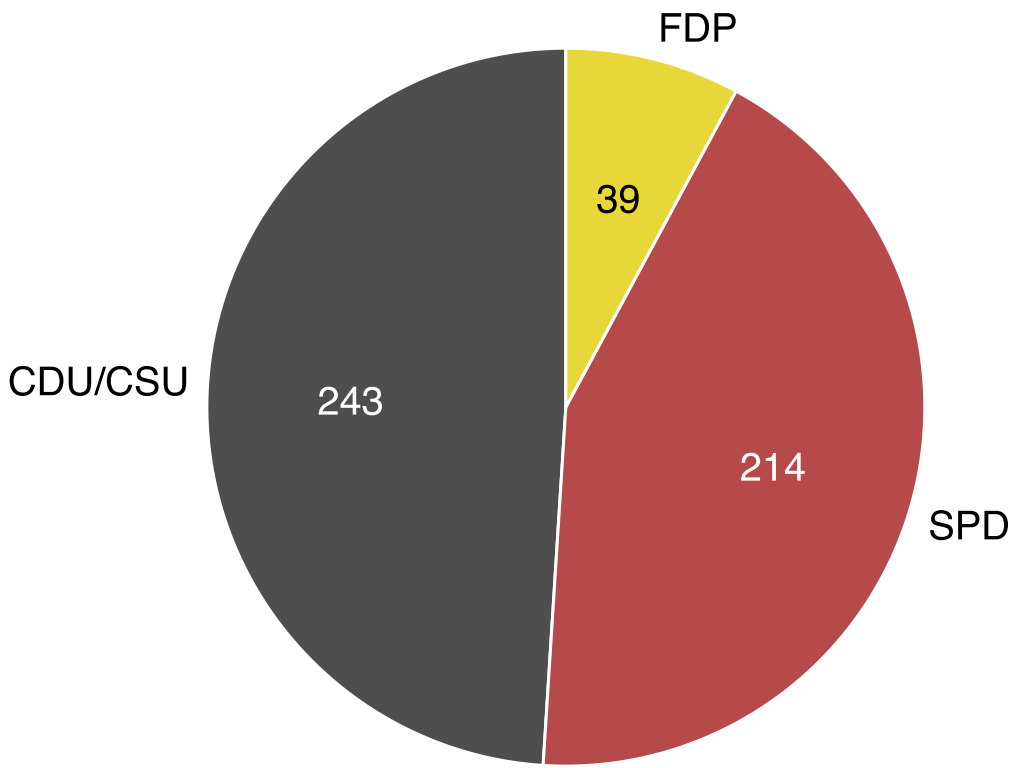
Visualizing proportions

Claus O. Wilke

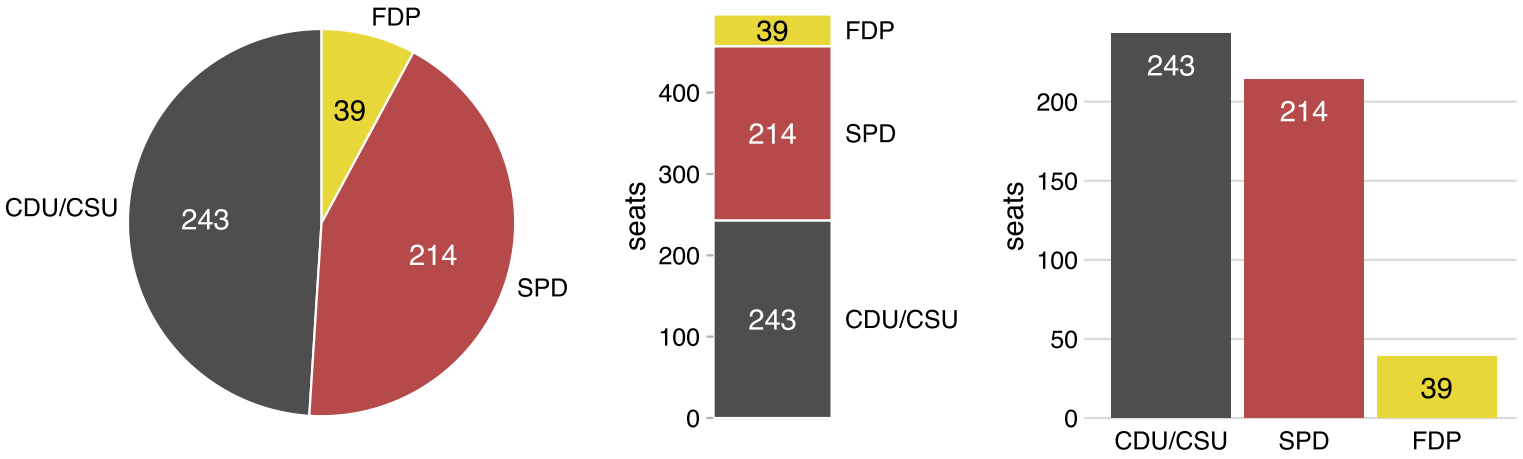
last updated: 2021-03-09

The archetypal visualization of proportions: pie chart



Party composition of the 8th German Bundestag, 1976–1980

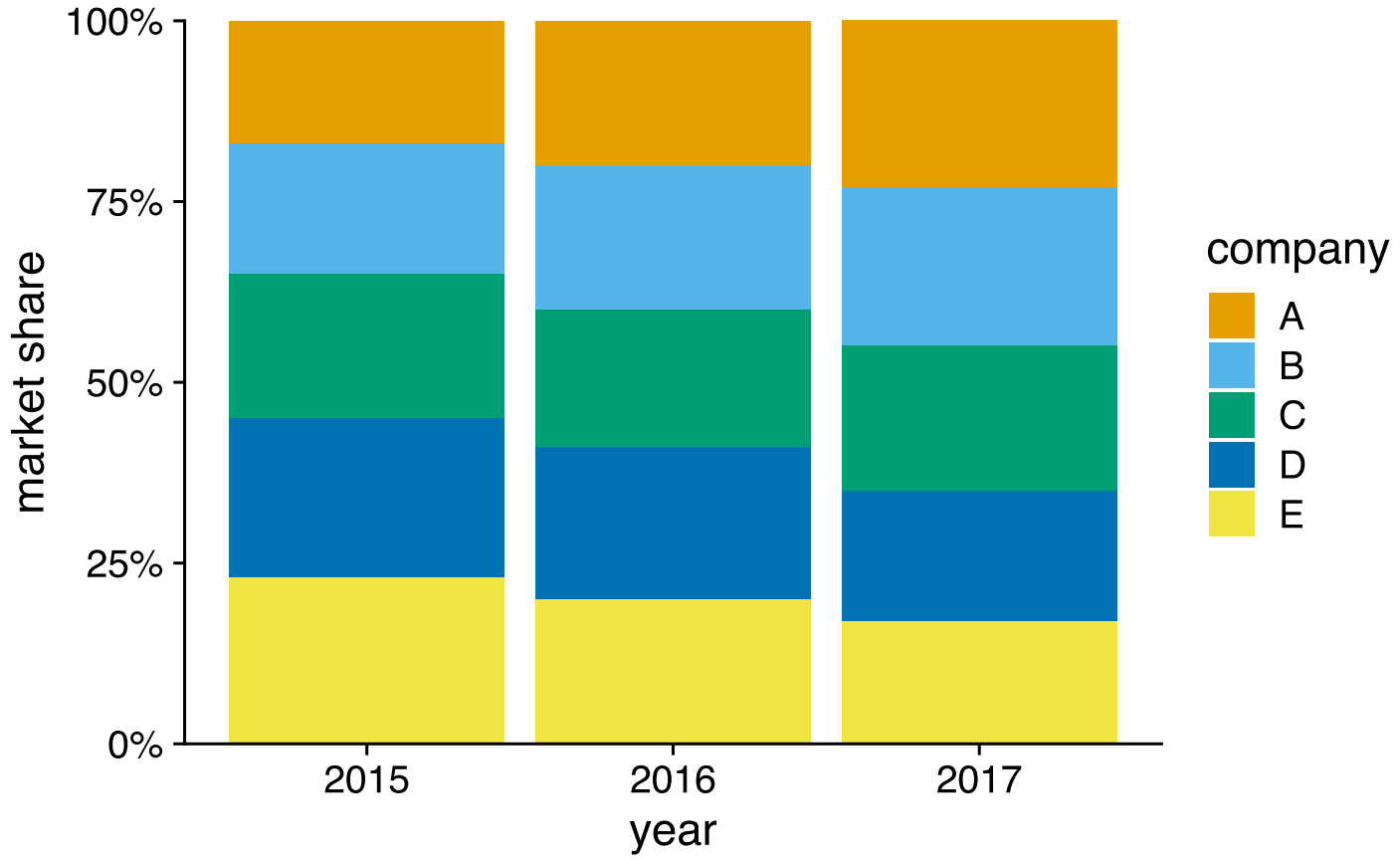
Pie chart vs stacked bars vs side-by-side bars



Example where side-by-side bars are preferred

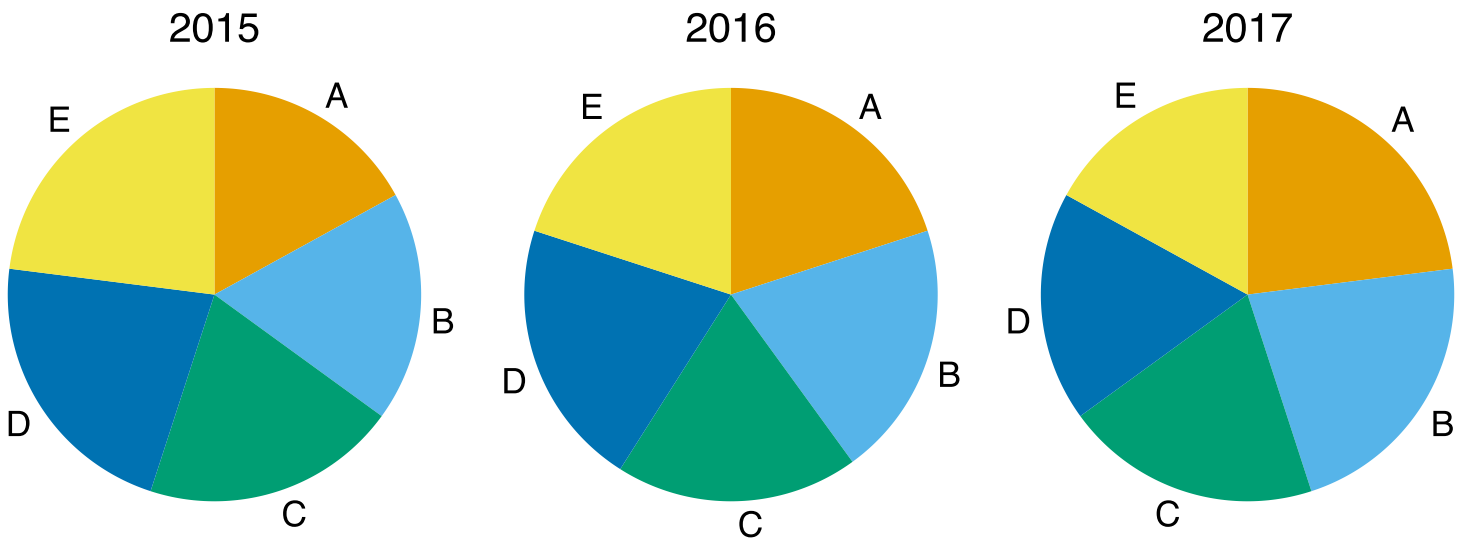
Inspired by: https://en.wikipedia.org/wiki/File:Piecharts.svg

Example where side-by-side bars are preferred



Inspired by: https://en.wikipedia.org/wiki/File:Piecharts.svg

Example where side-by-side bars are preferred

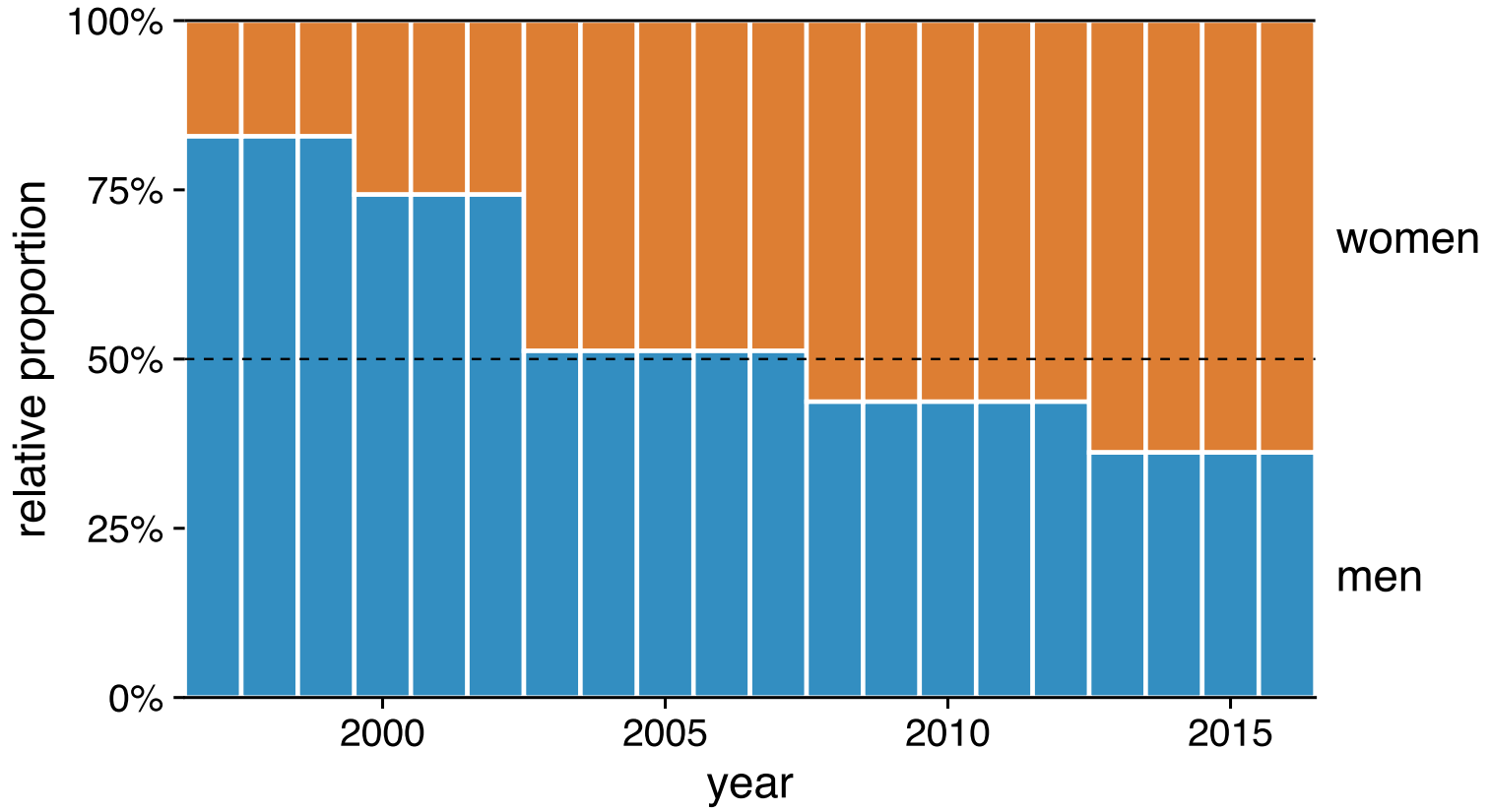


Inspired by: https://en.wikipedia.org/wiki/File:Piecharts.svg

Example where side-by-side bars are preferred

Inspired by: https://en.wikipedia.org/wiki/File:Piecharts.svg

Example where stacked bars are preferred



Change in the gender composition of the Rwandan parliament from 1997 to 2016

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets	✖	✖	✔

Pros and cons of different approaches

	Pie chart	Stacked bars	Side-by-side bars
Allows easy comparison of relative proportions	✖	✖	✔
Shows data as proportions of a whole	✔	✔	✖
Emphasizes simple fractions (1/2, 1/3, ...)	✔	✖	✖
Visually appealing for small datasets	✔	✖	✔
Works well for a large number of subsets	✖	✖	✔
Works well for time series and similar	✖	✔	✖

No one visualization fits all scenarios!

Making pie charts with ggplot

We have three options:

- `geom_bar()`/`geom_col()` with poolar coordinates
  Pros: simple
  Cons: hard to customize

- `geom_arc_bar()` with `stat_pie()`
  Pros: relatively simple, some customization
  Cons: requires **ggforce** & some more complex code

- `geom_arc_bar()` with manual computation
  Pros: maximum flexibility for customization
  Cons: requires **ggforce** & much more complex code

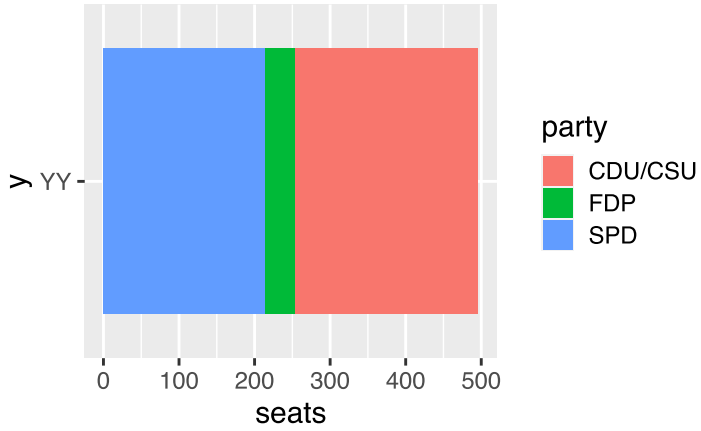
Making pie charts with ggplot: polar coords

```
# the data
bundestag <- tibble(
  party = c("CDU/CSU", "SPD", "FD
  seats = c(243, 214, 39)
)

bundestag
```

```
# A tibble: 3 x 2
  party   seats
  <chr>   <dbl>
1 CDU/CSU   243
2 SPD       214
3 FDP        39
```

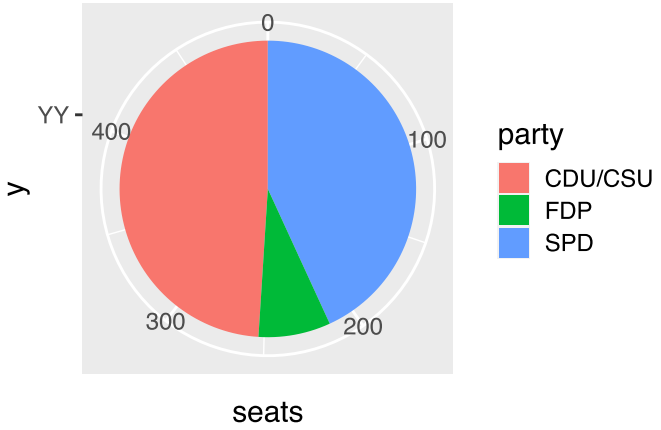
Making pie charts with ggplot: polar coords

```
# the data
bundestag <- tibble(
  party = c("CDU/CSU", "SPD", "FD
  seats = c(243, 214, 39)
)

# make bar chart
ggplot(bundestag) +
  aes(seats, "YY", fill = party)
  geom_col()
```



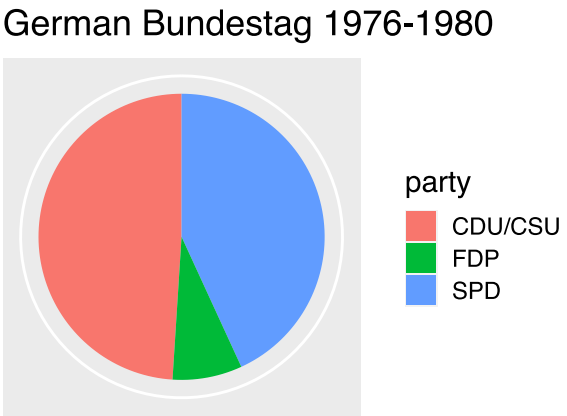
Making pie charts with ggplot: polar coords

```
# the data
bundestag <- tibble(
  party = c("CDU/CSU", "SPD", "FD
  seats = c(243, 214, 39)
)

# make bar chart in polar coords
ggplot(bundestag) +
  aes(seats, "YY", fill = party)
  geom_col() +
  coord_polar()
```



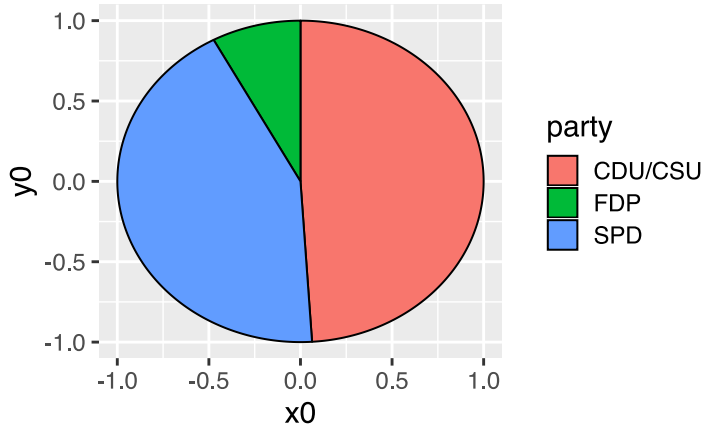
Making pie charts with ggplot: polar coords

```
# the data
bundestag <- tibble(
  party = c("CDU/CSU", "SPD", "FD
  seats = c(243, 214, 39)
)

# make bar chart in polar coords
ggplot(bundestag) +
  aes(seats, "YY", fill = party)
  geom_col() +
  coord_polar() +
  scale_x_continuous(
    name = NULL, breaks = NULL
  ) +
  scale_y_discrete(
    name = NULL, breaks = NULL
  ) +
  ggtitle("German Bundestag 1976-
```



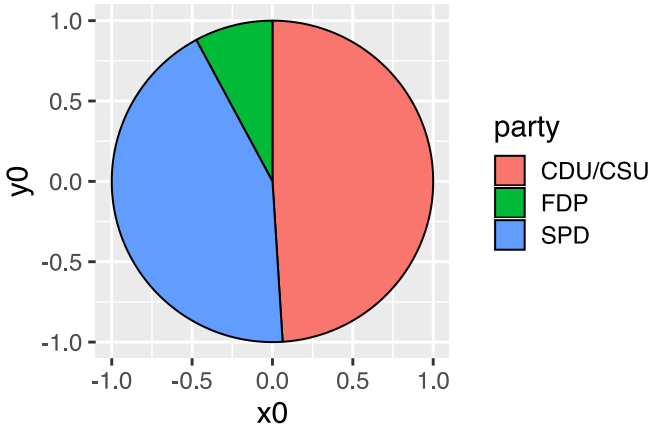
Making pie charts with ggplot: ggforce stat pie

```
library(ggforce)

ggplot(bundestag) +
  aes(
    x0 = 0, y0 = 0, # position of
    r0 = 0, r = 1,  # inner and o
    amount = seats, # size of pie
    fill = party
  ) +
  geom_arc_bar(stat = "pie")
```



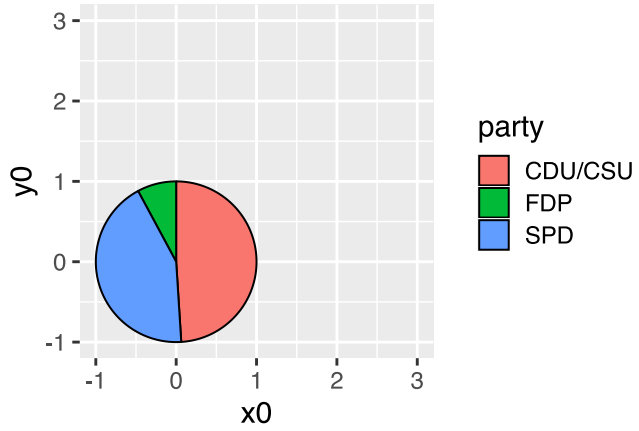
Making pie charts with ggplot: ggforce stat pie

```
library(ggforce)

ggplot(bundestag) +
  aes(
    x0 = 0, y0 = 0, # position of
    r0 = 0, r = 1,  # inner and o
    amount = seats, # size of pie
    fill = party
  ) +
  geom_arc_bar(stat = "pie") +
  coord_fixed()
```



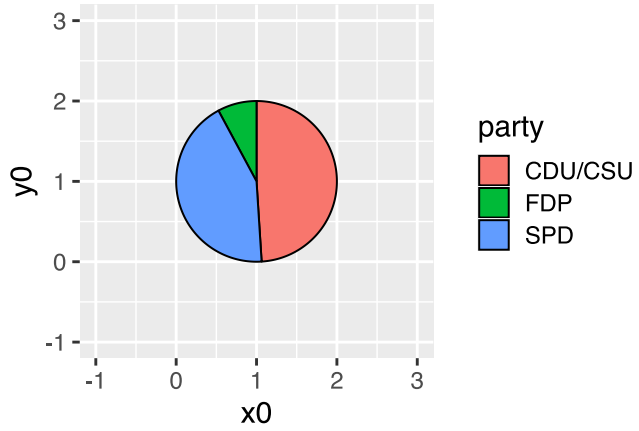
Making pie charts with ggplot: ggforce stat pie

```
library(ggforce)

ggplot(bundestag) +
  aes(
    x0 = 0, y0 = 0, # position of
    r0 = 0, r = 1,  # inner and o
    amount = seats, # size of pie
    fill = party
  ) +
  geom_arc_bar(stat = "pie") +
  coord_fixed(
    xlim = c(-1, 3), ylim = c(-1,
  )
```



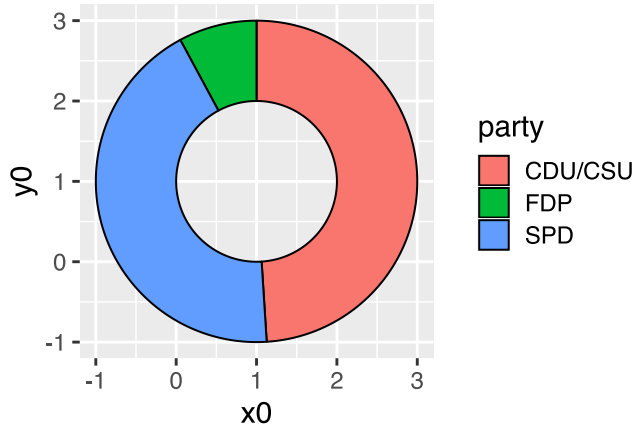
Making pie charts with ggplot: ggforce stat pie

```
library(ggforce)

ggplot(bundestag) +
  aes(
    x0 = 1, y0 = 1, # position of
    r0 = 0, r = 1,  # inner and o
    amount = seats, # size of pie
    fill = party
  ) +
  geom_arc_bar(stat = "pie") +
  coord_fixed(
    xlim = c(-1, 3), ylim = c(-1,
  )
```



Making pie charts with ggplot: ggforce stat pie

```
library(ggforce)

ggplot(bundestag) +
  aes(
    x0 = 1, y0 = 1, # position of
    r0 = 1, r = 2,  # inner and o
    amount = seats, # size of pie
    fill = party
  ) +
  geom_arc_bar(stat = "pie") +
  coord_fixed(
    xlim = c(-1, 3), ylim = c(-1,
  )
```



Making pie charts with ggplot: ggforce manual comp.

```
# prepare pie data
pie_data <- bundestag %>%
  arrange(seats) # sort so pie slices end up sorted

pie_data
```

```
# A tibble: 3 x 2
  party   seats
  <chr>   <dbl>
1 FDP        39
2 SPD       214
3 CDU/CSU   243
```

Making pie charts with ggplot: ggforce manual comp.

```
# prepare pie data
pie_data <- bundestag %>%
  arrange(seats) %>% # sort so pie slices end up sorted
  mutate(
    end_angle = 2*pi*cumsum(seats)/sum(seats)    # ending angle for eacl
  )

pie_data
```

```
# A tibble: 3 x 3
  party   seats end_angle
  <chr>   <dbl>     <dbl>
1 FDP        39     0.494
2 SPD       214     3.20
3 CDU/CSU   243     6.28
```

Making pie charts with ggplot: ggforce manual comp.

```
# prepare pie data
pie_data <- bundestag %>%
  arrange(seats) %>% # sort so pie slices end up sorted
  mutate(
    end_angle = 2*pi*cumsum(seats)/sum(seats),   # ending angle for eacl
    start_angle = lag(end_angle, default = 0)    # starting angle for ea
  )

pie_data
```

```
# A tibble: 3 x 4
  party   seats end_angle start_angle
  <chr>   <dbl>     <dbl>       <dbl>
1 FDP        39     0.494       0
2 SPD       214     3.20        0.494
3 CDU/CSU   243     6.28        3.20
```

Making pie charts with ggplot: ggforce manual comp.

```
# prepare pie data
pie_data <- bundestag %>%
  arrange(seats) %>% # sort so pie slices end up sorted
  mutate(
    end_angle = 2*pi*cumsum(seats)/sum(seats),   # ending angle for eacl
    start_angle = lag(end_angle, default = 0),   # starting angle for ea
    mid_angle = 0.5*(start_angle + end_angle),   # middle of each pie s
  )

pie_data
```

```
# A tibble: 3 x 5
  party   seats end_angle start_angle mid_angle
  <chr>   <dbl>     <dbl>       <dbl>     <dbl>
1 FDP        39     0.494       0         0.247
2 SPD       214     3.20        0.494     1.85
3 CDU/CSU   243     6.28        3.20      4.74
```

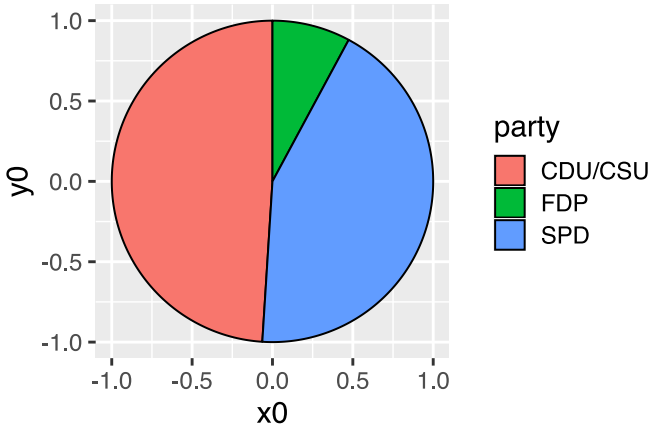
Making pie charts with ggplot: ggforce manual comp.

```
# prepare pie data
pie_data <- bundestag %>%
  arrange(seats) %>% # sort so pie slices end up sorted
  mutate(
    end_angle = 2*pi*cumsum(seats)/sum(seats),   # ending angle for eacl
    start_angle = lag(end_angle, default = 0),   # starting angle for ea
    mid_angle = 0.5*(start_angle + end_angle),   # middle of each pie s
    # horizontal and vertical justifications for outer labels
    hjust = ifelse(mid_angle > pi, 1, 0),
    vjust = ifelse(mid_angle < pi/2 | mid_angle > 3*pi/2, 0, 1)
  )

pie_data
```

```
# A tibble: 3 x 7
  party   seats end_angle start_angle mid_angle hjust vjust
  <chr>   <dbl>     <dbl>       <dbl>     <dbl> <dbl> <dbl>
1 FDP        39     0.494       0         0.247     0     0
2 SPD       214     3.20        0.494     1.85      0     1
3 CDU/CSU   243     6.28        3.20      4.74      1     0
```

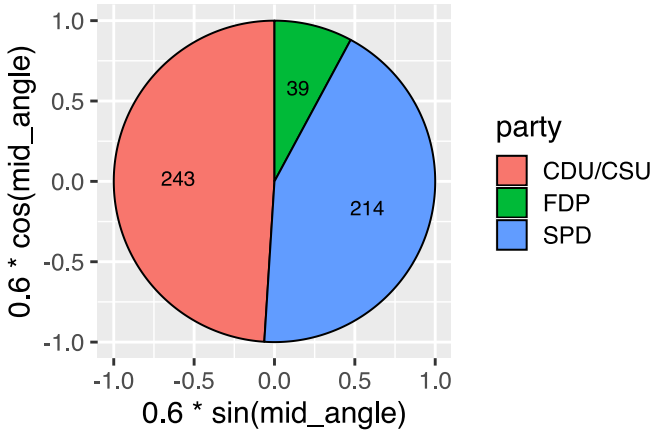
Making pie charts with ggplot: ggforce manual comp.

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1
    start = start_angle, end = en
    fill = party
  ) +
  geom_arc_bar() +
  coord_fixed()
```



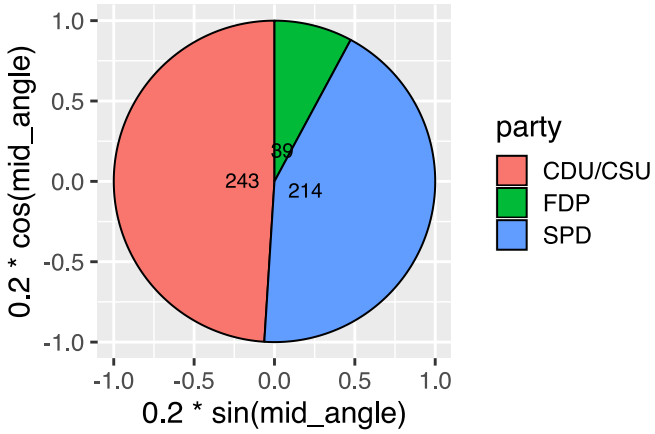
Making pie charts with ggplot: **ggforce manual comp.**

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1
    start = start_angle, end = en
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts insi
    aes(
      x = 0.6 * sin(mid_angle),
      y = 0.6 * cos(mid_angle),
      label = seats
    )
  ) +
  coord_fixed()
```



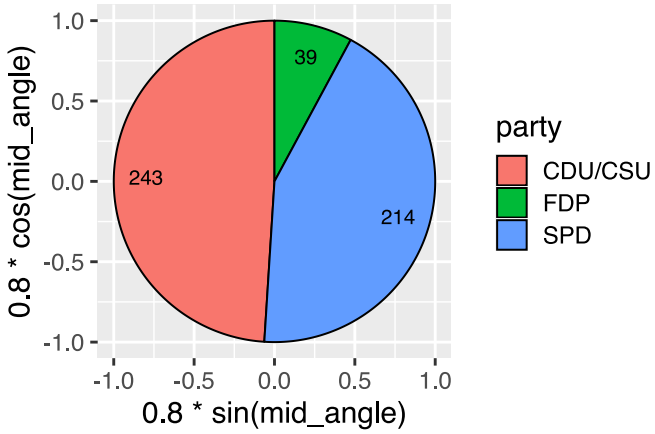
Making pie charts with ggplot: ggforce manual comp.

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1
    start = start_angle, end = en
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts insi
    aes(
      x = 0.2 * sin(mid_angle),
      y = 0.2 * cos(mid_angle),
      label = seats
    )
  ) +
  coord_fixed()
```



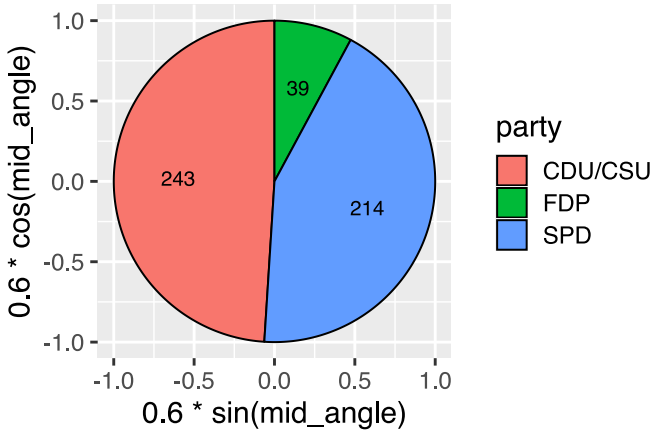
Making pie charts with ggplot: **ggforce manual comp.**

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1
    start = start_angle, end = en
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts insi
    aes(
      x = 0.8 * sin(mid_angle),
      y = 0.8 * cos(mid_angle),
      label = seats
    )
  ) +
  coord_fixed()
```



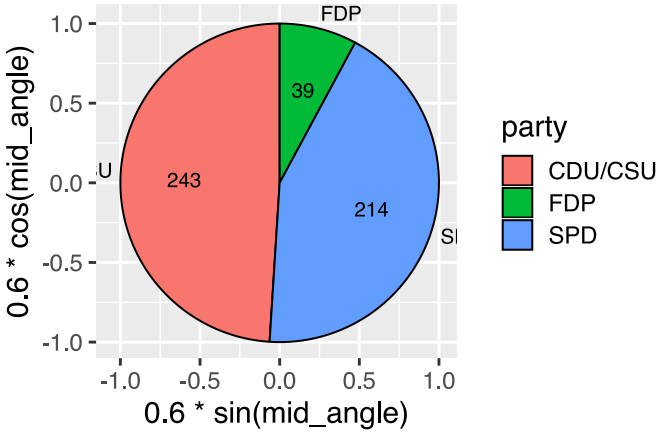
Making pie charts with ggplot: **ggforce manual comp.**

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1
    start = start_angle, end = en
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts insi
    aes(
      x = 0.6 * sin(mid_angle),
      y = 0.6 * cos(mid_angle),
      label = seats
    )
  ) +
  coord_fixed()
```



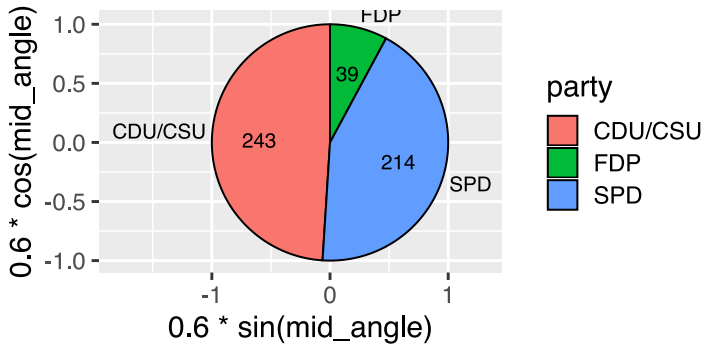
Making pie charts with ggplot: ggforce manual comp.

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1,
    start = start_angle, end = end_
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts inside
    aes(
      x = 0.6 * sin(mid_angle),
      y = 0.6 * cos(mid_angle),
      label = seats
    )
  ) +
  geom_text( # place party name out
    aes(
      x = 1.05 * sin(mid_angle),
      y = 1.05 * cos(mid_angle),
      label = party,
      hjust = hjust, vjust = vjust
    )
```



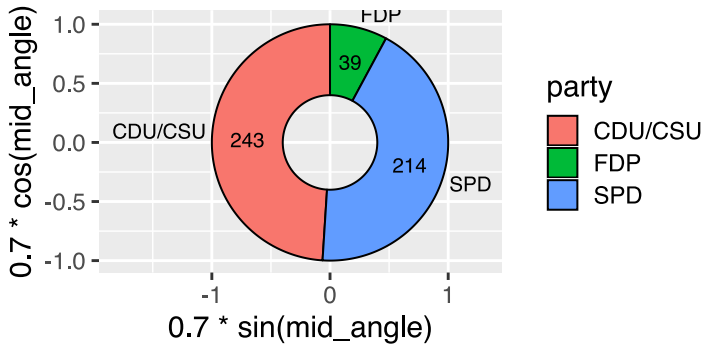
Making pie charts with ggplot: ggforce manual comp.

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0, r = 1,
    start = start_angle, end = end_
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts inside
    aes(
      x = 0.6 * sin(mid_angle),
      y = 0.6 * cos(mid_angle),
      label = seats
    )
  ) +
  geom_text( # place party name out
    aes(
      x = 1.05 * sin(mid_angle),
      y = 1.05 * cos(mid_angle),
      label = party,
      hjust = hjust, vjust = vjust
    )
```



Making pie charts with ggplot: ggforce manual comp.

```
ggplot(pie_data) +
  aes(
    x0 = 0, y0 = 0, r0 = 0.4, r = 1
    start = start_angle, end = end_
    fill = party
  ) +
  geom_arc_bar() +
  geom_text( # place amounts inside
    aes(
      x = 0.7 * sin(mid_angle),
      y = 0.7 * cos(mid_angle),
      label = seats
    )
  ) +
  geom_text( # place party name out
    aes(
      x = 1.05 * sin(mid_angle),
      y = 1.05 * cos(mid_angle),
      label = party,
      hjust = hjust, vjust = vjust
    )
```



Further reading

- Fundamentals of Data Visualization: Chapter 10: Visualizing proportions
- Fundamentals of Data Visualization: Chapter 11: Visualizing nested proportions
- **ggplot2** reference documentation: `position_stack()`, `position_fill()`
- **ggplot2** reference documentation: `position_dodge()`
- **ggforce** reference documentation: `geom_arc_bar()`