IP-PAD Data Visualization Workshop

Amsterdam Meeting – Jakob Kasper & Olaf Borghi

Why do we Visualise Data?

- Accurately showcase and make sense of information

Why do we Visualise Data?

- Accurately showcase and make sense of information

→ **Table?**

Table 2: Mean values and differences in means for amount donated in "crackdown" (treatment) and "no crackdown" (control) conditions; values represent posterior medians

H_{1b}	$Amount_{Treatment}$	$Amount_{Control}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Crackdown − No crackdown	16.34	12.93	3.39	26.3%	0.97
Humanitarian assistance − Human rights	14.06	14.85	-0.82	-5.5%	0.67
Private − Government funding	15.13	13.71	1.42	10.4%	0.79
H_{2b} and H_{3b}	$Amount_{Crackdown}$	$Amount_{No\ crackdown}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Human rights issues	17.4	14.86	2.54	17.2%	0.83
Humanitarian assistance issues	15.91	11.68	4.3	36.9%	0.95
Government funding	13.83	12.24	1.61	13.1%	0.74
Private funding	18.95	14.23	4.62	32.4%	0.97
H_{2b} and H_{3b} (nested)	$Amount_{Crackdown}$	$Amount_{No\ crackdown}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Human rights issues, Government funding	10.56	15.15	-4.46	-29.5%	0.91
Human rights issues, Private funding	23.76	14.5	9.19	63.8%	0.99
Humanitarian assistance issues, Government funding	21.42	11.89	9.35	77.9%	0.99
Humanitarian assistance issues, Private funding	15.69	15.72	-0.05	-0.3%	0.51

Source: https://datavizf25.classes.andrewheiss.com

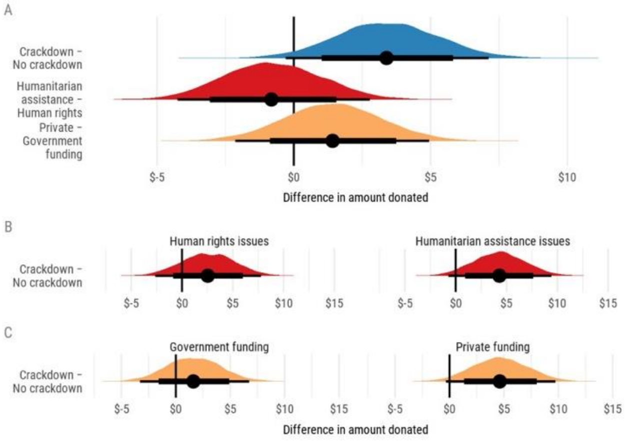
Why do we Visualise Data?

- Accurately showcase and make sense of information

→ **Accessibly** & **Aesthetically**

Table 2: Mean values and differences in means for amount donated in "crackdown" (treatment) and "no crackdown" (control) conditions; values represent posterior medians

H_{1b}	$Amount_{Treatment}$	$Amount_{Control}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Crackdown – No crackdown	16.34	12.93	3.39	26.3%	0.97
Humanitarian assistance – Human rights	14.06	14.85	-0.82	-5.5%	0.67
Private – Government funding	15.13	13.71	1.42	10.4%	0.79
H_{2b} and H_{3b}	$Amount_{Crackdown}$	$Amount_{No\ crackdown}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Human rights issues	17.4	14.86	2.54	17.2%	0.83
Humanitarian assistance issues	15.91	11.68	4.3	36.9%	0.95
Government funding	13.83	12.24	1.61	13.1%	0.74
Private funding	18.95	14.23	4.62	32.4%	0.97
H_{2b} and H_{3b} (nested)	$Amount_{Crackdown}$	$Amount_{No\ crackdown}$	Δ	$\%\Delta$	$p(\Delta \neq 0)$
Human rights issues, Government funding	10.56	15.15	-4.46	-29.5%	0.91
Human rights issues, Private funding	23.76	14.5	9.19	63.8%	0.99
Humanitarian assistance issues, Government funding	21.42	11.89	9.35	77.9%	0.99
Humanitarian assistance issues, Private funding	15.69	15.72	-0.05	-0.3%	0.51



Source: https://datavizf25.classes.andrewheiss.com

Data visualization can help at many stages

Data Cleaning

E.g., explore number of NA responses

Data Descriptives

E.g., show distribution of Likert Scale in subgroups

Model Fitting (regression…)

E.g., investigate model fit

Model interpretation

E.g., show & make sense of results

Agenda

1. Types of Software & Tools
2. Introduction to ggplot2
3. Common Plots in the Social and Cognitive Sciences
4. Good and Bad Practices
5. Useful Extensions to ggplot2

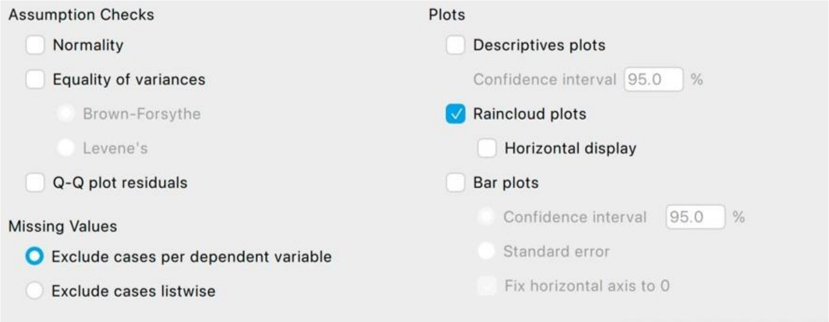
1. Types of Software & Tools

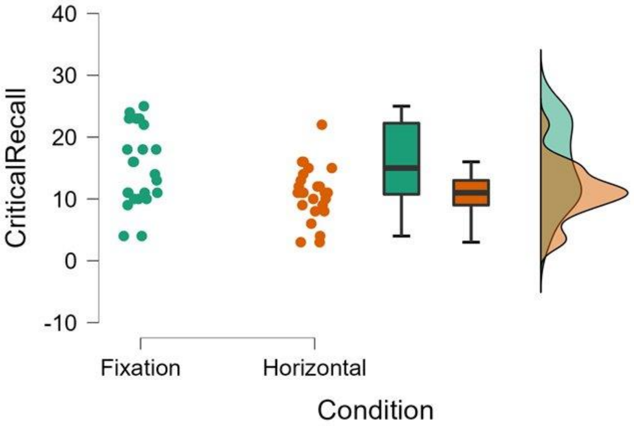
Free & Open-Source Software

- JASP: Point & click statistics program
- Python: High-level, general-purpose programming language
  - Several packages for data analysis & visualization
  - Powerful for neuroscience visualization
- **R + tidyverse:** Programming language for statistical computing and data visualization

JASP https://jasp-stats.org

- Powerful statistical software
  - From descriptives to Bayesian models, meta-analysis, SEM
- Beautiful visualisations in one click





European Commission

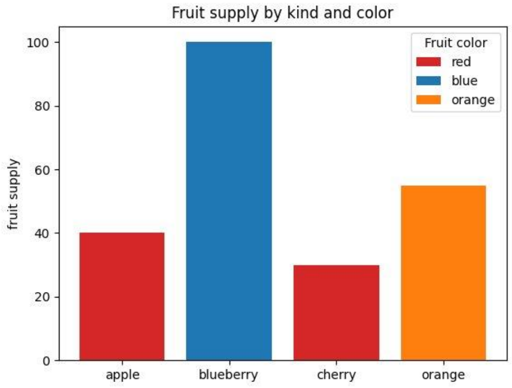
UK Research and Innovation

Python https://python.org

- General purpose programming language
  - Specific packages for data analysis & visualisation

matplotlib

```python
import matplotlib.pyplot as plt

fig, ax = plt.subplots()

fruits = ['apple', 'blueberry', 'cherry', 'orange']
counts = [40, 100, 30, 55]
bar_labels = ['red', 'blue', '_red', 'orange']
bar_colors = ['tab:red', 'tab:blue', 'tab:red', 'tab:orange']

ax.bar(fruits, counts, label=bar_labels, color=bar_colors)

ax.set_ylabel('fruit supply')
ax.set_title('Fruit supply by kind and color')
ax.legend(title='Fruit color')

plt.show()
```

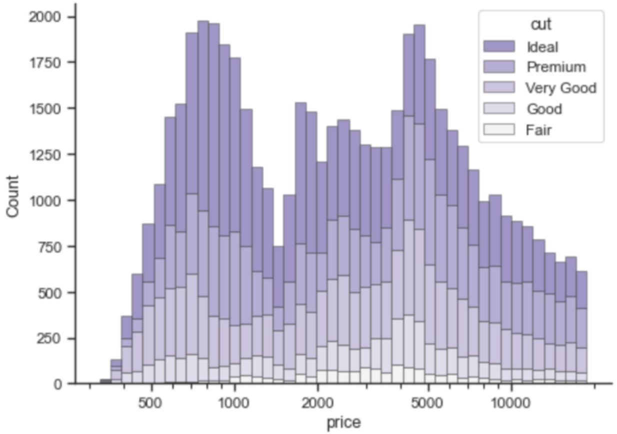


Python https://python.org

- General purpose programming language
  - Specific packages for data analysis & visualisation

seaborn

```
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt

sns.set_theme(style="ticks")

diamonds = sns.load_dataset("diamonds")

f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)

sns.histplot(
    diamonds,
    x="price", hue="cut",
    multiple="stack",
    palette="light:m_r",
    edgecolor=".3",
    linewidth=.5,
    log_scale=True,
)
ax.xaxis.set_major_formatter(mpl.ticker.ScalarFormatter())
ax.set_xticks([500, 1000, 2000, 5000, 10000])
```



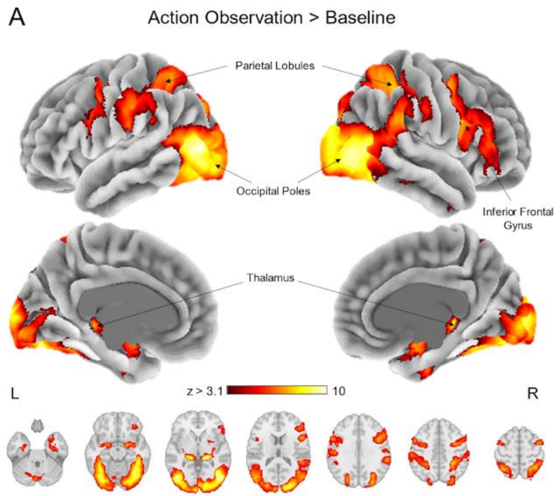
Python https://python.org

- General purpose programming language
    - Specific packages for data analysis & visualisation

nilearn

https://nilearn.github.io/dev/index.html

- Analysis and Visualisation of fMRI & MRI
- nilearn.plotting.plot_surf()



Our focus: R + tidyverse

Why R?

- Purpose-built for data analysis
- Open-source + actively maintained
- Massive community + packages

Why tidyverse?

- Seamlessly integrated packages
- Readable syntax across tasks
- Makes complex tasks simple

+

+

```
install.packages(“tidyverse”)
library(tidyverse)
```

2. Introduction to ggplot2

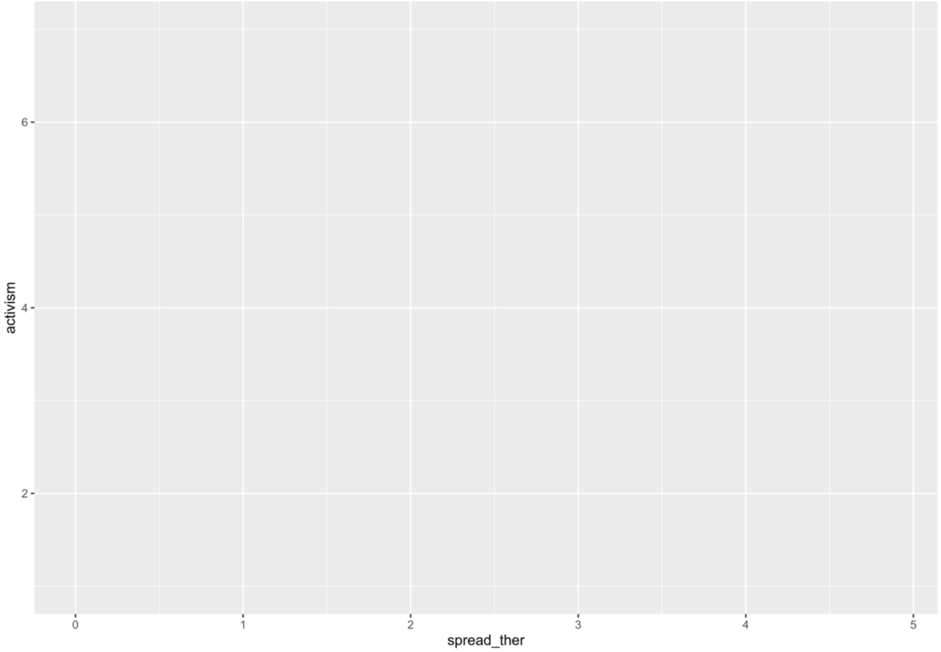
Introduction to ggplot2

- "ggplot" = "Grammar of Graphics plot 2"
- ***Grammar of Graphics*** = theory by Leland Wilkinson which provides a framework for visualizations based on the principles of layers, aesthetics, and mappings
- Every ggplot2 plot has three key components:
  1. Data
  2. Aesthetic mappings (variables & visual properties)
  3. At least one layer (how to render each observation)

European Commission
UK Research and Innovation

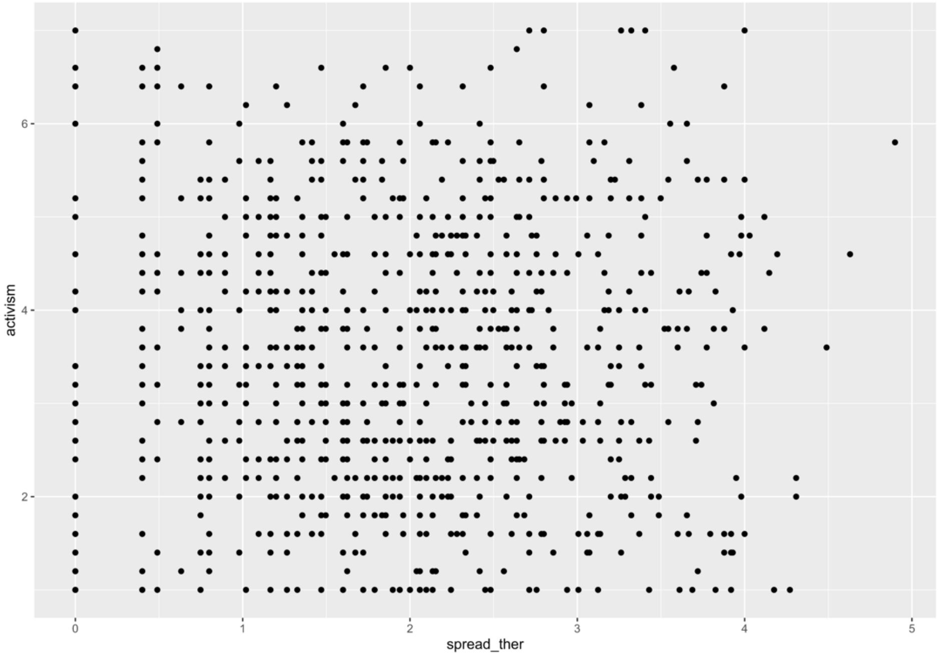
Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism))
```



Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism))+
  geom_point()
```

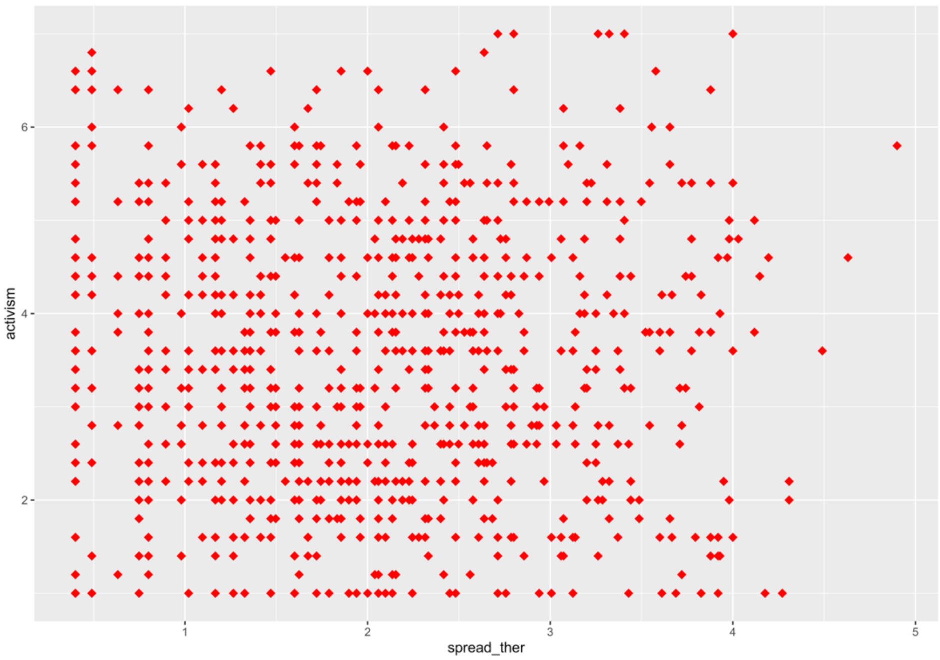


Aesthetics, Mapping, & Layers

- Position (i.e., x and y-axis)
- Color (line)
- Fill (fill)
- Shape (of points)
- Line type
- Size
- Opacity

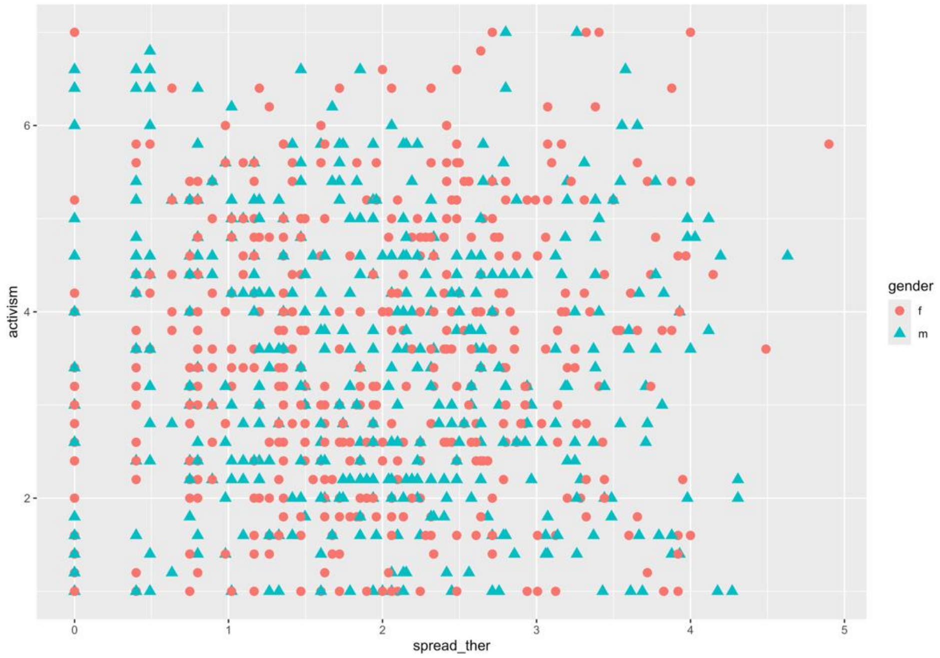
Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism))+
  geom_point(color = "red",
             shape = 18)
```



Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender))+
  geom_point(size = 3)
```

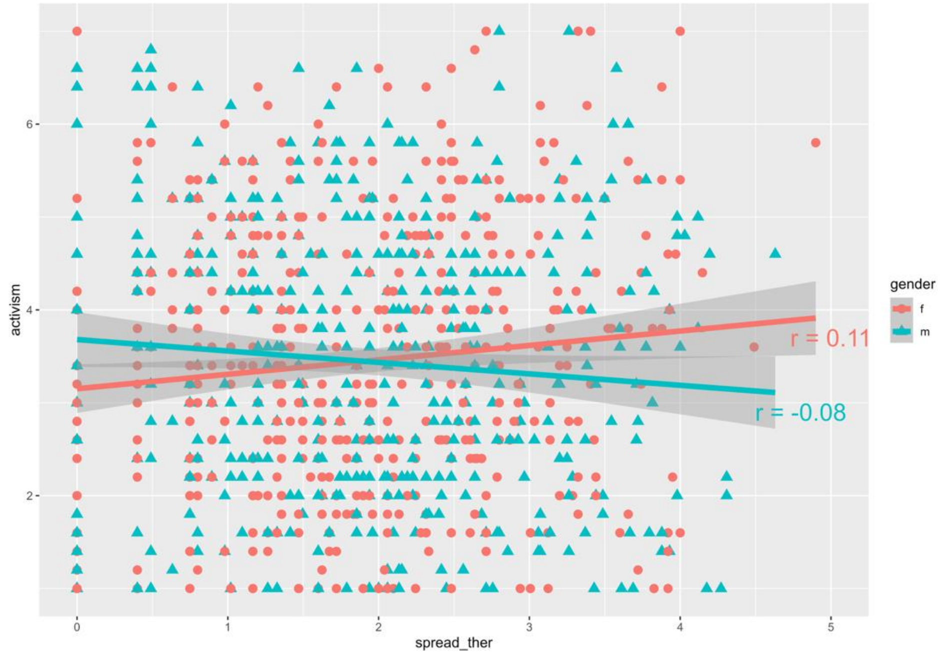


Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender))+
  geom_point(size = 3)+
  geom_smooth(linewidth = 2,
              method = "lm")
```

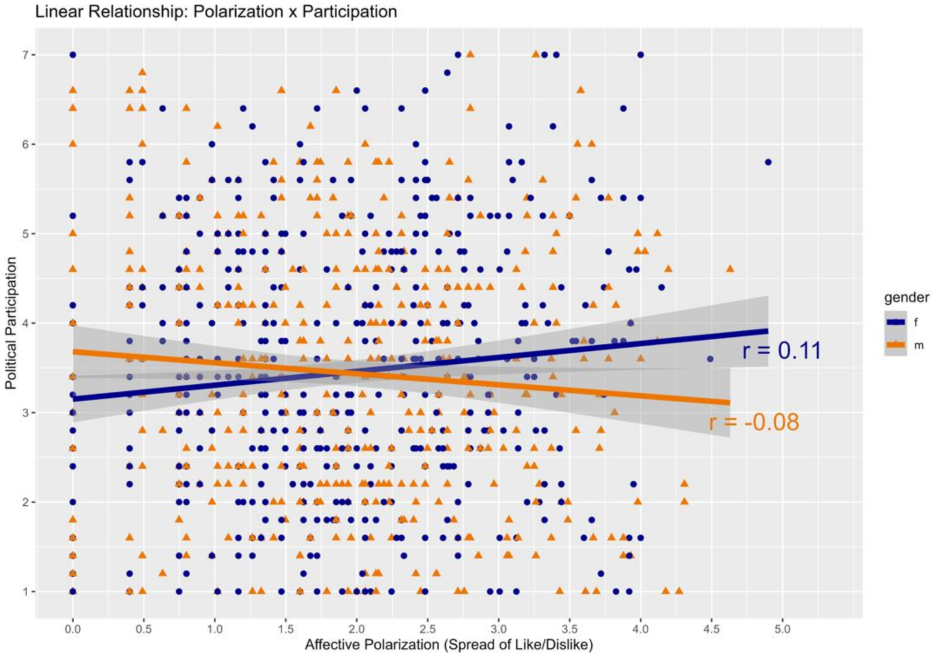
Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender)) +
  geom_point(size = 3) +
  geom_smooth(linewidth = 2, method = "lm")
  geom_text(data = cor_labels,
            aes(label = label),
            show.legend = FALSE,
            size = 6.5)
```



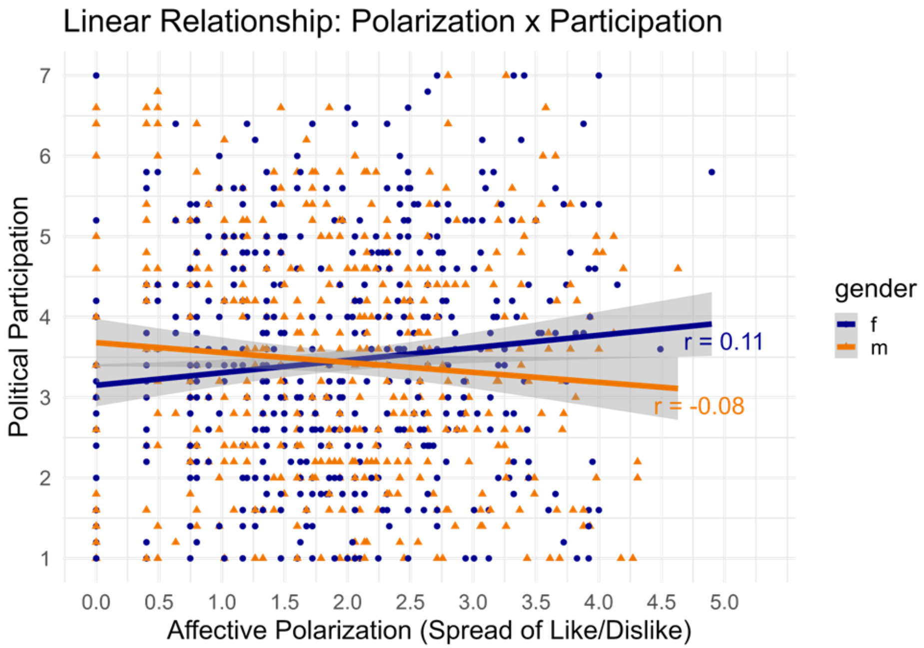
Aesthetics, Mapping, & Layers

```r
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender))+
  geom_point(size = 2)+
  geom_smooth(linewidth = 2,
              method = "lm")+
  geom_text(data = cor_labels,
            aes(label = label),
            show.legend = FALSE,
            size = 6.5)+
  ggtitle("Linear Relationship: Polarization x Participation")+
  scale_x_continuous(
    name = "Affective Polarization (Spread of Like/Dislike)",
    limits = c(0, 5.3),
    breaks = seq(0, 5, 0.5))+
  scale_y_continuous(
    name = "Political Participation",
    limits = c(1, 7),
    breaks = seq(1, 7, 1))+
  scale_color_manual(
    values = c(
      m = "#EE7600",
      f = "#00008B"))+
```



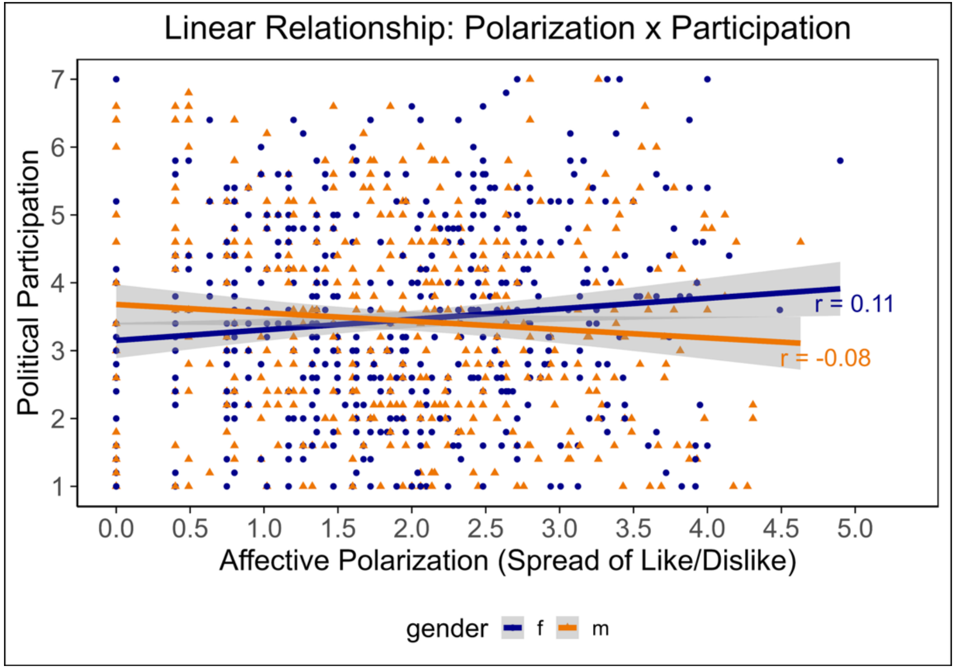
Aesthetics, Mapping, & Layers

```
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender))+
  geom_point(size = 2)+
  geom_smooth(linewidth = 2,
              method = "lm")+
  geom_text(data = cor_labels,
            aes(label = label),
            show.legend = FALSE,
            size = 6.5)+
  ggtitle("Linear Relationship: Polarization x Participation")+
  scale_x_continuous(
    name = "Affective Polarization (Spread of Like/Dislike)",
    limits = c(0, 5.3),
    breaks = seq(0, 5, 0.5))+
  scale_y_continuous(
    name = "Political Participation",
    limits = c(1, 7),
    breaks = seq(1, 7, 1))+
  scale_color_manual(
    values = c(
      m = "#EE7600",
      f = "#00008B"))+
  theme_minimal(base_size = 20)
```



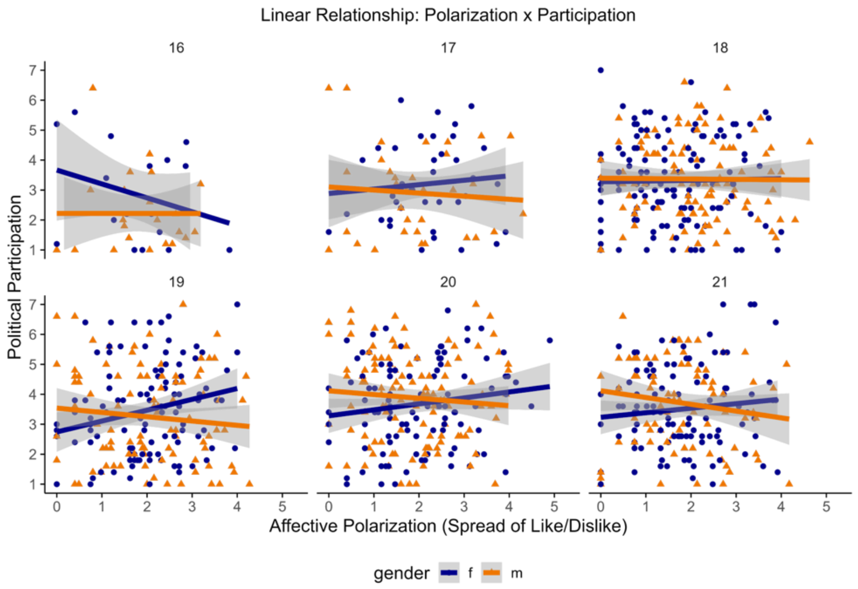
Aesthetics, Mapping, & Layers

```
theme(
  axis.title.x = element_text(size = 22),
  axis.title.y = element_text(size = 22),
  axis.text.x = element_text(size = 20),
  axis.text.y = element_text(size = 20),
  legend.text = element_text(25),
  legend.position = "bottom",
  plot.title = element_text(size = 25, hjust = 0.5),
  axis.line = element_line(linewidth = 0.5),
  axis.ticks = element_line(linewidth = 0.5),
  panel.background = element_rect("white"),
  plot.background = element_rect("white"),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()
)
```



Grouping

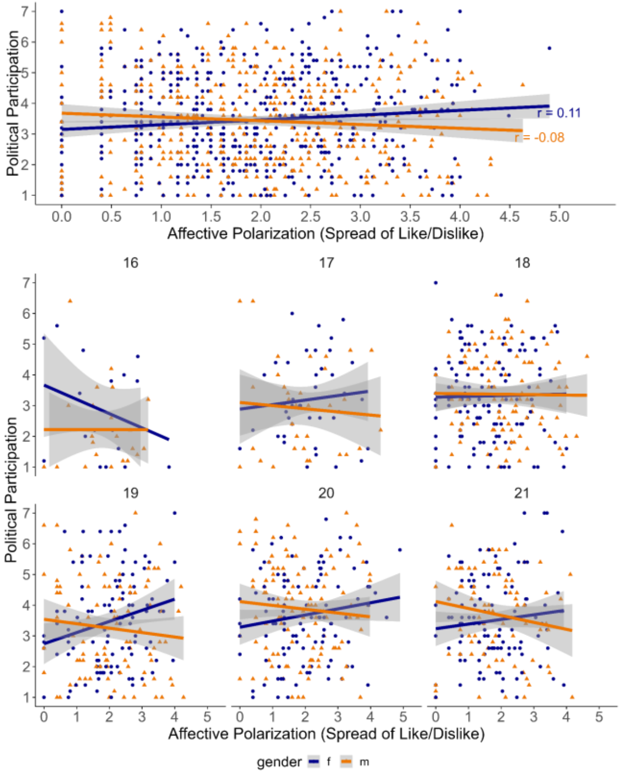
```r
ggplot(data = df_nl,
       mapping = aes(x = spread_ther,
                     y = activism,
                     color = gender,
                     shape = gender))+
  geom_point(size = 2)+
  geom_smooth(linewidth = 2,
              method = "lm")+
  ggtitle("Linear Relationship: Polarization x Participation")+
  scale_x_continuous(
    name = "Affective Polarization (Spread of Like/Dislike)",
    limits = c(0, 5.3),
    breaks = seq(0, 5, 0.5))+
  scale_y_continuous(
    name = "Political Participation",
    limits = c(1, 7),
    breaks = seq(1, 7, 1))+
  scale_color_manual(
    values = c(
      m = "#EE7600",
      f = "#00008B"))+
  theme_minimal(base_size = 15)+
  theme(legend.position = "bottom",
        plot.title = element_text(size = 15, hjust = 0.5),
        axis.line = element_line(linewidth = 0.5),
        axis.ticks = element_line(linewidth = 0.5),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank())+
  facet_wrap(~age, ncol = 3)
```



Grouping & Saving

```
library(ggpubr)
final_plot <- ggarrange(plot_ovr,
                        plot_by_age,
                        ncol = 1,
                        heights = c(1, 2),
                        common.legend = TRUE,
                        legend = "bottom")

ggsave(plot = final_plot,
       "final_plot.png",
       width = 12,
       height = 15,
       dpi = 750)
```



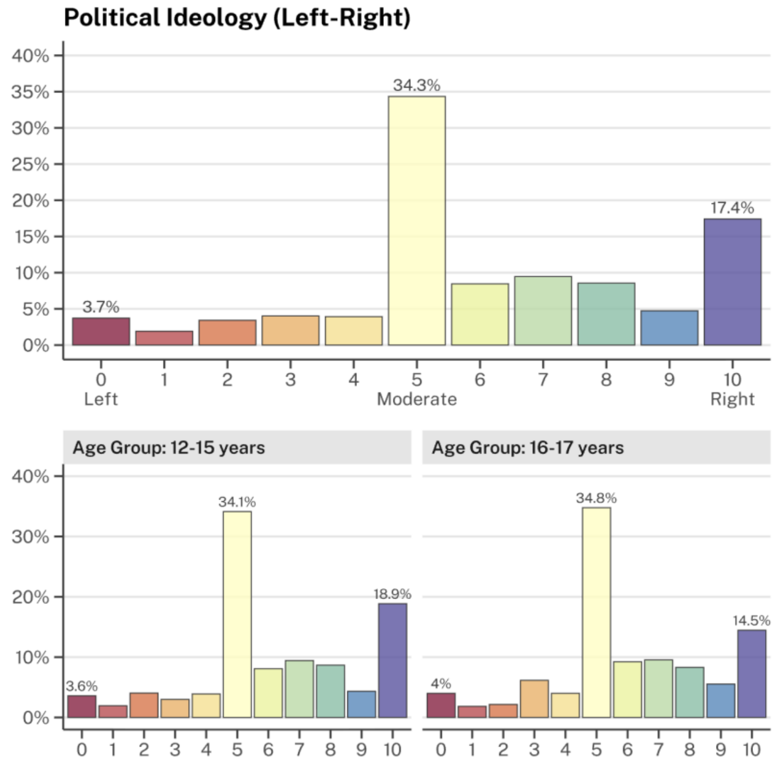
3. Common Plots in the Social and Cognitive Sciences

Data Descriptives

Descriptives for Categorical Variables

- Bar plot

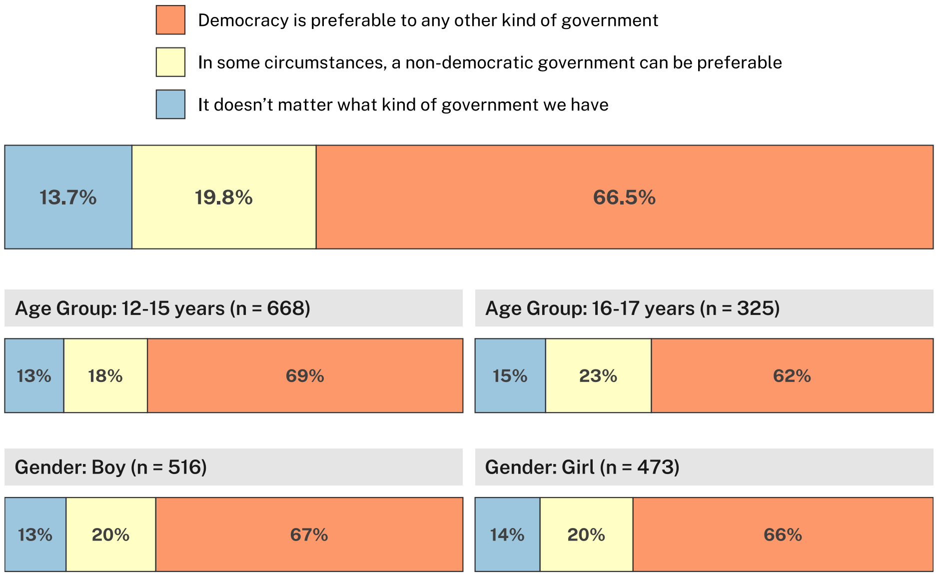
```
```{r}
# Total Sample
p_lr_all ← df ▷
  filter(!is.na(`Left-Right`)) ▷
  count(`Left-Right`) ▷
  mutate(
    percent = n / sum(n) * 100,
    `Left-Right` = factor(`Left-Right`, levels = 0:10),
    show_label = `Left-Right` %in% c("0", "5", "10")
  ) ▷
  ggplot(aes(x = `Left-Right`, y = percent / 100)) +
  geom_col(aes(fill = `Left-Right`), alpha = 0.8, color = "grey20", linewidth = 0.3) +
  geom_text(aes(label = if_else(show_label, paste0(round(percent, 1), "%"), "")),
            vjust = -0.5, size = 3.5, color = "grey25") +
  scale_fill_manual(values = ideo_col, drop = FALSE) +
  scale_x_discrete(labels = c("0\nLeft", "1", "2", "3", "4",
                              "5\nModerate", "6", "7", "8", "9", "10\nRight")) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1),
                     limits = c(0, 0.40),
                     breaks = seq(0, 0.40, by = 0.05)) +
  labs(title = "Political Ideology (Left-Right)", x = NULL, y = NULL) +
  theme_ideo()
```


```

Descriptives for Categorical Variables

- Stacked bar charts for Likert Scales

```r
p_demsys_all ← df ▷
  count(`Democracy as System`) ▷
  mutate(pct = n / sum(n)) ▷
  ggplot(aes(x = pct, y = "", fill = `Democracy as System`)) +
  geom_col(position = position_stack(), alpha = 0.9, color = "grey20", linewidth = 0.3) +
  geom_text(
    aes(label = if_else(pct ⩾ 0.05, paste0(round(pct * 100, 1), "%"), "")),
    position = position_stack(vjust = 0.5),
    size = 4,
    color = "grey25",
    fontface = "bold"
  ) +
  scale_fill_manual(values = demsys_col) +
  scale_x_continuous(labels = percent_format(), expand = c(0, 0)) +
  labs(
    x = NULL,
    y = NULL,
    fill = NULL
  ) +
  theme_likert() +
  guides(fill = guide_legend(reverse = FALSE, nrow = 3, label.theme = element_text(size = 10)))
```

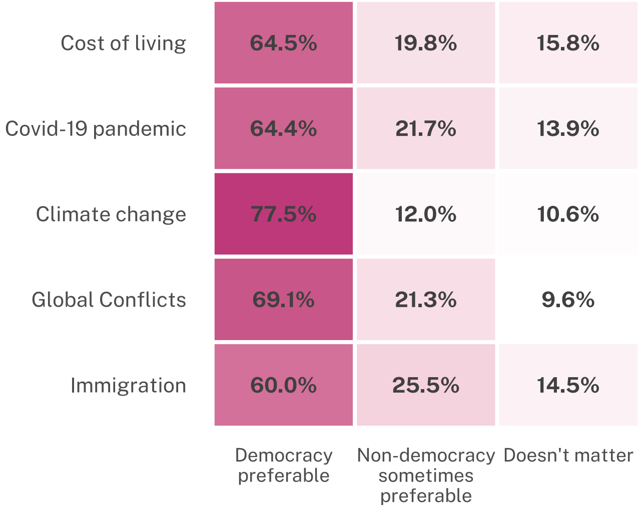


Relationship between Categorical Variables

Tile Plots

- E.g., Is Democratic Support Associated with which Crisis Young People found most impactful on their lives?

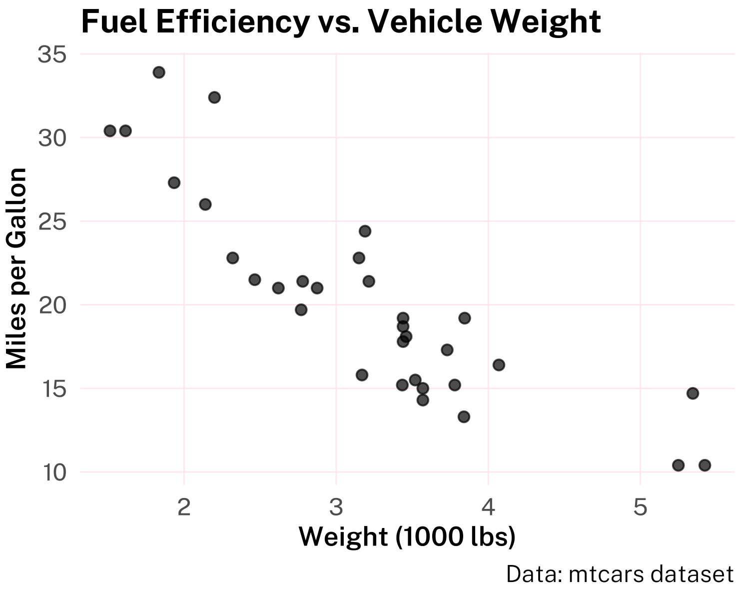
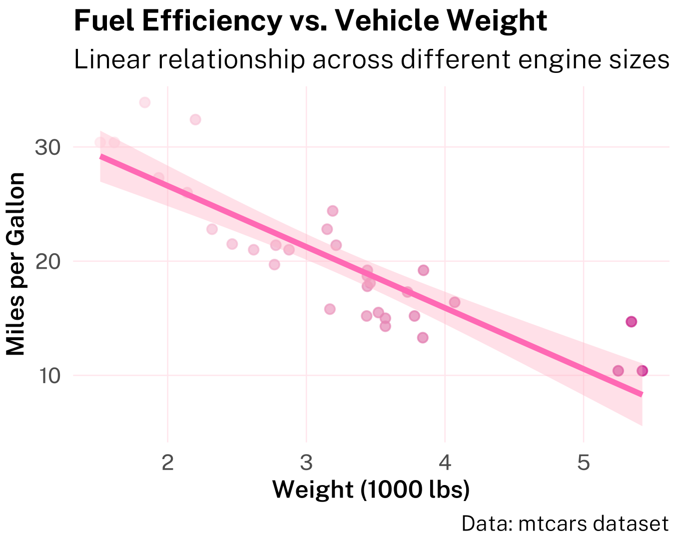
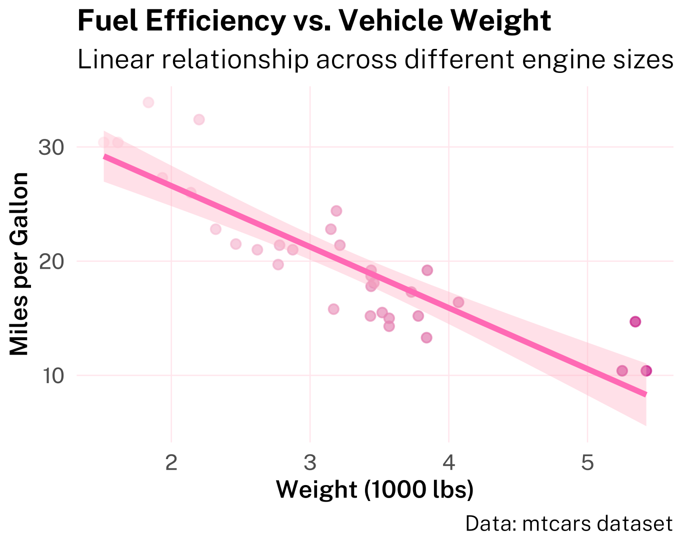
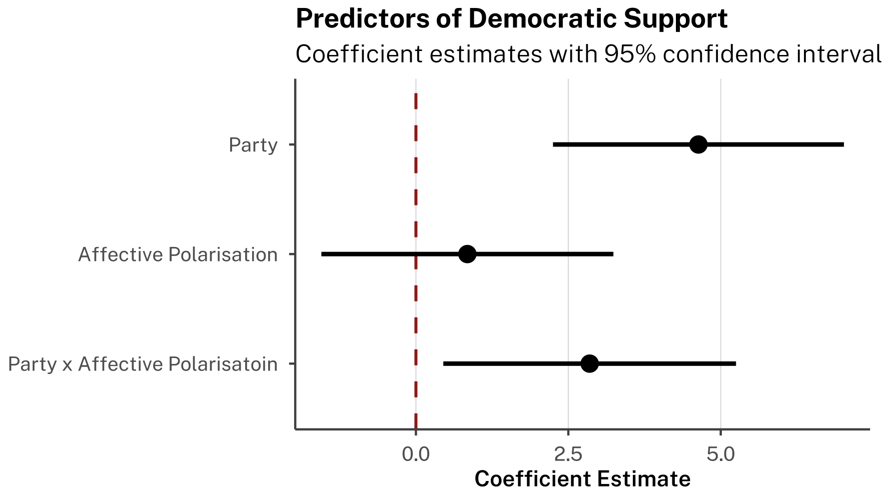
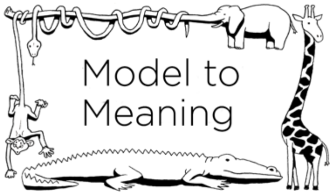
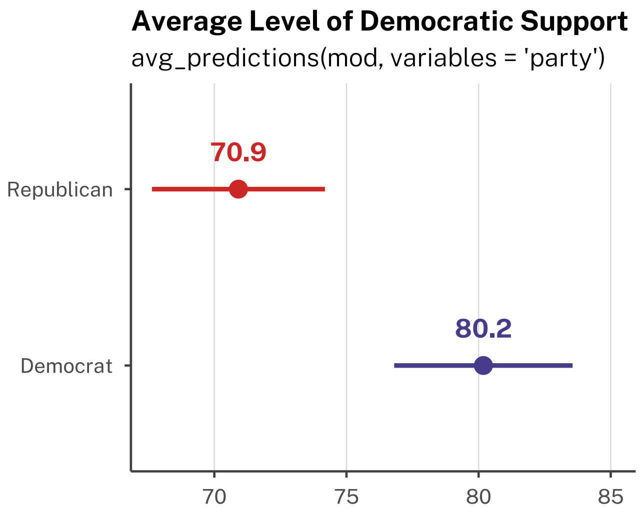
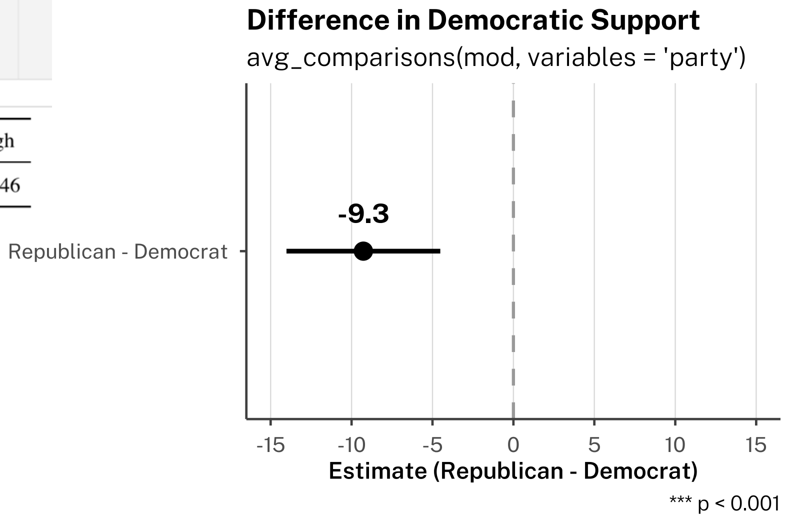
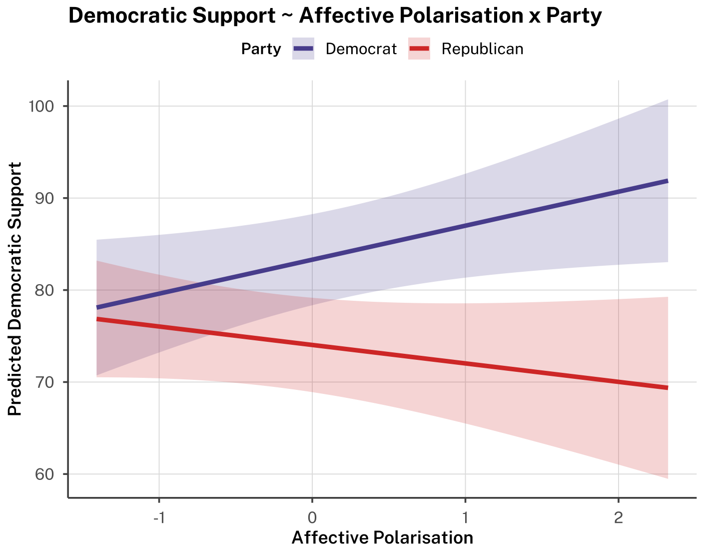
```r
p_formative_tile <- df_formative_tile |>
  ggplot(aes(x = `Democracy as System`, y = `Formative Issue`)) +
  geom_tile(aes(fill = pct), color = "white", linewidth = 1) +
  geom_text(aes(label = sprintf("%.1f%%", pct)),
            color = "grey25", size = 4, fontface = "bold") +
  scale_fill_gradient(low = "white", high = "#BE3979") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 15)) +
  labs(
    x = NULL,
    y = NULL,
    fill = "Percentage",
  ) +
  theme_nice() +
  theme(
    legend.position = "none",
    panel.grid = element_blank(),
    axis.text.x = element_text(size = 10, hjust = 0.5),
    axis.text.y = element_text(size = 11),
    plot.title = element_text(face = "bold", size = 14, hjust = 0),
    plot.subtitle = element_text(size = 11, hjust = 0, color = "grey40")
  )
```



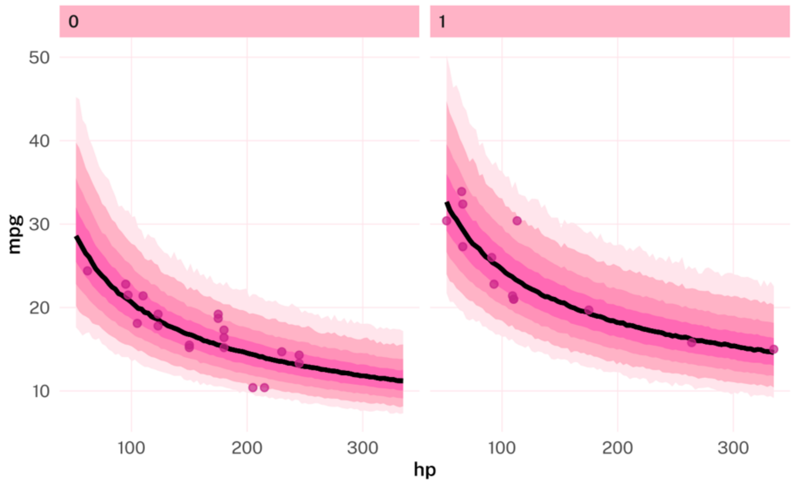
Relationship between Continuous Variables

Scatter plot

````
```{r fig.height=4, fig.width=5, fig.dpi=600, fig.showtext=TRUE}
my_first_plot <- mtcars |>
  mutate(cyl = factor(cyl, labels = c("4 Cylinders", "6 Cylinders", "8 Cylinders"))) |>
  ggplot(aes(x = wt, y = mpg)) +
  geom_point(size = 2, alpha = 0.7) +
  labs(
    title = "Fuel Efficiency vs. Vehicle Weight",
    x = "Weight (1000 lbs)",
    y = "Miles per Gallon",
    color = "Weight",
    caption = "Data: mtcars dataset"
  ) +
  theme_nice() +
  theme(legend.position = "none")

my_first_plot
````


````

Relationship between Continuous Variables

Scatter plot + **linear regression line**

```
```{r fig.height=4, fig.width=5, fig.dpi=600, fig.showtext=TRUE}
my_first_plot <- mtcars |>
  mutate(cyl = factor(cyl, labels = c("4 Cylinders", "6 Cylinders", "8 Cylinders"))) |>
  ggplot(aes(x = wt, y = mpg)) +
  geom_point(aes(color = wt), size = 2, alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE, color = "#FF69B4", fill = "#FFB3C6", linewidth = 1.4) +
  scale_color_gradient(low = "#FFE5EC", high = "#C71585") +
  labs(
    title = "Fuel Efficiency vs. Vehicle Weight",
    subtitle = "Linear relationship across different engine sizes",
    x = "Weight (1000 lbs)",
    y = "Miles per Gallon",
    color = "Weight",
    caption = "Data: mtcars dataset"
  ) +
  theme_nice() +
  theme(legend.position = "none")

my_first_plot
```
```



Relationship between Continuous Variables

geom_smooth / stat_smooth actually fit models → **quick spotting of patterns in the data** (y ~ x)

```
```{r fig.height=4, fig.width=5, fig.dpi=600, fig.showtext=TRUE}
my_first_plot <- mtcars |>
  mutate(cyl = factor(cyl, labels = c("4 Cylinders", "6 Cylinders", "8 Cylinders"))) |>
  ggplot(aes(x = wt, y = mpg)) +
  geom_point(aes(color = wt), size = 2, alpha = 0.7) +
  geom_smooth(method = "lm", se = TRUE, color = "#FF69B4", fill = "#FFB3C6", linewidth = 1.4) +
  scale_color_gradient(low = "#FFE5EC", high = "#C71585") +
  labs(
    title = "Fuel Efficiency vs. Vehicle Weight",
    subtitle = "Linear relationship across different engine sizes",
    x = "Weight (1000 lbs)",
    y = "Miles per Gallon",
    color = "Weight",
    caption = "Data: mtcars dataset"
  ) +
  theme_nice() +
  theme(legend.position = "none")

my_first_plot
```
```



Model Interpretation

Model interpretation

- We often don't just describe data but also **fit statistical models** (e.g., t-test, logistic regression, linear mixed models,...)
- We often have more than just an outcome and predictor
  - E.g., control variables, interaction, experimental conditions ...
- Rather than using geom_smooth() → **We want to plot the effects from our models!**

Example: Democratic Support

1. Do Democrats and Republicans differ in their level of democratic support (controlling for age, gender, education)?
2. Is the association between affective polarisation and democratic support different for Democrats and Republicans (controlling for age, gender, education)?

```{r}
mod <- lm(democratic_support ~ party * affective_polarisation + age + gender + education, data = survey_data)
summary(mod)
```

```
Call:
lm(formula = democratic_support ~ party * affective_polarisation +
    age + gender + education, data = survey_data)

Residuals:
    Min       1Q   Median       3Q      Max
-59.655 -11.565    3.185   15.211   34.831

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                       58.09580    5.92879   9.799  < 2e-16 ***
party1                             4.63471    1.21219   3.823 0.000163 ***
affective_polarisation             0.84399    1.21666   0.694 0.488460
age                                0.22099    0.07675   2.879 0.004302 **
genderWoman                       -0.04972    2.42582  -0.020 0.983662
genderOther                      -21.18842   20.28665  -1.044 0.297201
educationCollege, but not degree   7.01527    4.37776   1.602 0.110206
education2-year college degree     8.16225    4.84624   1.684 0.093278 .
education4-year college degree     9.12924    3.98320   2.292 0.022671 *
educationPostgraduate degree       2.91476    4.12847   0.706 0.480781
party1:affective_polarisation      2.85017    1.21957   2.337 0.020161 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.64 on 273 degrees of freedom
Multiple R-squared:  0.1264,    Adjusted R-squared:  0.0944
F-statistic:  3.95 on 10 and 273 DF,  p-value: 4.778e-05
```

Coefficient Forest Plots

```r
forest_data |>
  ggplot(aes(x = estimate, y = term, xmin = conf.low, xmax = conf.high)) +
  # Add vertical line at zero (null effect)
  geom_vline(xintercept = 0, linetype = "dashed", color = "firebrick4", linewidth = 0.8) +
  # Add point estimates with confidence intervals
  geom_pointrange(color = "black", linewidth = 1.2, size = 0.8) +
  # Labels
  labs(
    title = "Predictors of Democratic Support",
    subtitle = "Coefficient estimates with 95% confidence intervals",
    x = "Coefficient Estimate",
    y = NULL
  ) +
  theme(panel.grid.major.y = element_blank())
```



A Very Short Intro to `marginaleffects`

- Marginal effects → model-agnostic
  - Does not care about contrast coding (e.g., contr.sum)
- Models as prediction machines (Rohrer & Arel-Bundock, 2025)
  - **Predictions** - Model-based estimates (e.g., predicted means across countries)
  - **Comparisons** - Differences between conditions (e.g., treatment vs. control)
  - **Slopes** - Rate of change of a predictor (e.g., how effect of X varies by moderator)



Average predictions (marginal means)

Do Democrats and Republicans differ in their level of democrat support (averaged across all other variables in our model)?

```{r}
mm_data <- avg_predictions(mod, variables = "party") |>
  as_tibble() |>
  select(party, estimate, p.value, conf.low, conf.high)

mm_data |> tt()
```

party	estimate	p.value	conf.low	conf.high
Democrat	80.18098	0	76.80524	83.55672
Republican	70.91155	0	67.63894	74.18416



Average comparisons

- Is the difference in democratic support between Democrats and Republicans significant (averaged over all other variables in the model)?

```{r}
com_data <- avg_comparisons(mod, variables = "party") |>
  as_tibble() |>
  select(contrast, estimate, p.value, conf.low, conf.high)

com_data |> tt()
```

contrast	estimate	p.value	conf.low	conf.high
Republican - Democrat	-9.269427	0.0001316052	-14.02111	-4.517746



Average Slopes by Party

What is the association between affective polarisation and democratic support in Democrats and Republicans (on average)? → Helps understand significant interactions!

```{r}
avg_slopes(mod, variables = "affective_polarisation", by = "party") |>
  as_tibble() |>
  select(party, estimate, p.value, conf.low, conf.high) |>
  tt()
```

party	estimate	p.value	conf.low	conf.high
Democrat	3.694161	0.03574634	0.2460141	7.142308
Republican	-2.006172	0.23398036	-5.3099284	1.297584

Interpreting interactions

Is the association between affective polarisation and democratic support different for Democrats and Republicans (on average across other variables)?

```
```{r fig.height=5.5, fig.width=7, fig.dpi=600, fig.showtext=TRUE}
plot_predictions(mod,
                 condition = c("affective_polarisation", "party"),
                 draw = FALSE) |>  # Get the data instead of the plot
  ggplot(aes(x = affective_polarisation, y = estimate,
             ymin = conf.low, ymax = conf.high,
             color = party, fill = party)) +
  # Confidence ribbon
  geom_ribbon(alpha = 0.2, color = NA) +
  # Prediction line with custom thickness
  geom_line(linewidth = 1.5) +
  # Colors
  scale_color_manual(values = c("Democrat" = "slateblue4", "Republican" = "firebrick3")) +
  scale_fill_manual(values = c("Democrat" = "slateblue4", "Republican" = "firebrick3")) +
  # Labels
  labs(
    title = "Democratic Support ~ Affective Polarisation x Party",
    x = "Affective Polarisation",
    y = "Predicted Democratic Support",
    color = "Party",
    fill = "Party"
  ) +
  theme_nice()
```


```

More fancy Bayesian stuff …

````
```{r}
m_mpg_am = brm(
  mpg ~ log(hp) * am,
  data = mtcars,
  family = lognormal
)
```
````

```
mtcars %>%
  data_grid(hp = seq_range(hp, n = 101), am) %>%
  add_predicted_draws(m_mpg_am) %>%
  ggplot(aes(x = hp, y = mpg)) +
  # Pink ribbons with custom line color
  stat_lineribbon(aes(y = .prediction), .width = c(.99, .95, .8, .5),
                  color = "black", linewidth = 1.4) +
  geom_point(data = mtcars, color = "#C71585", size = 2, alpha = 0.7) +
  # Pink color palette for ribbons (lightest to darkest)
  scale_fill_manual(values = c("#FFE5EC", "#FFB3C6", "#FF92B8", "#FF69B4")) +
  labs(y = "mpg") +
  facet_wrap(~ am) +
  theme_nice() +
  theme(
    # Light pink grid lines
    panel.grid.major = element_line(color = "#FFE5EC", linewidth = 0.3),
    # Make axis lines and ticks invisible
    axis.line = element_blank(),
    axis.ticks = element_blank(),
    legend.position = "none"
  )
```



European Commission

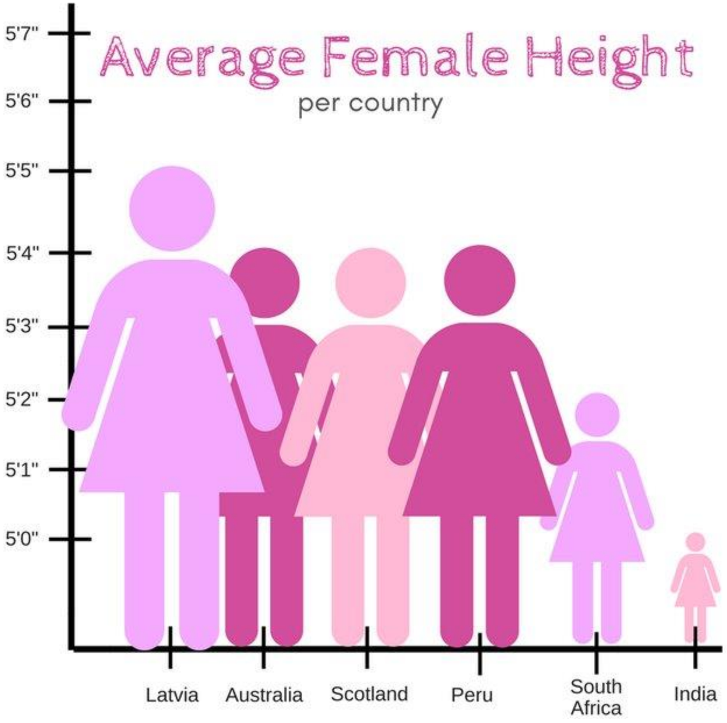
UK Research and Innovation

Will sometimes
depend on context
& purpose!!!

Bad Practices

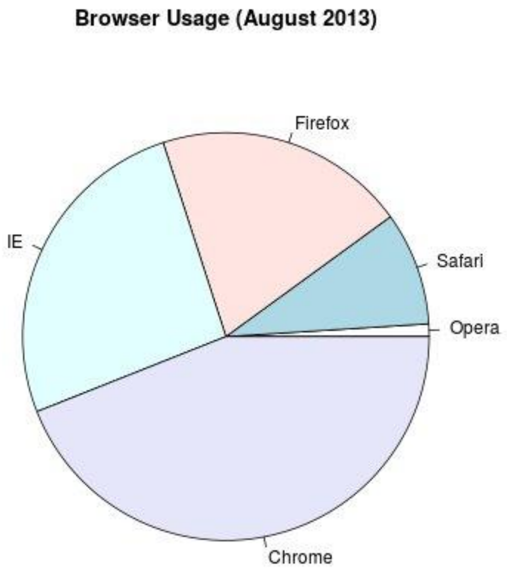
Bad Practices

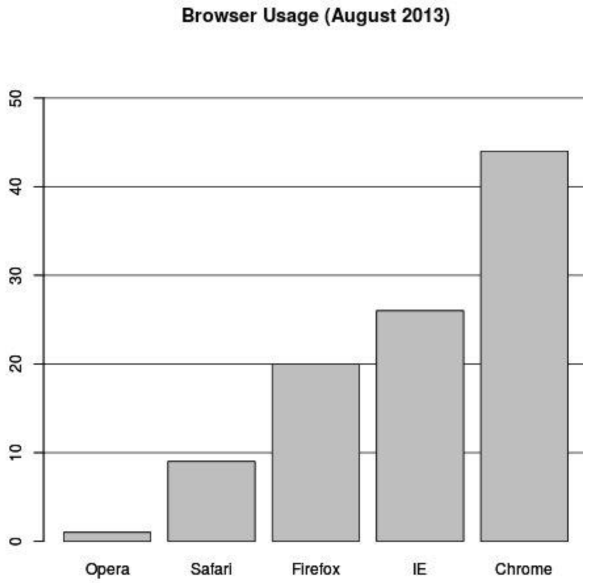
- Misleading y-axis



Bad Practices

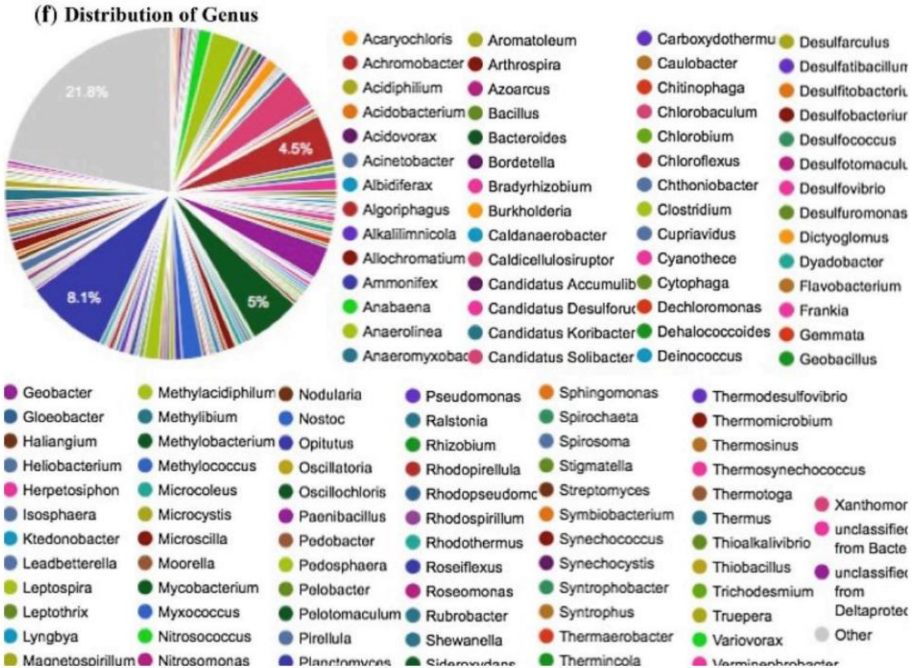
- Pie charts (in general)





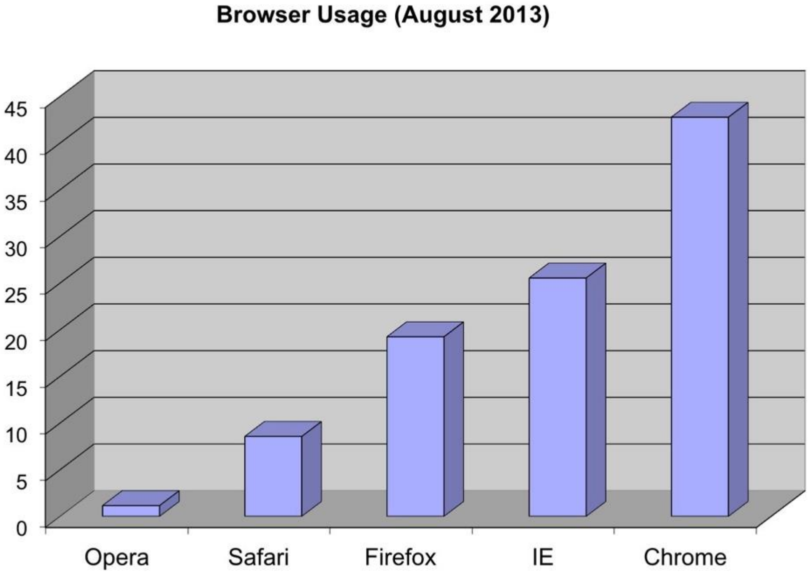
Bad Practices

- Pie charts (in general) and visual overload
- Legend



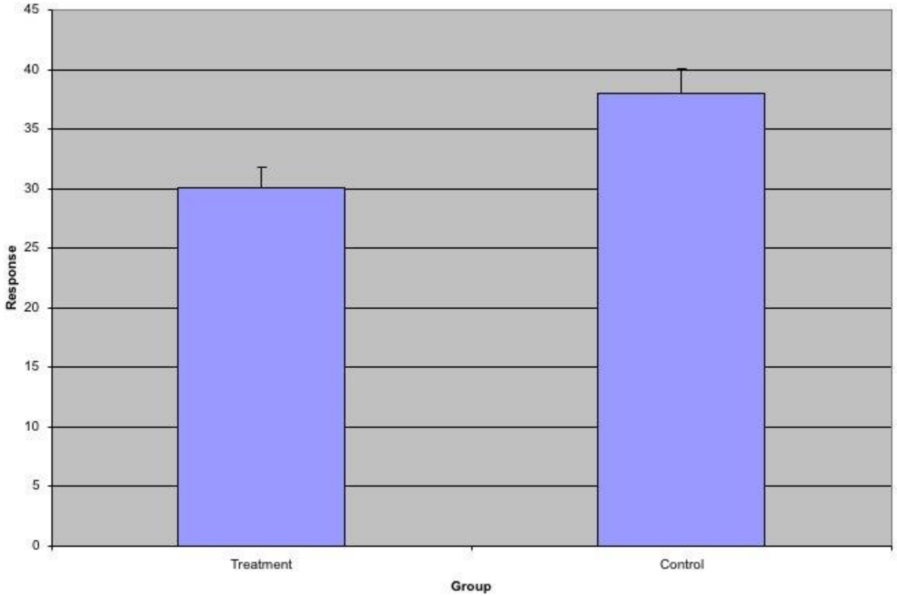
Bad Practices

- Low data/ink-ratio or unnecessary 3D elements



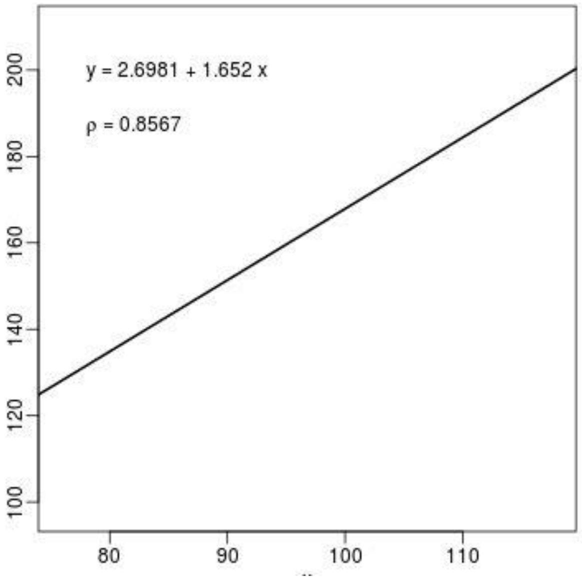
Bad Practices

- Bar plots to summarize data:



Bad Practices

- Regression line without confidence bands or data points:



Bad Practices

- Use of AI to generate the figure

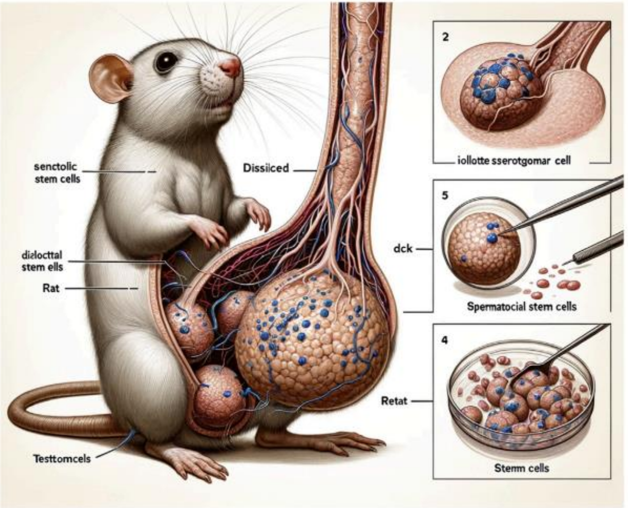
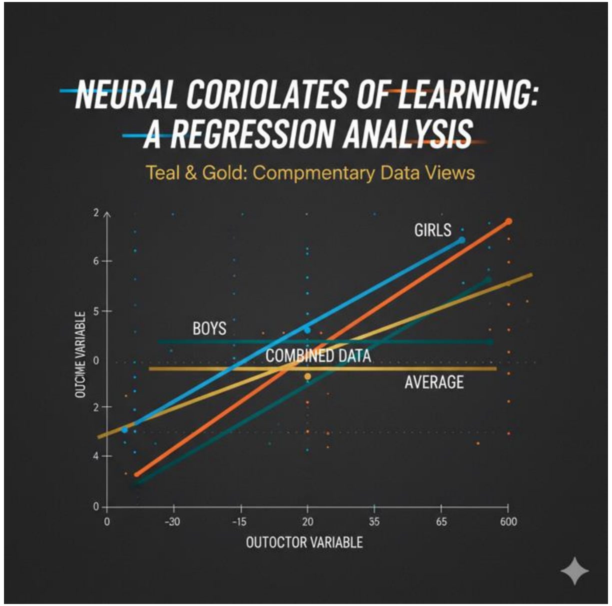


FIGURE 1
Spermatogonial stem cells, isolated, purified and cultured from rat testes.



Source: https://www.frontiersin.org/journals/cell-and-developmental-biology/articles/10.3389/fcell.2023.1339390/full

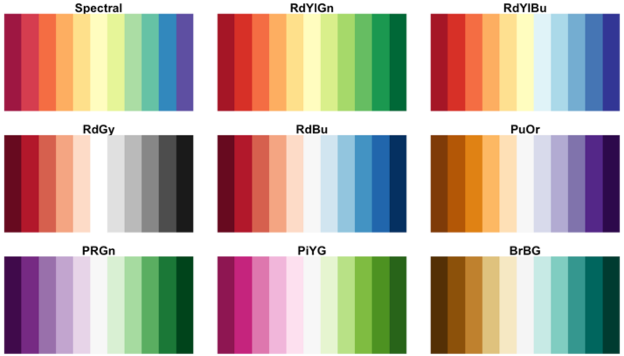
Bad Practices

- Lack of accessibility:
  - Small font
  - No colorblind-friendly palette
  - Missing labels or figure notes
  - …

Supplementary Resources

Aesthetics

Color Palettes



- RColorBrewer: https://colorbrewer2.org/
    - ggplot: scale_fill_brewer(palette = "Spectral")
- Viridis Color Palettes: https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html
    - ggplot: scale_color_viridis_d(option = "inferno")
- Beyonce Palettes: https://github.com/dill/beyonce
- Paul Tol's Notes: https://cran.r-project.org/web/packages/khroma/vignettes/tol.html

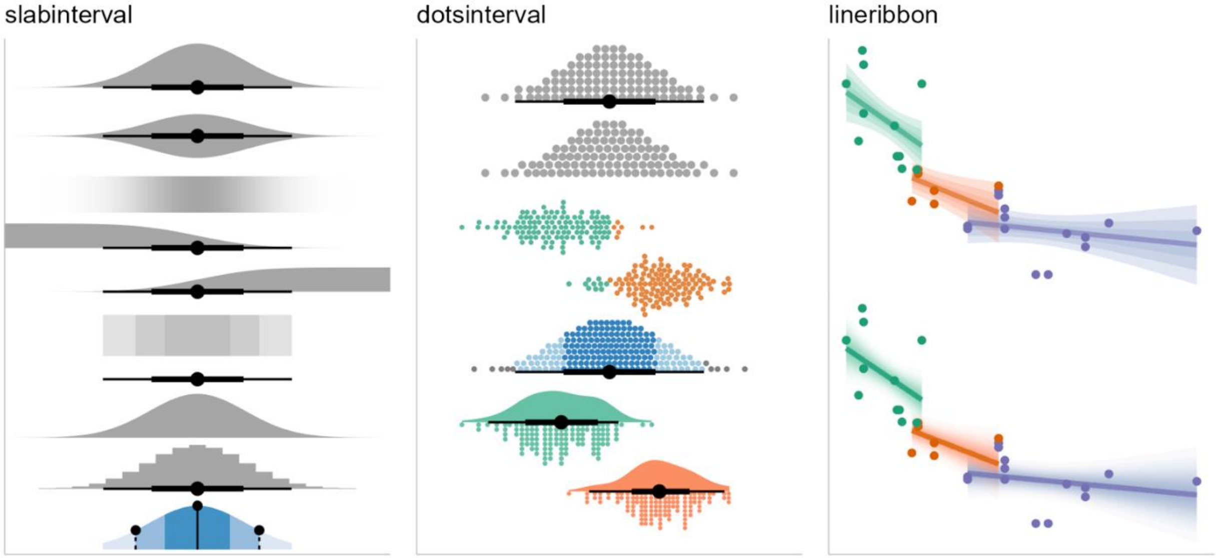
Fonts (Sans Serif Fonts)

- Public Sans
- **Roboto Condensed**
- Arial Narrow
- Open Sans
- Source Sans Pro

Useful Extensions to ggplot2 in R

ggdist

- Visualizations of distributions and uncertainty



Some examples from the three main families of ggdist geometries

Visualising Textual Responses

library(wordcloud2) in R

```{r}
wc_right <- df |>
  select(`Right Meaning`) |>
  unnest_tokens(word, `Right Meaning`) |>
  anti_join(stop_words) |>
  count(word, sort = TRUE) |>
  filter(n > 2) |>
  wordcloud2(color = brewer.pal(8, "Dark2"),
             shuffle = FALSE, rotateRatio = 0.35)

wc_right
```

What do you associate with the term 'Right'?

Visualising Correlations (corrplot)

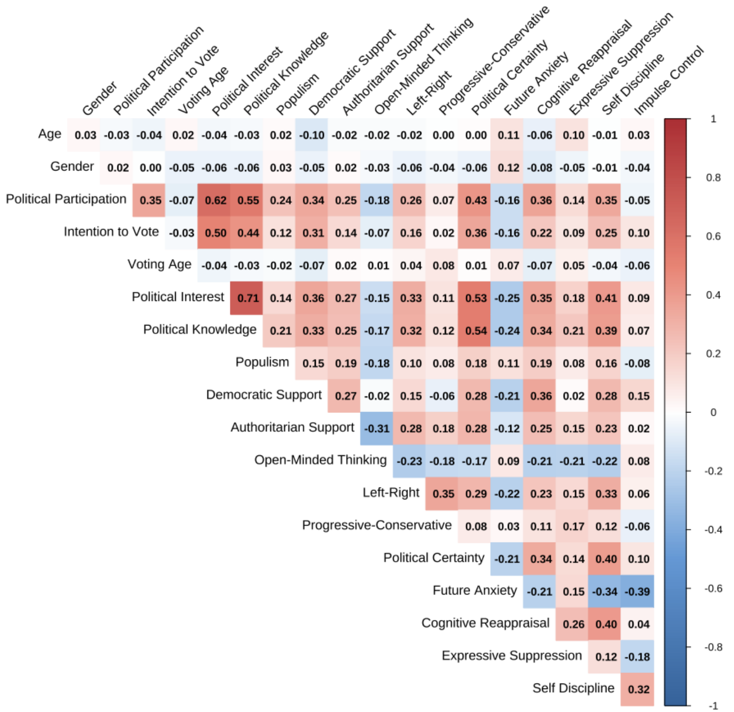
```
library("corrplot")
```

```r
```{r fig.width=12, fig.height=10}
# select data
corr_data <- df |>
  select(Age, Gender,
         `Political Participation`, `Intention to Vote`, `Voting Age`,
         `Political Interest`, `Political Knowledge`,
         Populism, `Democratic Support`, `Authoritarian Support`, `Open-Minded Thinking`,
         `Left-Right`, `Progressive-Conservative`, `Political Certainty`,
         `Future Anxiety`, `Cognitive Reappraisal`, `Expressive Suppression`,
         `Self Discipline`, `Impulse Control`) |>
  mutate(across(c(Gender, `Political Interest`), as.numeric))

# correlation analysis
corrs <- cor(corr_data)
corrs_p <- cor.mtest(corr_data, conf.level = 0.95)

# plot it
col <- colorRampPalette(c("#4477AA", "#77AADD", "#FFFFFF", "#EE9988", "#BB4444"))

corrplot(corrs, method="color", col=col(200),
         type="upper", order="original",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", tl.srt=45, # Text label color and rotation,
         # p.mat = corrs_p$p, sig.level = 0.05, insig = "blank",
         diag=FALSE, number.cex=0.85
         )
```
```



European Commission

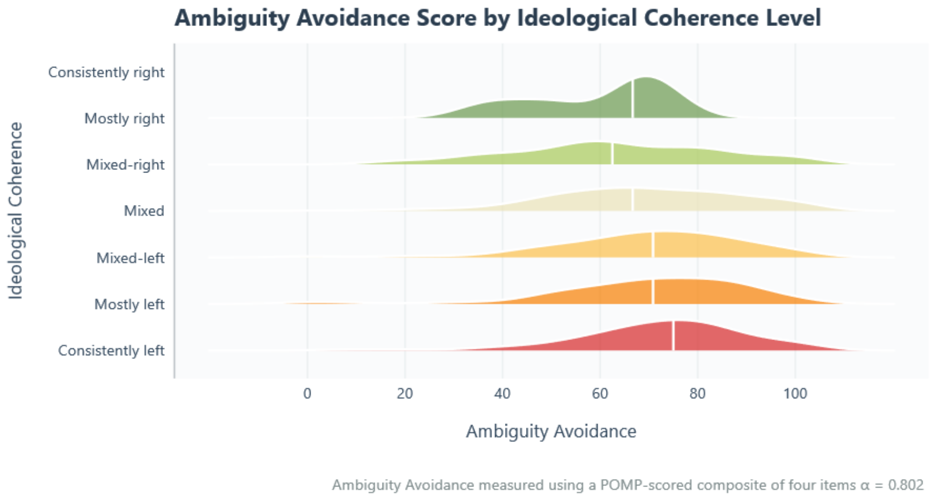
UK Research and Innovation

Ridge Plots (ggridges)

ggridges 0.5.7 Reference Articles Changelog

ggridges: Ridgeline plots in ggplot2

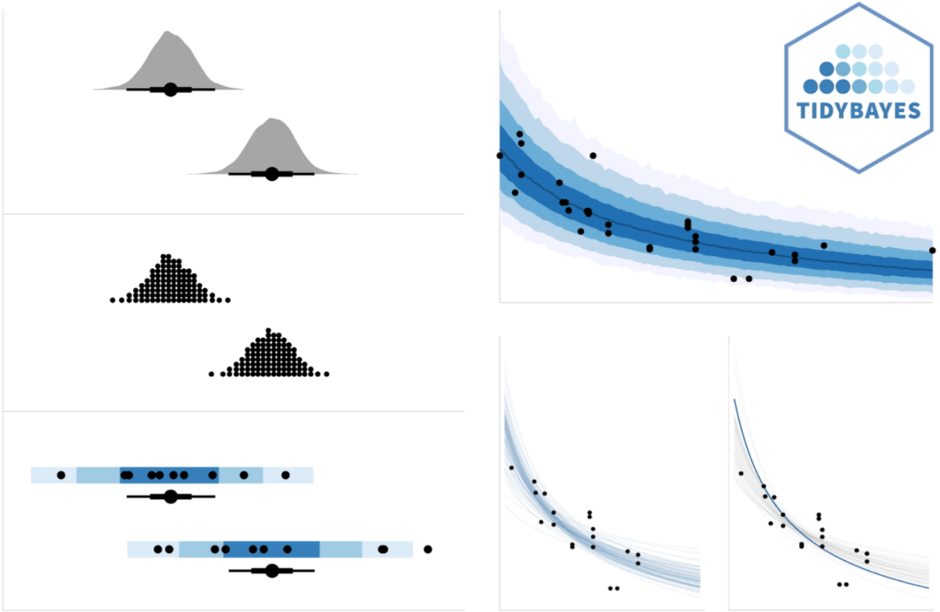
Ridgeline plots are partially overlapping line plots that create the impression of a mountain range. They can be quite useful for visualizing changes in distributions over time or space.



tidybayes

tidybayes: Bayesian analysis + tidy data + geoms

R-CMD-check passing | codecov 91% | CRAN 3.0.7 | downloads 16K/month | DOI 10.5281/zenodo.13770114



Preview of tidybayes plots

European Commission
UK Research and Innovation

ggstats

- E.g., includes geom_likert()

`ggstats`: extension to `ggplot2` for plotting stats

The `ggstats` package provides new statistics, new geometries and new positions for `ggplot2` and a suite of functions to facilitate the creation of statistical plots.

Installation & Documentation

To install **stable version**:

```
install.packages("ggstats")
```

```
ggplot(diamonds) +
  aes(y = clarity, fill = cut) +
  geom_bar(position = "likert") +
  geom_text(
    aes(by = clarity, label = custom_label(after_stat(prop))),
    stat = "prop",
    position = position_likert(vjust = .5)
  ) +
  scale_x_continuous(label = label_percent_abs()) +
  scale_fill_likert() +
  xlab("proportion")
```