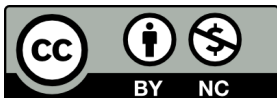


Business 4720 - Class 7

Data Visualization with R

Joerg Evermann

Faculty of Business Administration
Memorial University of Newfoundland
jevermann@mun.ca



Unless otherwise indicated, the copyright in this material is owned by Joerg Evermann. This material is licensed to you under the [Creative Commons by-attribution non-commercial license \(CC BY-NC 4.0\)](#)

XKCD comics are copyright by their creator (www.xkcd.com) and licensed under CC-BY-NC

This Class

What You Will Learn:

- ▶ Introduction to Visualization
- ▶ Visualizing data with R using the ggplot2 library

Why Visualize?

"A Picture is Worth 1000 Words"

- ▶ Humans are good at visual pattern recognition, but
 - ▶ Humans also identify patterns where there are none!
 - ▶ It's easy to mislead or deceive with visualization (others and oneself!)

Why Visualize?

Visual Discovery: Sense Making

- ▶ Exploration, confirmation, or verification
- ▶ Iterative, dynamic

Declarative Visualization: Storytelling

- ▶ Explanation
- ▶ Affirming, convincing
- ▶ Presenting, explaining
- ▶ Decision support
- ▶ Static

Operational Visualization: Monitoring

- ▶ Supervision, alarms
- ▶ Operational decision making

Purpose of Visualization

- ▶ Simplify, summarize & abstract
- ▶ Compare
- ▶ Identify trends, patterns & relationships
- ▶ Gain insights

Quantitative Messages

- 1 Time-series (e.g. line chart)
- 2 Ranking (e.g. bar chart)
- 3 Part-whole (e.g. pie chart)
- 4 Deviation (e.g. bar chart)
- 5 Frequency distribution (e.g. histogram, boxplot)
- 6 Correlation (e.g. scatter plot)
- 7 Nominal comparison (e.g. bar chart)
- 8 Geographic distribution (e.g. cartogram)

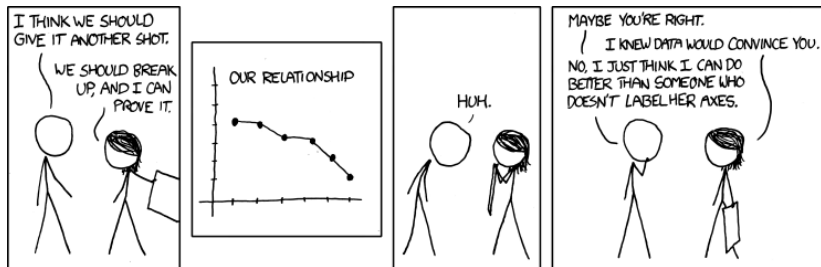
General Guidelines

- ▶ Do not deceive your target audience
- ▶ Do not diminish or hide relationships or trends
- ▶ Do not exaggerate relationships or trends
- ▶ Do not confuse or obfuscate

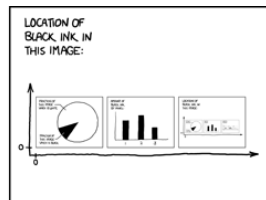
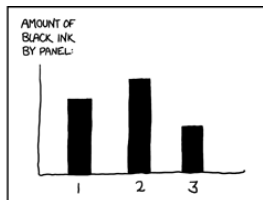
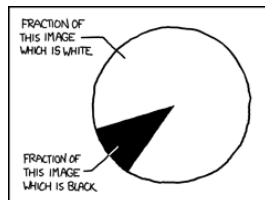
Specific "no-nos"

- ▶ Graph unrelated data to suggest non-existent relationships
- ▶ Scale multiple vertical axes to suggest correlations
- ▶ Truncate or scale axes to hide or exaggerate trend
- ▶ Scale in multiple dimensions
- ▶ Plot cumulative growth to hide trend
- ▶ Use maps for non-geographic data
- ▶ Use incomplete data ("cherry-picking")
- ▶ Use invalid data

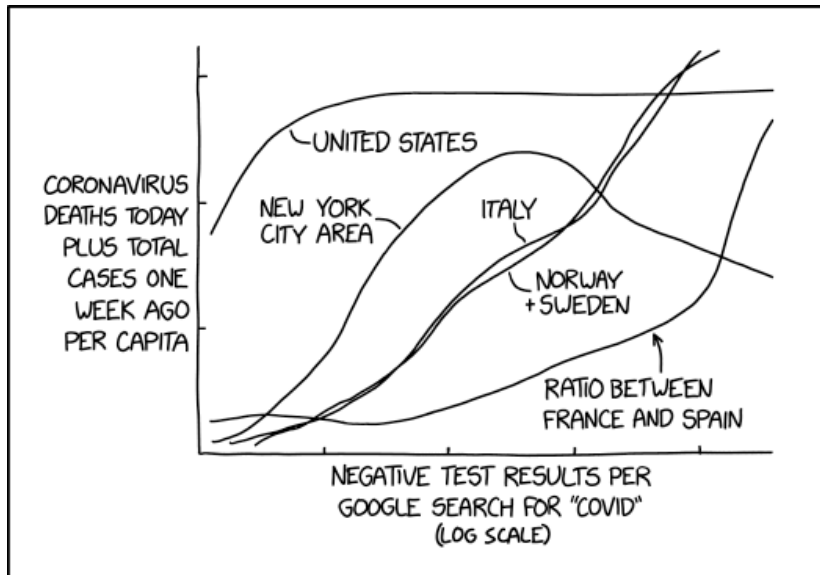
Label your Axes (XKCD)



Use Meaningful Data (XKCD)

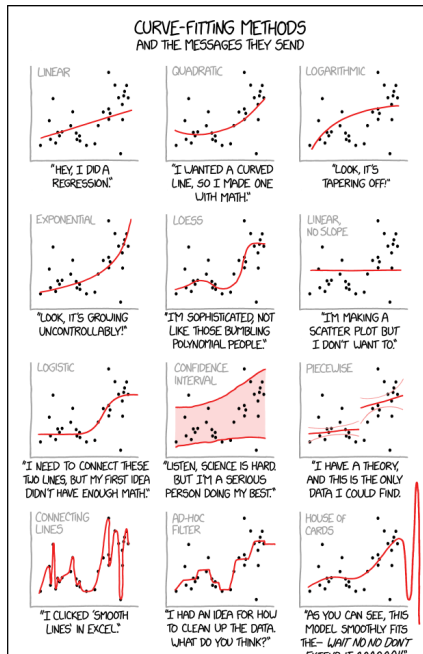


Use Related Data (XKCD)

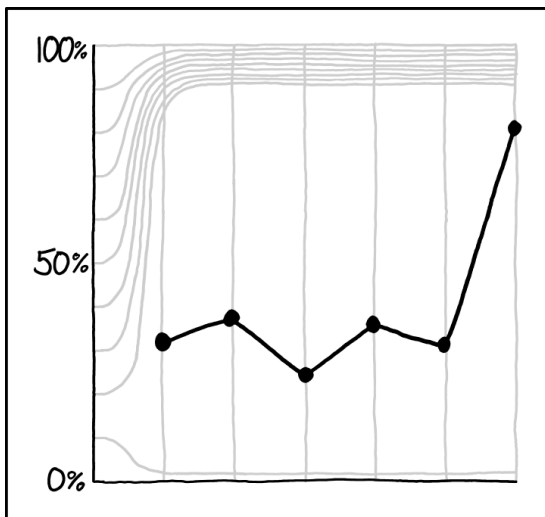


I'M A HUGE FAN OF WEIRD GRAPHS, BUT EVEN I ADMIT SOME OF THESE CORONAVIRUS CHARTS ARE LESS THAN HELPFUL.

Do Not Mislead (XKCD)



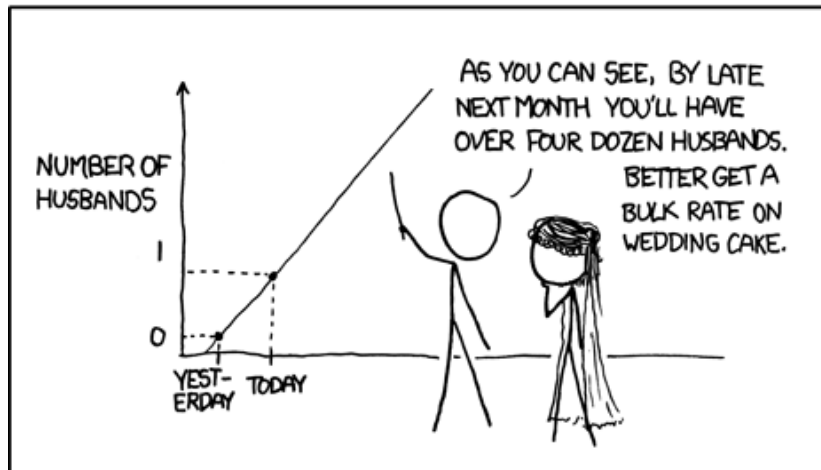
Choose Your Axes Meaningfully



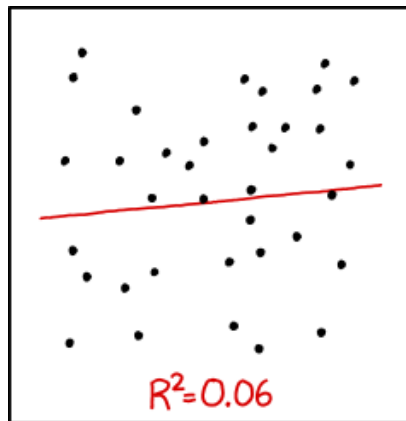
PEOPLE HAVE WISED UP TO THE "CAREFULLY CHOSEN Y-AXIS RANGE" TRICK, SO WE MISLEADING GRAPH MAKERS HAVE HAD TO GET CREATIVE.

Be Careful When Extrapolating (XKCD)

MY HOBBY: EXTRAPOLATING

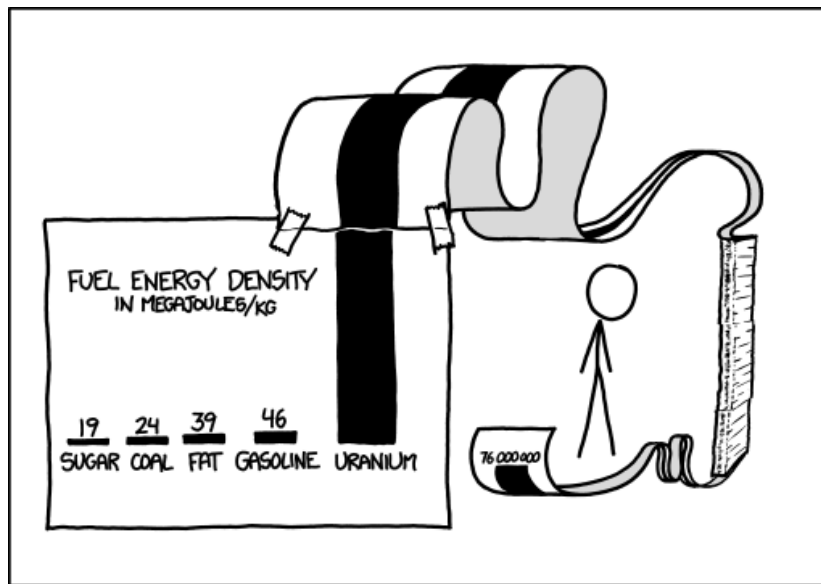


Verify Trends (XKCD)



I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.

Use Appropriate Scales (XKCD)

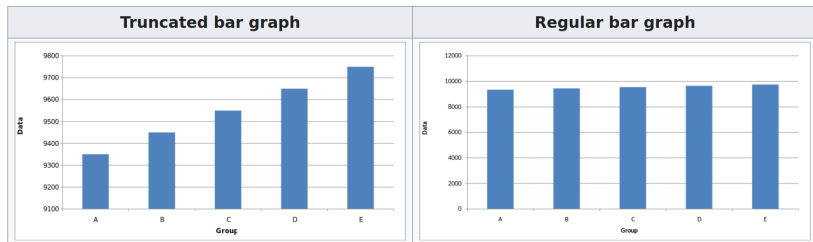


SCIENCE TIP: LOG SCALES ARE FOR QUITTERS WHO CAN'T
FIND ENOUGH PAPER TO MAKE THEIR POINT *PROPERLY*.

Don't Lose Your Point

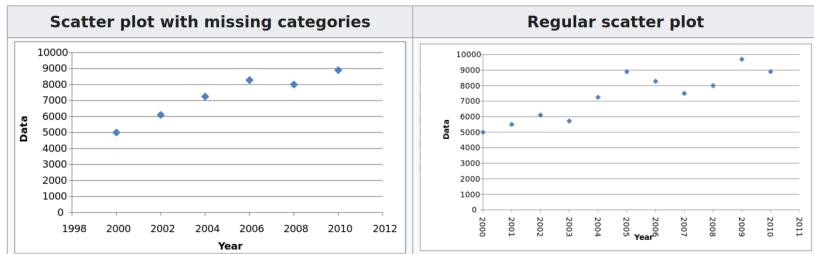


Dark Patterns – Truncated Axes



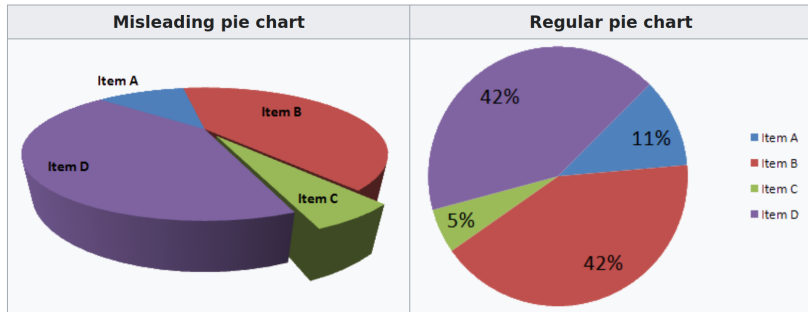
https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – Omitted Data



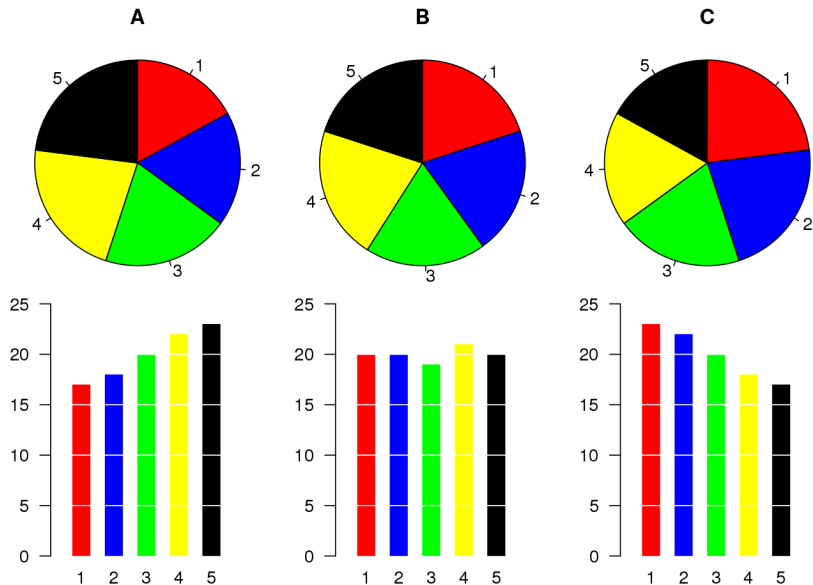
https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – 3D Pie Charts



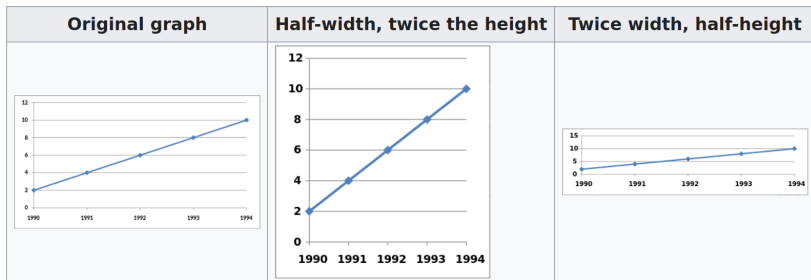
https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – Comparing Pie Charts



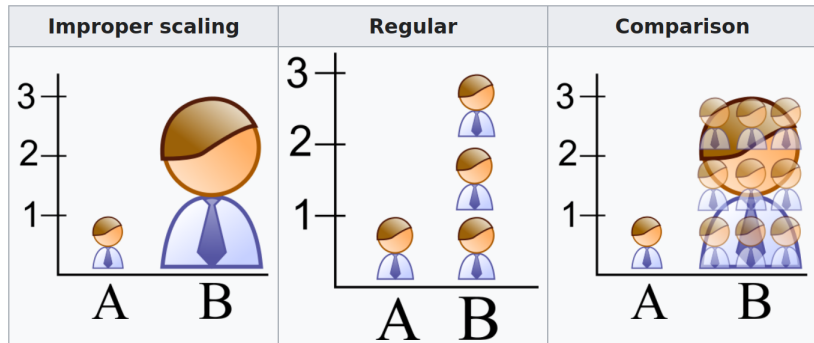
https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – Scaling Axes and Aspect Ratios



https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – Scaling Multiple Dimensions



https://en.wikipedia.org/wiki/Misleading_graph

- ▶ Colour is an important visualization element
- ▶ A colour palette defines a set of colours for a graph/plot.

Desirable Characteristics

- ▶ Colourful (range of values)
- ▶ Perceptually uniform (even perceptual distances)
- ▶ Robust to colourblindness (CVD)
- ▶ Pretty

Types of Colour Palettes

Sequential/Monochrome

- ▶ Varying hue, light to dark/deep
- ▶ Discrete or continuous
- ▶ Show progression for data with inherent order

Diverging



Sequential



Divergent

- ▶ From one colour to another via white or black
- ▶ Discrete or continuous
- ▶ Useful for showing deviations or extremes on either size of midpoint
- ▶ Data with meaningful center

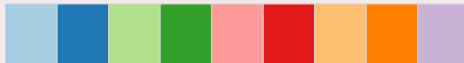
Diverging



Spectral

- ▶ Uses a number of different colours
- ▶ Discrete
- ▶ Data without inherent ordering
- ▶ Limited to few categories as colours become too similar

Spectral



CVD (Colour Vision Deficiency)

- ▶ Monochromatism
- ▶ Protanopia (missing "S-cone", blue)
- ▶ Deuteranopia (missing "M-cone", green)
- ▶ Tritanopia (missing "L-cone", red)

1 in 12 men have CVD

1 in 200 women have CVD

2.6 million Canadians are colour blind



MUN Faculty of Education Class Room

Copyright Memorial University of Newfoundland

Simulated Colour Vision Deficiencies



Monochromatism



Protanopia



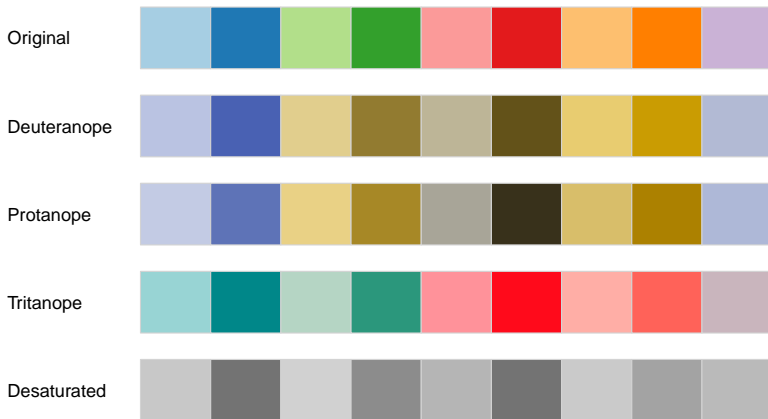
Deuteranopia



Tritanopia

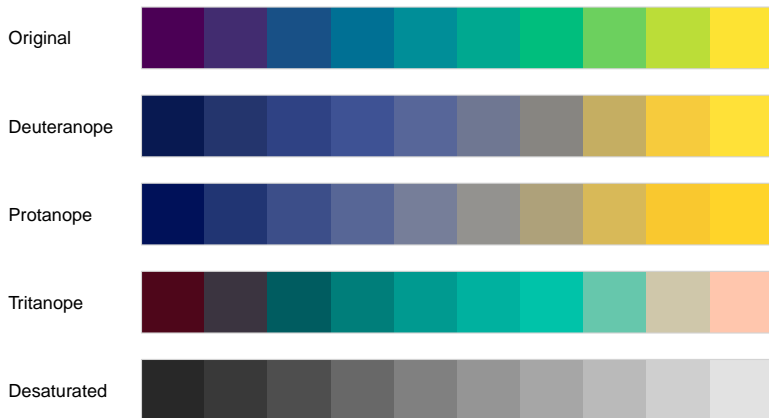
Example: Colourbrewer Palette "Paired"

Brewer Paired



Viridis Colour Palette

Viridis Palette



Popular Graphics Libraries and Frameworks

R

- ▶ GGPlot (and related libraries such as GGPatten)
- ▶ Plotly for R
- ▶ GGVis (for Dashboards)
- ▶ Shiny (for Dashboards)

Python

- ▶ Matplotlib
- ▶ Seaborn
- ▶ Plotly (Express, GO, Dash)
- ▶ Plotnine ("GGPlot for python")
- ▶ Shiny (for Dashboards)

Web & JS

- ▶ D3, ChartJS, GoogleCharts

Plot Elements

Map Data to Plot Elements

- ▶ X, Y axis
- ▶ Colour (point, line, fill)
- ▶ Transparency ("alpha")
 - ▶ Be aware of print versus screen or color vision deficiency
- ▶ Pattern (fill)
- ▶ Size, Weight/Width (point, line)
- ▶ Shape, Style (point, line)

Other Plot Elements

- ▶ Title, sub-title, captions
- ▶ Axis titles, axis labels and "ticks"
- ▶ Legend(s)

Example Dataset

- ▶ Government of Canada, Open Government Portal
- ▶ Fuel Consumption Ratings – Battery-electric vehicles – 2012–2023; last updated Oct 10, 2023
- ▶ <https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64>

Column	Data Type
Make	Discrete
Model	Discrete
Year	Numeric
Category	Discrete
City	Numeric ¹
Hwy	Numeric
Comb	Numeric
Range	Numeric ²

¹Fuel consumption in l/100km equivalent

²Range in km

Example Dataset – Read Data

R

```
# Load tidyverse package  
library(tidyverse)  
  
# Read the data set to a Tibble  
data <- read_csv('https://evermann.ca/busi4720/fuel.csv')  
  
# Ensure vehicle category is a factor (categorical)  
data$Fuel <- as.factor(data$Fuel)
```

Load Graphics Libraries

R

```
library(ggplot2)
library(ggpattern)
library(ggstream)
library(ggsci)
library(scales)
library(ggrepel)
library(ggradar)
```

Histogram

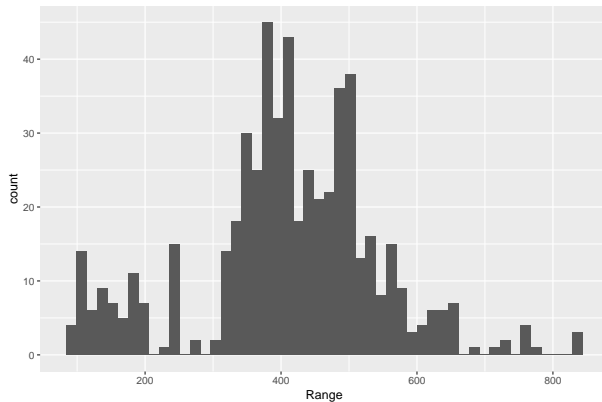
```
data |>
  ggplot(aes(x=Range)) +
  geom_histogram(bins=50)
```

- ▶ Aesthetic `aes()` determines mapping of data to plot elements
- ▶ Add "geoms" that determine the type of plot
- ▶ Geoms can have their own additional aesthetics and other options

```
ggsave("histogram.pdf", height=5, width=7.5, units='in')
```

- ▶ Save plot in different formats (PDF, PNG, JPEG, ...)

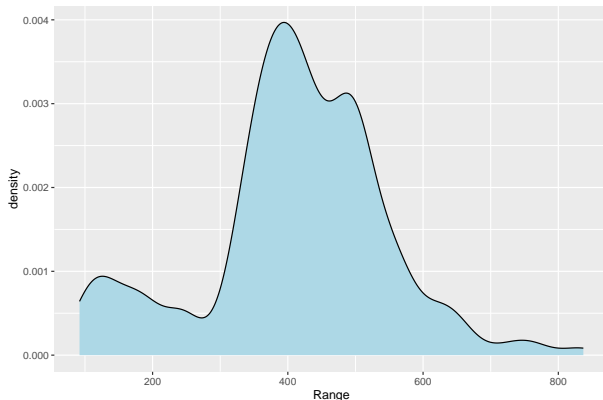
Histogram



- Shows frequencies or counts

Density Plot

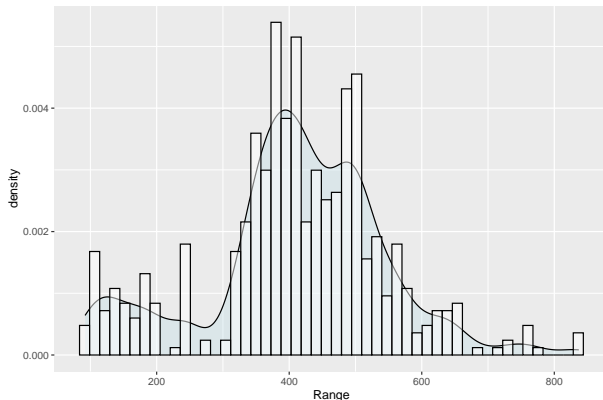
```
data |>  
  ggplot(aes(Range)) +  
  geom_density(kernel='gaussian', fill='lightblue')
```



- Shows probability density functions
- Kernel is used to smooth the curve (how to weight points)

Combining Geoms

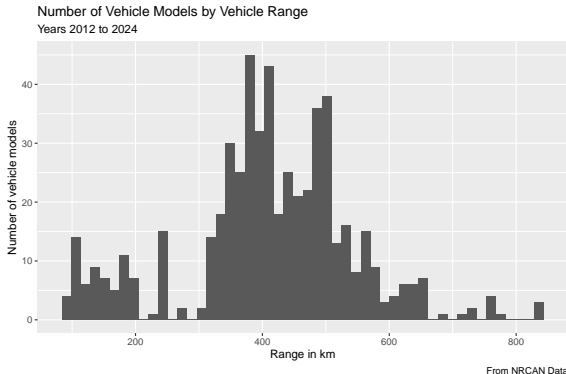
```
R  
data |> ggplot(aes(Range)) +  
  geom_density(kernel='gaussian',  
    alpha=0.25, fill='lightblue') +  
  geom_histogram(aes(y=after_stat(density)), bins=50,  
    alpha=0.5, fill='white', color='black')
```



- The `alpha` option controls transparency

Labelling

```
data |> ggplot(aes(x=Range)) +  
  geom_histogram(bins=50) +  
  labs(x = 'Range in km',  
       y = 'Number of vehicle models',  
       title='Number of Vehicle Models by Vehicle Range',  
       subtitle='Years 2012 to 2024',  
       caption='From NRCAN Data')
```



Hands-On Exercises

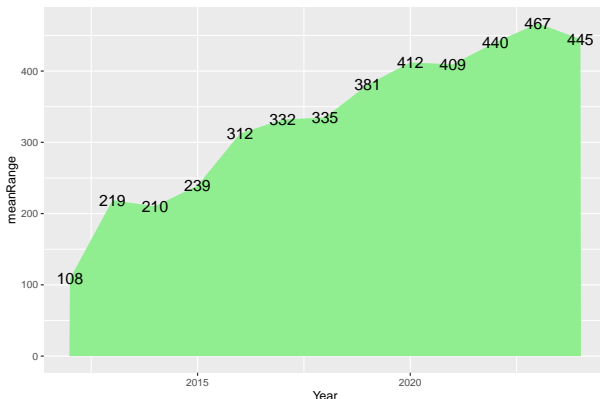
- 1 Read the EV fuel efficiency data set into R .
- 2 Create a blue histogram of highway efficiency with 25 bins.
- 3 Add labels for the axes, and add a title.

Tips

- ▶ Use the `read_csv()` function from the tidyverse library
- ▶ The column name is `Hwy`
- ▶ Use the `geom_histogram` geom
- ▶ Use the `bins=...` option
- ▶ Use the `fill='...'` option
- ▶ Use the `labs` geom for labels

Area Plot

```
R  
data %>% group_by(Year) %>%  
  summarize(meanRange = mean(Range)) %>% ungroup() %>%  
  ggplot(aes(Year, meanRange)) +  
    geom_area(fill='lightgreen') +  
    geom_text(aes(label=round(meanRange)), size=5)
```



- The `ggplot()` function is the last in a data processing pipeline

Column Chart

- ▶ Data must have one variable mapped to Y axis
- ▶ Data must have one variable mapped to fill colour

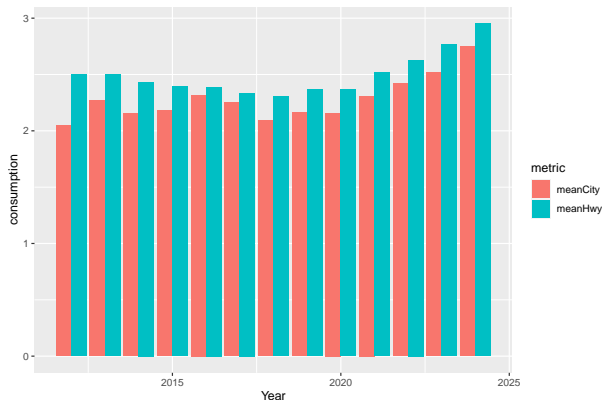
```
R
col.data <- data %>%
  group_by(Year) %>%
  summarize(
    meanCity = mean(City),
    meanHwy = mean(Hwy) %>%
  ungroup() %>%
  pivot_longer(
    cols=c('meanCity', 'meanHwy'),
    names_to='metric',
    values_to='consumption')
```

	Year	meanCity	meanHwy
	<dbl>	<dbl>	<dbl>
1	2012	2.05	2.5
2	2013	2.27	2.5
3	2014	2.16	2.43

	Year	metric	consumption
	<dbl>	<chr>	<dbl>
1	2012	meanCity	2.05
2	2012	meanHwy	2.5
3	2013	meanCity	2.27
4	2013	meanHwy	2.5
5	2014	meanCity	2.16
6	2014	meanHwy	2.43

Column Chart [cont'd]

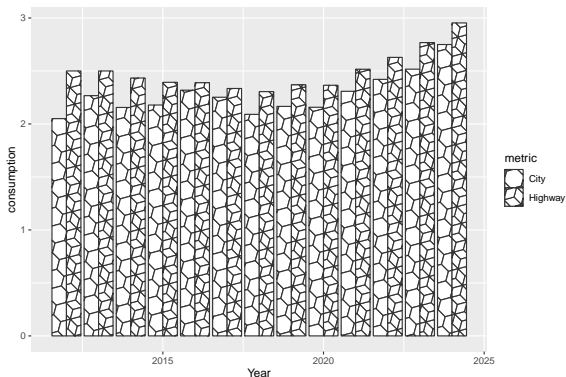
```
col.data |>  
  ggplot(aes(x=Year, y=consumption, fill=metric)) +  
    geom_col(position='dodge')
```



- The option `position = 'dodge'` places the columns next to each other, instead of on top of each other.

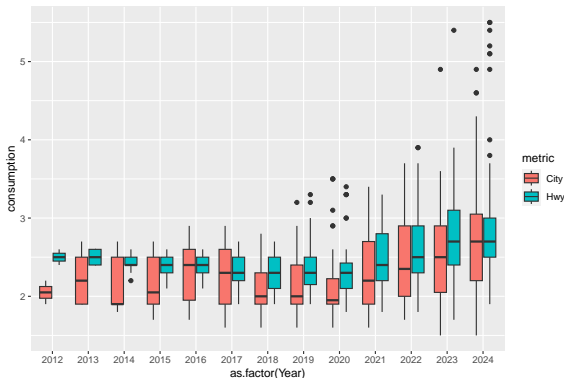
Column Chart with Patterns

```
R  
col.data |> ggplot(aes(x=Year, y=consumption)) +  
  geom_col_pattern(aes(pattern_type=metric),  
    pattern='polygon_tiling', pattern_angle=45,  
    pattern_fill='white', position='dodge') +  
  scale_pattern_type_manual(  
    values = c('hexagonal', 'rhombille'),  
    labels=c("City", "Highway"))
```



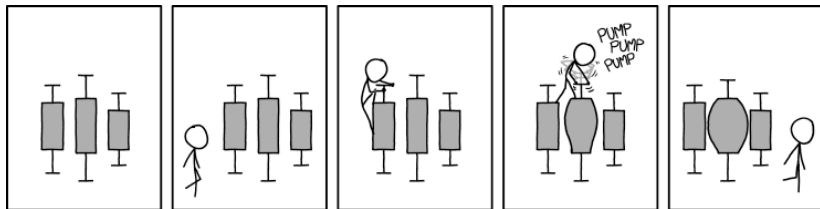
Box Plot

```
data |>
  pivot_longer(cols=c('City', 'Hwy'),
               names_to='metric', values_to='consumption') |>
  ggplot(aes(x=as.factor(Year), y=consumption, fill=metric)) +
  geom_boxplot()
```



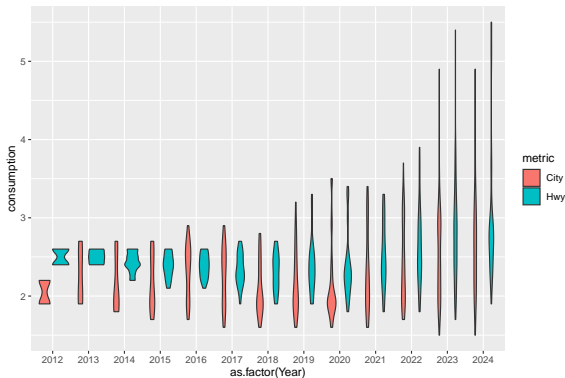
- Shows distribution
- Median
- 1st quartile Q_1
- 3rd quartile Q_3
- "Inter-quartile range"
- $IQR = Q_3 - Q_1$
- "Whiskers"
- $Q_3 + 1.5 \times IQR$
- $Q_1 - 1.5 \times IQR$
- "Outliers"

Boxplot (XKCD)



Violin Plot

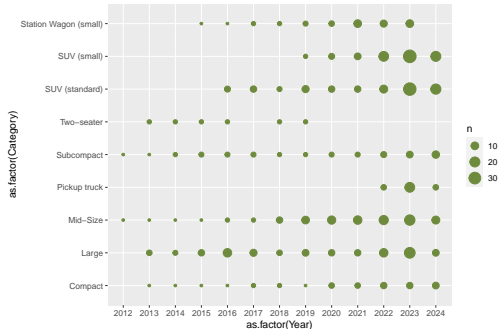
```
data |>
  pivot_longer(cols=c('City', 'Hwy'),
               names_to='metric',
               values_to='consumption') |>
  ggplot(aes(x=as.factor(Year), y=consumption, fill=metric)) +
  geom_violin()
```



- Shows detailed density
- But no summary statistics

Count Plot

```
data %>%  
ggplot(aes(as.factor(Year), as.factor(Category))) +  
  geom_count(color='darkolivegreen4') +  
  scale_y_discrete(  
    labels=c('Compact', 'Large', 'Mid-Size', 'Pickup truck',  
             'Subcompact', 'Two-seater', 'SUV (standard)',  
             'SUV (small)', 'Station Wagon (small)'))
```



► Explicit labels for discrete y axis

Jitter Plot

R

```
data |>
  filter(Year >= 2020) |>
  ggplot(aes(x=as.factor(Year), y=as.factor(Category),
             color=as.factor(Year))) +
  geom_jitter(width=0.2, height=0.2) +
  guides(color='none')
```

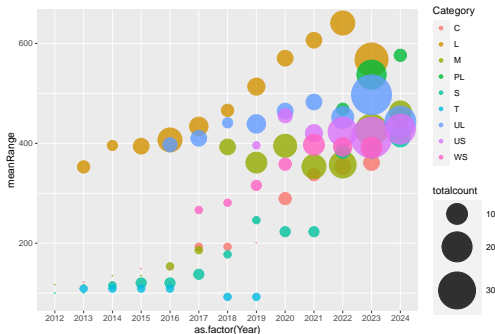


- Shows all data points (observations)
- No legend ("guide") for different colours
- Because colour is also mapped to Year, same as x axis

Points Plot

R

```
data |>
  group_by(Year, Category) |>
  summarize(totalcount=n(), meanRange=mean(Range)) |>
  ungroup () |>
  ggplot(aes(x=as.factor(Year), y=meanRange,
             size=totalcount, color=Category)) +
  geom_point(alpha=0.8) +
  scale_size_continuous(range=c(0, 20))
```

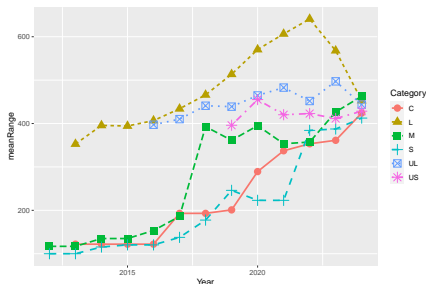


- Shows 4 variables
- Custom scale for point size

Lines and Points Plot

R

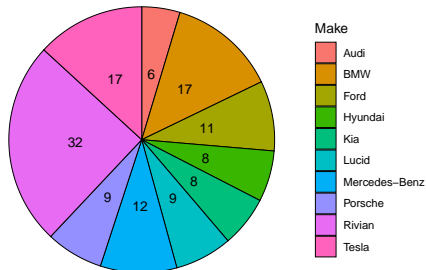
```
data |>
  filter(Category %in% c('C', 'L', 'M', 'S', 'US', 'UL')) |>
  group_by(Year, Category) |>
  summarize(meanRange = mean(Range)) |>
  ungroup() |>
  ggplot(aes(x=Year, y=meanRange,
             color=Category, shape=Category, linetype=Category)) +
  geom_line(size=1) +
  geom_point(size=4)
```



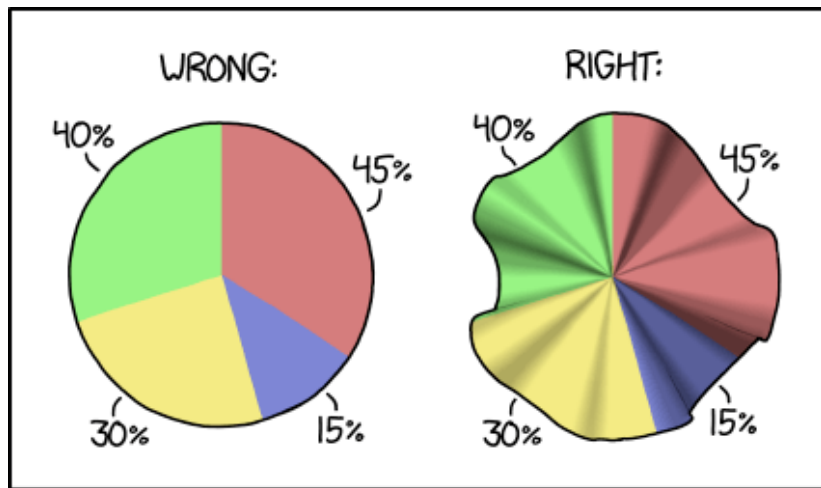
- ▶ 2 Geoms, line and point
- ▶ Category mapped to 3 plot elements

Pie Chart

```
data |>
  filter(Year==2023) |>
  group_by(Make) |>
  summarize(totalcount = n()) |>
  filter(totalcount >= 5) |>
  ungroup() |>
  ggplot(aes(x='', y=totalcount,
             fill=Make)) +
  geom_bar(stat='identity',
           color='black',
           size=0.25, width=1) +
  coord_polar('y',
              direction=-1,
              start=0) +
  geom_text(aes(
    label=ifelse(totalcount >= 5,
                  totalcount, ''),
    position = position_stack(
      vjust=0.5)) +
  theme_void()
```



Pie Charts (XKCD)



HOW TO MAKE A PIE CHART IF YOUR
PERCENTAGES DON'T ADD UP TO 100

Radar Plot

Prepare data:

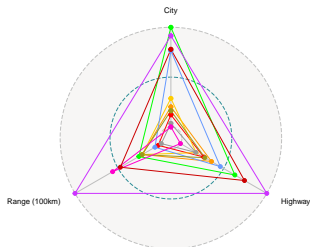
```
R
radardata <- data |>
  filter(Year == 2023) |> group_by(Make) |>
  summarize(meanCity = 1/mean(City),
             meanHwy = 1/mean(Hwy),
             meanRange = mean(Range)/100,
             nModels = n()) |>
  filter(nModels >= 5) |> ungroup() |>
  select(-nModels) |>
  mutate(across(-Make, rescale))
```

- The line `mutate(across(-Make, rescale))` rescales all columns except `Make` so values are between `[0, 1]`

```
Text
# A tibble: 10 x 4
  Make          meanCity meanHwy meanRange
  <chr>          <dbl>   <dbl>    <dbl>
1 Audi           0.122     0.270    0.0357
2 BMW            0.202     0.360    0.219
3 Ford           0.287     0.182    0.220
4 Hyundai        1         0.630    0.260
```

Radar Plot

```
R  
radardata |>  
  ggradar(  
    axis.labels=c(  
      'City',  
      'Highway',  
      'Range'),  
    values.radar='',  
    group.line.width=0.75,  
    group.point.size=3) +  
    scale_color_ucscgb()
```

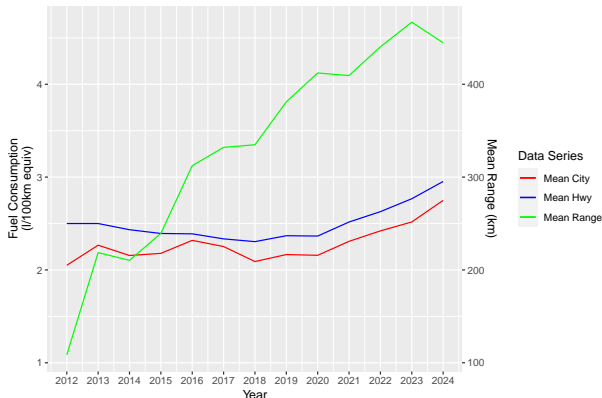


- Shows a comparison of different objects on a number of dimensions/aspects.
- Changed the colour palette to UCS CGB colours.

Lines with Multiple Axes

```
data |>
  group_by(Year) |>
  summarize(meanCity = mean(City),
            meanHwy = mean(Hwy),
            meanRange = mean(Range)/100) |>
  ungroup() |>
  ggplot(aes(x=Year)) +
  geom_line(aes(y=meanCity, color='Mean City')) +
  geom_line(aes(y=meanHwy, color='Mean Hwy')) +
  geom_line(aes(y=meanRange, color='Mean Range')) +
  scale_color_manual(name='Data Series',
                    values=c('Mean City' = 'red',
                              'Mean Hwy' = 'blue',
                              'Mean Range' = 'green')) +
  scale_y_continuous(labels=scales::comma,
                    name="Fuel Consumption\n(l/100km equiv)",
                    sec.axis=sec_axis(~ .*100,
                                      labels=scales::comma,
                                      name="Mean Range (km)")) +
  scale_x_continuous(breaks=seq(from=2012,to=2024,by=1))
```

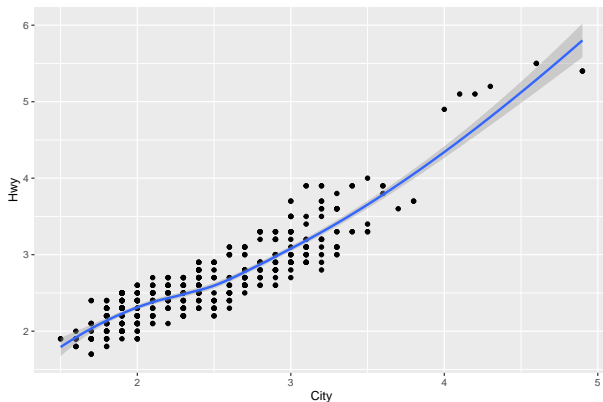
Lines with Multiple Axes



- ▶ Three geoms
- ▶ Manual colour scale
- ▶ Y axis scale with primary label
- ▶ Secondary axis with `sec.axis` option, scale factor to primary axis, and secondary label

Local Regression Smoothing Plot

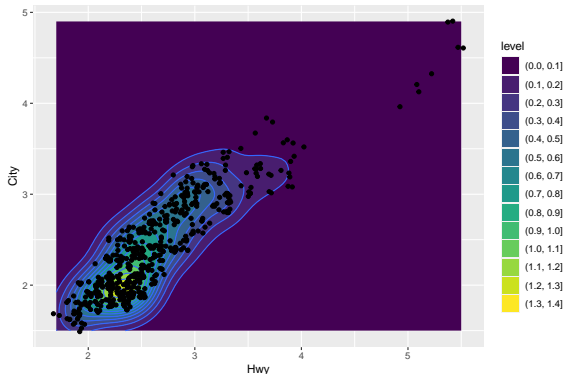
```
data |>  
  ggplot(aes(City, Hwy)) +  
    geom_point() +  
    geom_smooth()
```



- ▶ Two geoms
- ▶ Local regression line
- ▶ Uncertainty interval around regression line

2D Density Plot

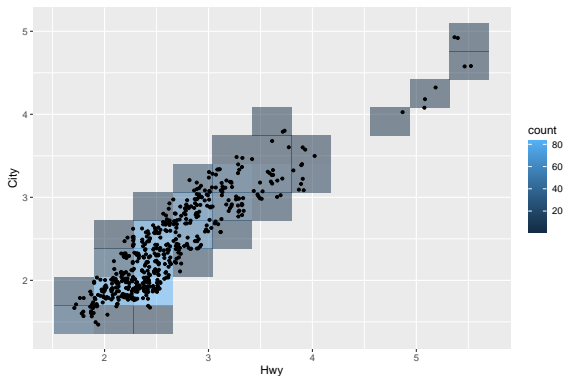
```
data |>  
  ggplot(aes(x=Hwy, y=City)) +  
    geom_density_2d_filled() +  
    geom_density_2d() +  
    geom_point(position='jitter')
```



- ▶ Shows (co-)distribution of observations
- ▶ Generalization of (1D) density plot
- ▶ Three geoms
- ▶ One for outline, one for fill, one for points

2D Bin Plot

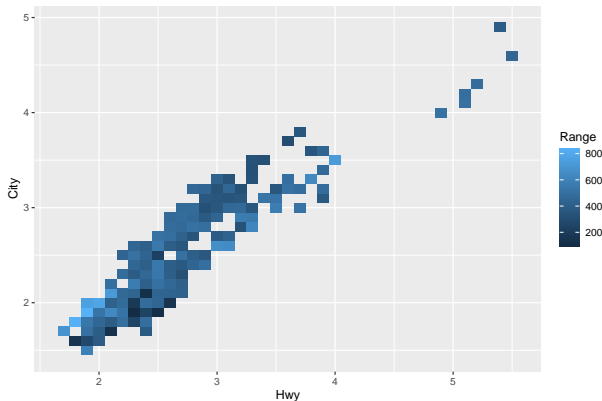
```
R  
data %>%  
  ggplot(aes(x=Hwy, y=City)) +  
    geom_bin2d(alpha=0.5, bins=10) +  
    geom_point(color="black", size=1, position='jitter')
```



- Shows (co-)distribution of observations
- Generalization of (1D) histogram plot

3D Raster/Tile Plot

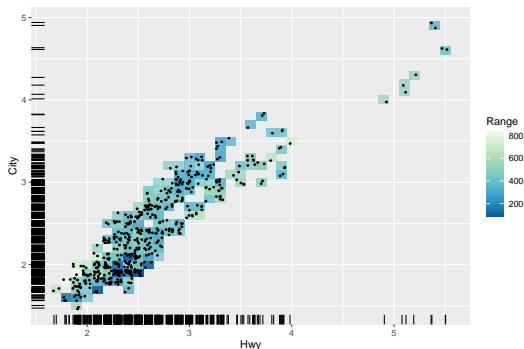
```
data |>  
  ggplot(aes(x=Hwy, y=City, fill=Range)) +  
  geom_tile()
```



► Plot of three variables

Complex Example – 3D Raster Plot with Rug

```
data |>  
  ggplot(aes(x=Hwy, y=City, fill=Range)) +  
    geom_tile() +  
    geom_point(size=0.5, position='jitter') +  
    geom_rug(position='jitter') +  
    scale_fill_distiller(palette=4, direction=-1)
```



- Geom for tiling
- Geom for points/observations
- Geom for "rugs"
- Custom colour scale

Hands-On Exercises

Using the Pagila database data from

<https://evermann.ca/busi4720/rentals.csv>, create

- 1 A histogram and/or density chart of film length by film category
- 2 A column chart of the mean rental payments for films by film category
- 3 A scatter plot of total rental payments by year and week
 - ▶ Add a local regression line to this plot
- 4 A pie or donut chart of rental counts by film rating

Tips:

- ▶ Use the `read_csv()` function to read from a URL
- ▶ The data is de-normalized, use the `unique()` function to get accurate film counts
- ▶ Use the `year()` and `week()` functions from the `lubridate` package (another package of the Tidyverse set)