  5 Europe     1977 Bulgaria                 70.8 8.80e⁶      7612
##  6 Europe     1977 Croatia                  70.6 4.32e⁶     11305
##  7 Europe     1977 Czech Republic           70.7 1.02e⁷     14800
##  8 Europe     1977 Denmark                  74.7 5.09e⁶     20423
##  9 Europe     1977 Finland                  72.5 4.74e⁶     15605
## 10 Europe     1977 France                   73.8 5.32e⁷     18293
## # ... with 20 more rows
```

Nesting as "super-grouping"

```
fit_ols <- function(df) {
    lm(lifeExp ~ log(gdpPercap), data = df)
}

out_le <- gapminder %>%
    group_by(continent, year) %>%
    nest() %>%
    mutate(model = map(data, fit_ols))

out_le
```

```
## # A tibble: 60 x 4
##    continent  year data              model
##    <fct>     <int> <list>            <list>
##  1 Asia       1952 <tibble [33 × 4]> <S3: lm>
##  2 Asia       1957 <tibble [33 × 4]> <S3: lm>
##  3 Asia       1962 <tibble [33 × 4]> <S3: lm>
##  4 Asia       1967 <tibble [33 × 4]> <S3: lm>
##  5 Asia       1972 <tibble [33 × 4]> <S3: lm>
##  6 Asia       1977 <tibble [33 × 4]> <S3: lm>
##  7 Asia       1982 <tibble [33 × 4]> <S3: lm>
##  8 Asia       1987 <tibble [33 × 4]> <S3: lm>
##  9 Asia       1992 <tibble [33 × 4]> <S3: lm>
## 10 Asia       1997 <tibble [33 × 4]> <S3: lm>
## # ... with 50 more rows
```

Nesting as "super-grouping"

```
library(broom)
```

More on broom later in the semester

```
fit_ols <- function(df) {
    lm(lifeExp ~ log(gdpPercap), data = df)
}
```

```
out_tidy <- gapminder %>%
    group_by(continent, year) %>%
    nest() %>%
    mutate(model = map(data, fit_ols),
           tidied = map(model, tidy)) %>%
    unnest(tidied, .drop = TRUE)
```

map()
tidy()

Nesting as “super-grouping”

```
> out_tidy
# A tibble: 120 x 7
   continent  year term           estimate std.error statistic  p.value
   <fct>     <int> <chr>             <dbl>     <dbl>     <dbl>    <dbl>
 1 Asia       1952 (Intercept)       15.8       9.27      1.71 0.0978
 2 Asia       1952 log(gdpPercap)     4.16      1.25      3.33 0.00228
 3 Asia       1957 (Intercept)       18.1       9.70      1.86 0.0720
 4 Asia       1957 log(gdpPercap)     4.17      1.28      3.26 0.00271
 5 Asia       1962 (Intercept)       16.6       9.52      1.74 0.0911
 6 Asia       1962 log(gdpPercap)     4.59      1.24      3.72 0.000794
 7 Asia       1967 (Intercept)       19.8       9.05      2.19 0.0364
 8 Asia       1967 log(gdpPercap)     4.50      1.15      3.90 0.000477
 9 Asia       1972 (Intercept)       21.9       8.14      2.69 0.0113
10 Asia       1972 log(gdpPercap)     4.44      1.01      4.41 0.000116
# … with 110 more rows
```

Nesting as “super-grouping”

```
out_tidy <- gapminder %>%
    group_by(continent, year) %>%
    nest() %>%
    mutate(model = map(data, fit_ols),
           tidied = map(model, tidy)) %>%
    unnest(tidied, .drop = TRUE) %>%
    filter(term %nin% “(Intercept)"
```

Nesting as “super-grouping”

```
> out_tidy
# A tibble: 60 x 7
   continent  year term              estimate std.error statistic        p.value
   <fct>     <int> <chr>                <dbl>     <dbl>     <dbl>          <dbl>
 1 Asia       1952 log(gdpPercap)        4.16      1.25      3.33 0.00228
 2 Asia       1957 log(gdpPercap)        4.17      1.28      3.26 0.00271
 3 Asia       1962 log(gdpPercap)        4.59      1.24      3.72 0.000794
 4 Asia       1967 log(gdpPercap)        4.50      1.15      3.90 0.000477
 5 Asia       1972 log(gdpPercap)        4.44      1.01      4.41 0.000116
 6 Asia       1977 log(gdpPercap)        4.87      1.03      4.75 0.0000442
 7 Asia       1982 log(gdpPercap)        4.78     0.852      5.61 0.00000377
 8 Asia       1987 log(gdpPercap)        5.17     0.727      7.12 0.0000000531
 9 Asia       1992 log(gdpPercap)        5.09     0.649      7.84 0.00000000760
10 Asia       1997 log(gdpPercap)        5.11     0.628      8.15 0.00000000335
```

~ /> _

Misc dplyr

```
organdata %>%
  select(matches("_lag"))
```

```
A tibble: 238 x 2
   gdp_lag health_lag
     <int>      <dbl>
1    16591       1224
2    16774       1300
3    17171       1379
4    17914       1455
5    18883       1540
6    19849       1626
7    21079       1737
8    21923       1846
9    22961       1948
10   24148       2077
# … with 228 more rows
```

Selection

```
organdata %>%
  select(world, everything())
```

```
# A tibble: 238 x 21
   world country year       donors   pop pop_dens   gdp gdp_lag health health_lag
   <chr> <chr>   <date>      <dbl> <int>    <dbl> <int>   <int>  <dbl>      <dbl>
 1 Libe… Austra… NA           NA    17065    0.220 16774   16591   1300       1224
 2 Libe… Austra… 1991-01-01  12.1   17284    0.223 17171   16774   1379       1300
 3 Libe… Austra… 1992-01-01  12.4   17495    0.226 17914   17171   1455       1379
 4 Libe… Austra… 1993-01-01  12.5   17667    0.228 18883   17914   1540       1455
 5 Libe… Austra… 1994-01-01  10.2   17855    0.231 19849   18883   1626       1540
 6 Libe… Austra… 1995-01-01  10.2   18072    0.233 21079   19849   1737       1626
 7 Libe… Austra… 1996-01-01  10.6   18311    0.237 21923   21079   1846       1737
 8 Libe… Austra… 1997-01-01  10.3   18518    0.239 22961   21923   1948       1846
 9 Libe… Austra… 1998-01-01  10.5   18711    0.242 24148   22961   2077       1948
10 Libe… Austra… 1999-01-01   8.67  18926    0.244 25445   24148   2231       2077
# … with 228 more rows, and 11 more variables: pubhealth <dbl>, roads <dbl>,
#   cerebvas <int>, assault <int>, external <int>, txp_pop <dbl>, opt <chr>,
#   consent_law <chr>, consent_practice <chr>, consistent <chr>, ccode <chr>
```

Selection

```
organdata %>%
  select(world, everything())
```

```
# A tibble: 238 x 21
   world country year       donors   pop pop_dens   gdp gdp_lag health health_lag
   <chr> <chr>   <date>      <dbl> <int>    <dbl> <int>   <int>  <dbl>      <dbl>
 1 Libe… Austra… NA           NA    17065    0.220 16774   16591   1300       1224
 2 Libe… Austra… 1991-01-01  12.1   17284    0.223 17171   16774   1379       1300
 3 Libe… Austra… 1992-01-01  12.4   17495    0.226 17914   17171   1455       1379
 4 Libe… Austra… 1993-01-01  12.5   17667    0.228 18883   17914   1540       1455
 5 Libe… Austra… 1994-01-01  10.2   17855    0.231 19849   18883   1626       1540
 6 Libe… Austra… 1995-01-01  10.2   18072    0.233 21079   19849   1737       1626
 7 Libe… Austra… 1996-01-01  10.6   18311    0.237 21923   21079   1846       1737
 8 Libe… Austra… 1997-01-01  10.3   18518    0.239 22961   21923   1948       1846
 9 Libe… Austra… 1998-01-01  10.5   18711    0.242 24148   22961   2077       1948
10 Libe… Austra… 1999-01-01   8.67  18926    0.244 25445   24148   2231       2077
# … with 228 more rows, and 11 more variables: pubhealth <dbl>, roads <dbl>,
#   cerebvas <int>, assault <int>, external <int>, txp_pop <dbl>, opt <chr>,
#   consent_law <chr>, consent_practice <chr>, consistent <chr>, ccode <chr>
```

Selection

```
organdata %>%
  summarize_all(funs(min, max), na.rm = TRUE)
```

```
organdata %>%
  summarize_if(is.numeric, funs(min, max), na.rm = TRUE)
```

```
organdata %>%
  group_by(country) %>%
  summarize_if(is.numeric, funs(min, max), na.rm = TRUE)
```

Summaries

```
organdata %>%
  filter_all(any_vars(is.na(.)))
```

Scoped Filters

```
iris %>%
     select(contains("Length")) %>%
     rowwise() %>%
     mutate(avg_length =
              mean(c(Petal.Length, Sepal.Length)))
```

```
Source: local data frame [150 x 3]
Groups: <by row>

# A tibble: 150 x 3
   Sepal.Length Petal.Length avg_length
          <dbl>        <dbl>      <dbl>
 1          5.1          1.4       3.25
 2          4.9          1.4       3.15
 3          4.7          1.3          3
 4          4.6          1.5       3.05
 5          5            1.4       3.2
 6          5.4          1.7       3.55
 7          4.6          1.4          3
 8          5            1.5       3.25
 9          4.4          1.4       2.9
10          4.9          1.5       3.2
# … with 140 more rows
```

Row-wise operations