

ODSC: ggplot2 Graph Gallery

Categories and distributions: amounts

by Martin Frigaard & Peter Spangler

Written: September 21 2021

Updated: April 03 2022

Resources :



The graphs

- **The ggplot2 book** by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen
- **Data Visualization: A Practical Introduction** by Kieran Healy (2018)
- **R Graphics Cookbook, 2nd edition** by Winston Chang (2022)

Graph Categories

- **Fundamentals of Data Visualization** by Claus O. Wilke (2019)
- **Data Visualisation: A Handbook for Data Driven Design** by Andy Kirk (2019)
- **Data Points** by Nathan Yau (2013)

Graph Categories: *The 'CHRTS' Families of Chart Types*



From *"Data Visualisation: A Handbook for Data Driven Design"*, Andy Kirk (2019)

Comparing categories and distributions

Hierarchies/part-to-whole relationships

Correlations and connections

Trends and intervals over time

Maps, overlays, and/or distortions

Graph Categories: Directory of Visualizations



From *"Fundamentals of Data Visualization"*, Claus O. Wilke (2019)

Amounts

Distributions

Proportions

X-Y relationships

Geospatial Data

Uncertainty

Comprehensive Graph Gallery



Comparing categories and values

- *Amounts*
- Distributions

Hierarchies and proportions

- Part-to-whole relationships

Trends, correlations and connections

- X-Y relationships

Maps, overlays, and distortions

- Geospatial Data

Statistical measures

- Uncertainty

Data



Data come from the following packages:

- **palmerpenguins**
- **fivethirtyeight**
- **ggplot2movies**

Or created using **tribble()**

```
tribble(  
  ~`variable 1`, ~`variable 2`,  
    "a",          1,  
    "b",          2,  
    "c",          3)
```

variable 1	variable 2
<chr>	<dbl>
a	1
b	2
c	3
3 rows	

Load data packages



```
library(palmerpenguins)  
library(fivethirtyeight)  
library(ggplot2movies)
```

palmerpenguins



palmerpenguins package website

```
palmerpenguins::penguins -> penguins
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

1-10 of 344 rows

Previous **1** 2 3 4 5 6 ... 35 Next

fivethirtyeight



fivethirtyeight package website

All datasets are listed below with descriptions

```
datasets("fivethirtyeight")
```

dataset

<chr>

US_births_1994_2003

US_births_2000_2014

ahca_polls

airline_safety

antiquities_act

august_senate_polls

avengers

bachelorette

bad_drivers

bechdel

1-10 of 129 rows | 1-1 of 2 columns

Previous **1** 2 3 4 5 6 ... 13 Next

ggplot2movies



ggplot2movies package website

We're using `movies_data` (derived version of the `ggplot2movies::movies`)

movies_data

title	year	length	budget	rating	mpaa
<chr>	<int>	<int>	<int>	<dbl>	<fct>
100 Mile Rule	2002	98	1100000	5.6	R
13 Going On 30	2004	98	37000000	6.4	PG-13
15 Minutes	2001	120	42000000	6.1	R
2 Fast 2 Furious	2003	107	76000000	5.1	PG-13
2046	2004	129	12000000	7.6	R
21 Grams	2003	124	20000000	8.0	R
25th Hour	2002	135	15000000	7.8	R
3000 Miles to Graceland	2001	125	62000000	5.4	R
40 Days and 40 Nights	2002	96	17000000	5.4	R
50 First Dates	2004	99	75000000	6.8	PG-13

1-10 of 751 rows | 1-6 of 7 columns

Previous **1** 2 3 4 5 6 ... 76 Next

Comparing Categories and Distributions



Amounts

Amounts: *Bars*



The bar chart (or graph) is typically used to display counts. Bar charts can be arranged vertically or horizontally, stacked, diverging, or dodged. In `ggplot2`, bar charts can be built using `geom_bar()` or `geom_col()`

Amounts: *Bars*



movies_data

title	year	length	budget	rating	mpaa
<chr>	<int>	<int>	<int>	<dbl>	<fct>
100 Mile Rule	2002	98	1100000	5.6	R
13 Going On 30	2004	98	37000000	6.4	PG-13
15 Minutes	2001	120	42000000	6.1	R
2 Fast 2 Furious	2003	107	76000000	5.1	PG-13
2046	2004	129	12000000	7.6	R
21 Grams	2003	124	20000000	8.0	R
25th Hour	2002	135	15000000	7.8	R
3000 Miles to Graceland	2001	125	62000000	5.4	R
40 Days and 40 Nights	2002	96	17000000	5.4	R
50 First Dates	2004	99	75000000	6.8	PG-13

1-10 of 751 rows | 1-6 of 7 columns

Previous **1** 2 3 4 5 6 ... 76 Next

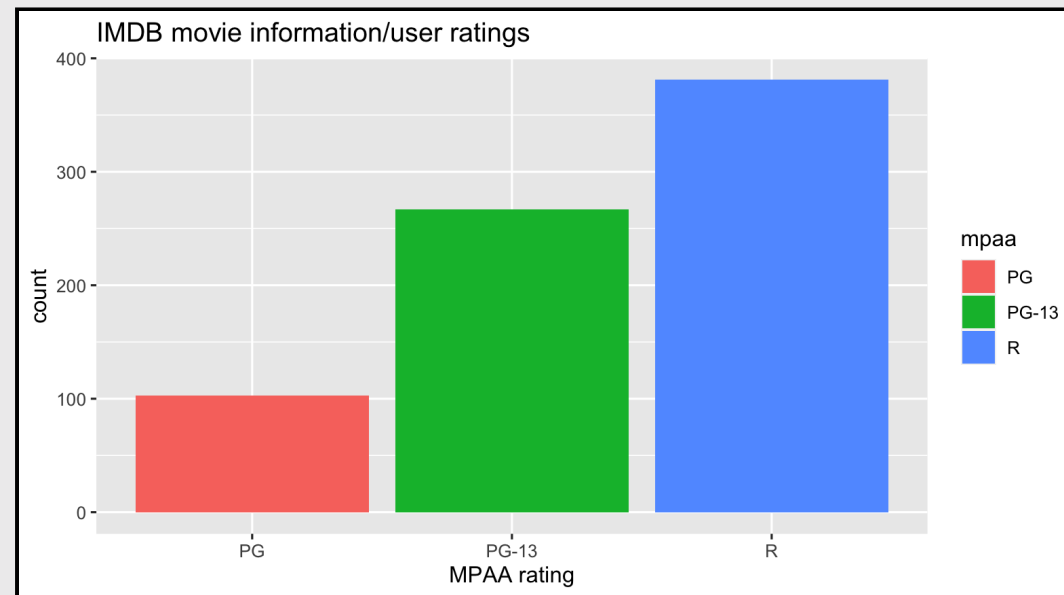
Amounts: *Bars*



Map *mpaa* to the *x* axis and to the *fill* aesthetic inside the *aes()* of *geom_bar()*, and add the labels

```
labs_geom_bar <- labs(  
  x = "MPAA rating",  
  title = "IMDB movie information/user ratings")
```

```
ggplot(data = movies_data,  
  aes(x = mpaa)) +  
  geom_bar(aes(fill = mpaa)) +  
  labs_geom_bar
```



Amounts: *Grouped Bars*



To create grouped bar charts (compare the values of a numerical variable across the levels of a categorical variable) we can use the `geom_col()` function.

Amounts: *Grouped Bars*



movies_data

title	year	length	budget	rating	mpaa
<chr>	<int>	<int>	<int>	<dbl>	<fct>
100 Mile Rule	2002	98	1100000	5.6	R
13 Going On 30	2004	98	37000000	6.4	PG-13
15 Minutes	2001	120	42000000	6.1	R
2 Fast 2 Furious	2003	107	76000000	5.1	PG-13
2046	2004	129	12000000	7.6	R
21 Grams	2003	124	20000000	8.0	R
25th Hour	2002	135	15000000	7.8	R
3000 Miles to Graceland	2001	125	62000000	5.4	R
40 Days and 40 Nights	2002	96	17000000	5.4	R
50 First Dates	2004	99	75000000	6.8	PG-13

1-10 of 751 rows | 1-6 of 7 columns

Previous **1** 2 3 4 5 6 ... 76 Next

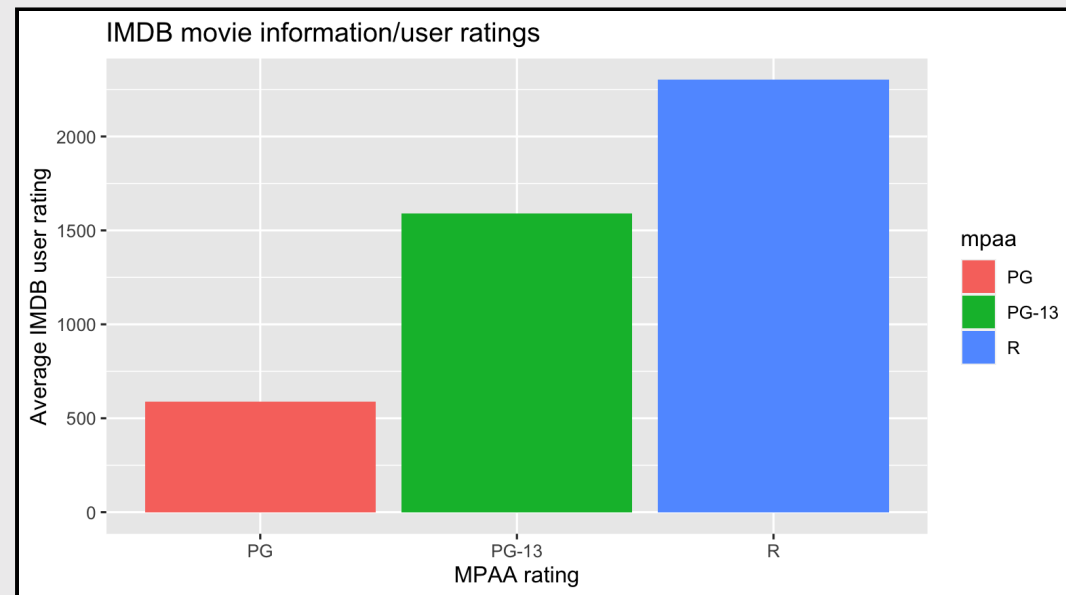
Amounts: *Grouped Bars*



Map *mpaa* to the *x* axis, *rating* to the *y* axis, and *mpaa* to *fill* inside the *aes()* of *geom_col()*, and add the labels

```
labs_geom_col <- labs(  
  x = "MPAA rating",  
  y = "Average IMDB user rating",  
  title = "IMDB movie information/user ratings")
```

```
ggplot(data = movies_data,  
       aes(x = mpaa,  
           y = rating)) +  
  geom_col(aes(fill = mpaa)) +  
  labs_geom_col
```



Amounts: *Stacked Bars*



We can also use bars to look at numeric and categorical variables using `geom_bar()` by setting `fill` argument.

Amounts: *Stacked Bars*



penguins

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

1-10 of 344 rows

Previous 1 2 3 4 5 6 ... 35 Next

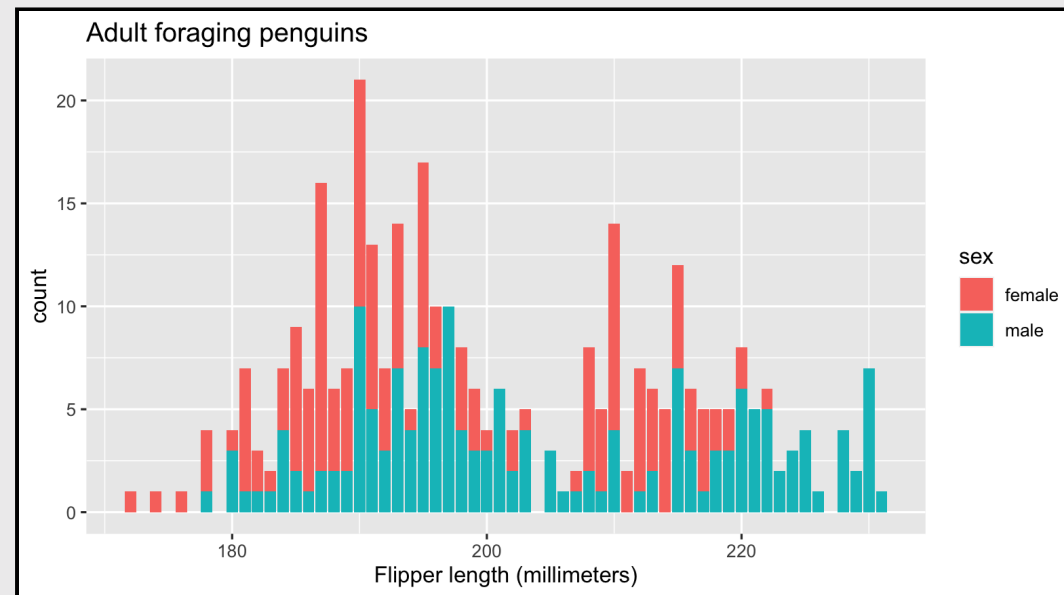
Amounts: *Stacked Bars*



Map *flipper_length_mm* to the *x* axis, *sex* to *fill*, the *geom_bar()* layer, and add the labels

```
labs_geom_bar_stacked <- labs(  
  x = "Flipper length (millimeters)",  
  title = "Adult foraging penguins")
```

```
# remove missing sex  
penguins_stacked <- filter(penguins,  
  !is.na(sex))  
ggplot(data = penguins_stacked,  
  aes(x = flipper_length_mm,  
    fill = sex)) +  
  geom_bar() +  
  labs_geom_bar_stacked
```



Amounts: *Stacked Bars*



We can extend `geom_bar()` by setting the `y` to a numeric variable and using both the `x` and `fill` aesthetics (two categorical variables).

Amounts: *Stacked Bars*



penguins

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
<fct>	<fct>	<dbl>	<dbl>	<int>	<int>	<fct>	<int>
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007
Adelie	Torgersen	38.9	17.8	181	3625	female	2007
Adelie	Torgersen	39.2	19.6	195	4675	male	2007
Adelie	Torgersen	34.1	18.1	193	3475	NA	2007
Adelie	Torgersen	42.0	20.2	190	4250	NA	2007

1-10 of 344 rows

Previous 1 2 3 4 5 6 ... 35 Next

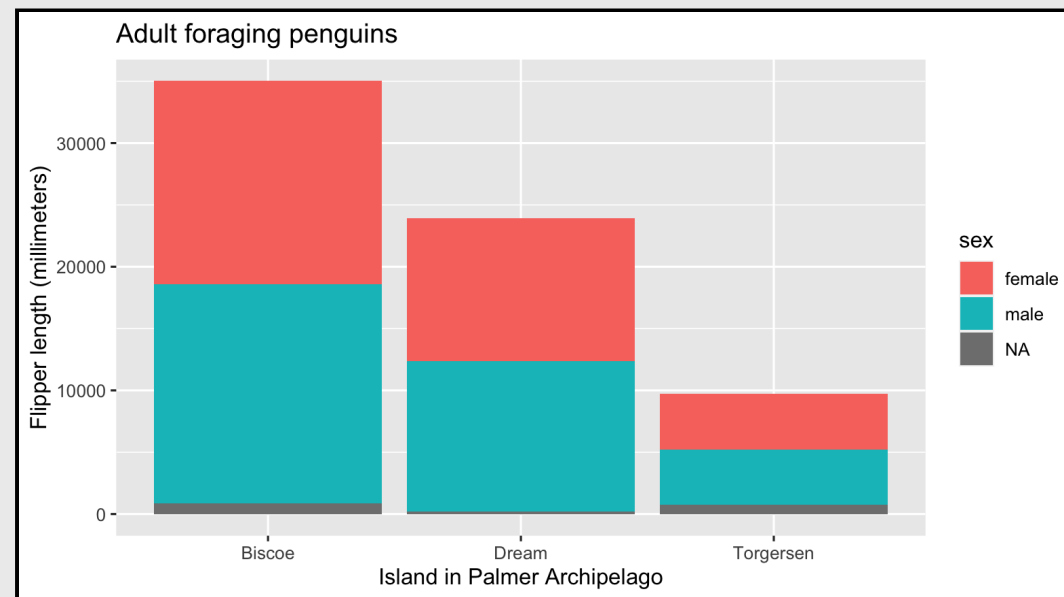
Amounts: *Stacked Bars*



Map *island* to the *x* axis, *flipper_length_mm* to the *y* axis, *sex* to *fill*, the *geom_bar()* layer (with *position* and *stat*), and add the labels

```
geom_bar_stacked_2 <- labs(  
  x = "Island in Palmer Archipelago",  
  y = "Flipper length (millimeters)",  
  title = "Adult foraging penguins")
```

```
ggplot(data = penguins,  
  aes(x = island,  
    y = flipper_length_mm,  
    fill = sex)) +  
  # use this to determine how many  
  # sex values are NA (and in what  
  # categories)  
  geom_bar(position = "stack",  
    stat = "identity") +  
  geom_bar_stacked_2
```



Amounts: *Diverging Bars*



If you have a numeric variable with positive and negative values, consider using diverging bars with `geom_bar()`

Amounts: *Diverging Bars*



```
unisex_names <- fivethirtyeight::unisex_names
unisex_names_diff <- mutate(unisex_names,
  male_female_diff = male_share - female_share,
  diff_cat = if_else(
    male_female_diff > 0,
    true = "More common male name",
    false = "More common female name"))
sample_names <- slice_sample(unisex_names_diff, n = 10)
```

name	total	male_share	female_share
<chr>	<dbl>	<dbl>	<dbl>
Haydin	572.9465	0.6200763	0.3799237
Cailen	365.5456	0.5149821	0.4850179
Deovion	113.0031	0.6227033	0.3772967
Mahari	277.9109	0.5637120	0.4362880
Harley	41237.5657	0.5717018	0.4282982
Lauris	131.2414	0.4732027	0.5267973
Kaydyn	323.8716	0.4721678	0.5278322
Embry	393.5799	0.4896606	0.5103394
Jamille	673.4794	0.3652454	0.6347546
Codie	3998.8170	0.6087125	0.3912875

1-10 of 10 rows | 1-4 of 7 columns

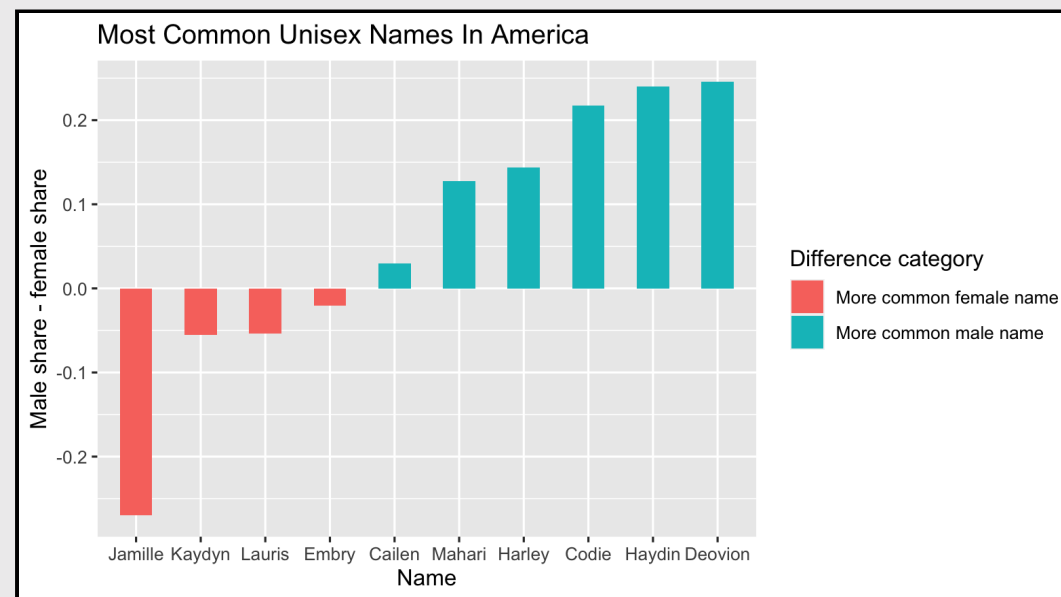
Amounts: *Diverging Bars*



Here we use the `reorder()` function to arrange the values of `male_female_diff` by `name`, and map the `diff_cat` to `label`.

```
labs_geom_bar_diverg <- labs(  
  x = "Name",  
  y = "Male share - female share",  
  title = "Most Common Unisex Names In America",  
  fill = "Difference category")
```

```
ggplot(data = sample_names,  
  aes(x = reorder(x = name,  
    male_female_diff),  
    # reorder this by x  
    y = male_female_diff,  
    label = diff_cat)) +  
  geom_bar(  
    aes(fill = diff_cat),  
    stat = "identity",  
    width = .5) +  
  labs_geom_bar_diverg
```



Amounts: *Diverging Bars (vertical)*



```
unisex_names <- fivethirtyeight::unisex_names
unisex_names_diff <- mutate(unisex_names,
  male_female_diff = male_share - female_share,
  diff_cat = if_else(male_female_diff > 0,
    true = "More common male name",
    false = "More common female name"))
sample_names <- slice_sample(unisex_names_diff, n = 20)
```

name	total	male_share	female_share
<chr>	<dbl>	<dbl>	<dbl>
Hyun	415.3097	0.6056871	0.3943129
Schylar	326.6207	0.3355031	0.6644969
Seville	125.2281	0.3883259	0.6116741
Asheton	169.1575	0.5442700	0.4557300
Teigan	714.6648	0.4235426	0.5764574
Micaiah	2749.9350	0.6204676	0.3795324
Kriston	1194.6709	0.4107587	0.5892413
Aalijah	227.3348	0.5851041	0.4148959
Christan	2684.9788	0.5011373	0.4988627
Lonie	755.8686	0.4401145	0.5598855

1-10 of 20 rows | 1-4 of 7 columns

Previous **1** 2 Next

Amounts: *Diverging Bars (vertical)*



Diverging bar-charts can be arranged vertically, too

Amounts: *Diverging Bars (vertical)*



For vertically arranged bars, we switch the *x* and *y* axis variables (and the *reorder()* function).

```
labs_geom_bar_diverg_vert <- labs(  
  x = "Name",  
  y = "Male share - female share",  
  title = "Most Common Unisex Names In America",  
  fill = "Difference category")
```

```
ggplot(data = sample_names,  
  aes(x = male_female_diff,  
    # reorder this by x  
    y = reorder(x = name,  
      male_female_diff),  
    label = diff_cat)) +  
  geom_bar(  
    aes(fill = diff_cat),  
    stat = "identity",  
    width = .5) +  
  labs_geom_bar_diverg_vert
```

