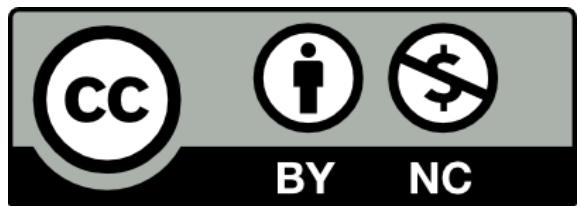
Business 4720 - Class 7

Data Visualization with R

Joerg Evermann

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



MEMORIAL UNIVERSITY

This Class

What You Will Learn:

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

MEMORIAL UNIVERSITY

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!)

MEMORIAL UNIVERSITY

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

MEMORIAL UNIVERSITY

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)

MEMORIAL UNIVERSITY

Honesty in Visualization

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

MEMORIAL UNIVERSITY

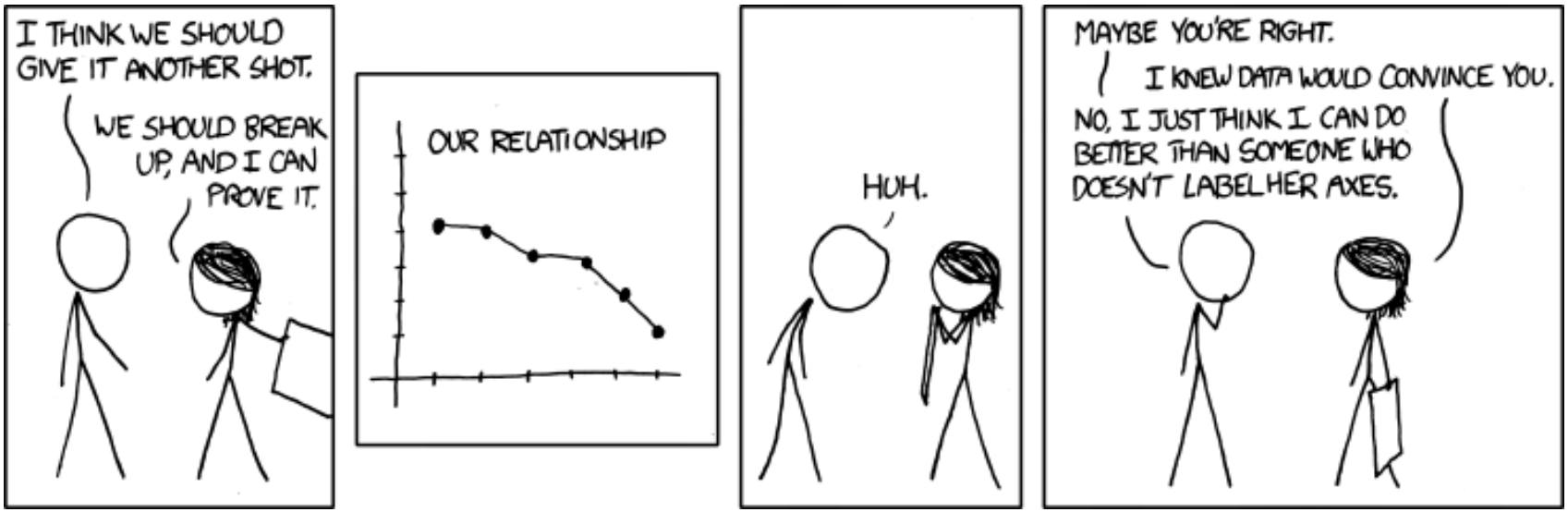
Honesty in Visualization

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

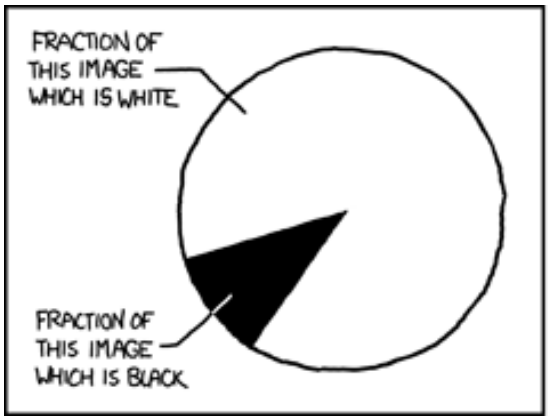
MEMORIAL UNIVERSITY

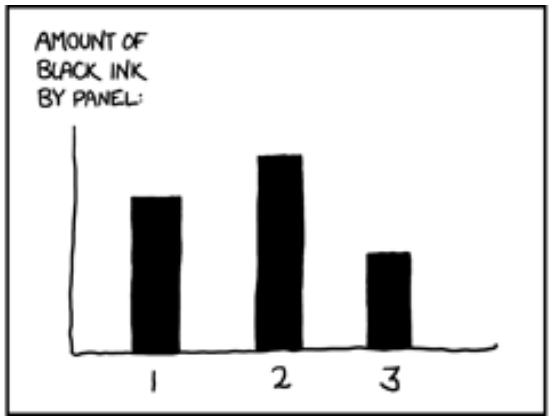
Label your Axes (XKCD)

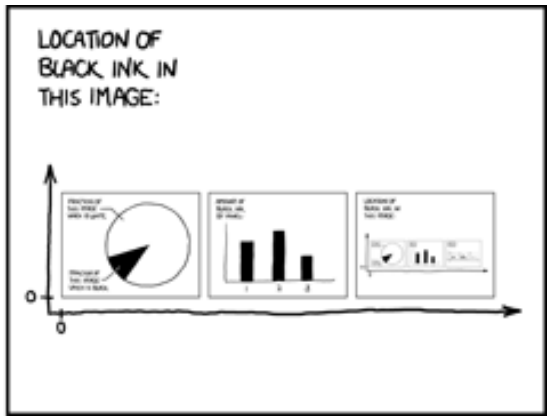


MEMORIAL UNIVERSITY

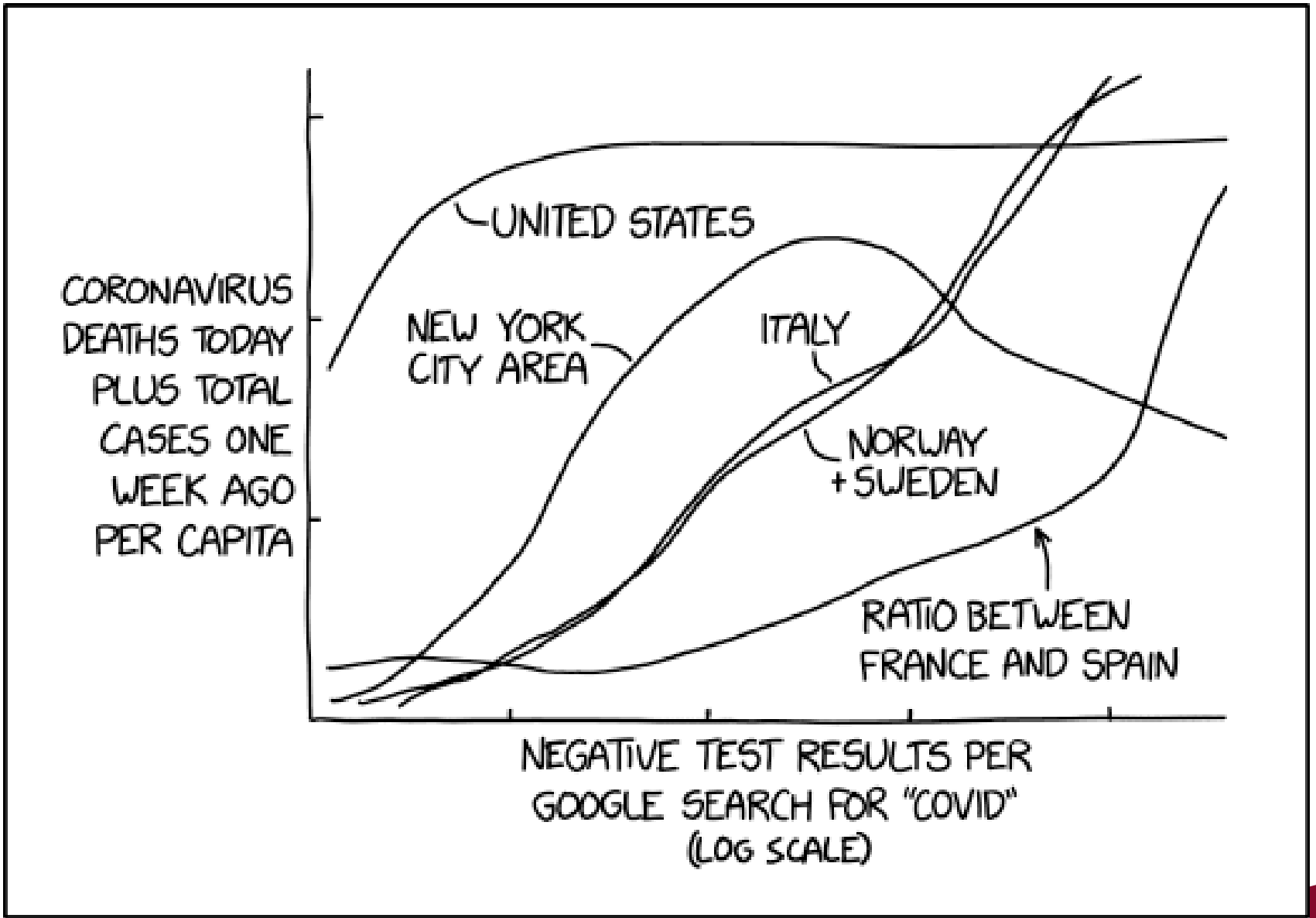
Use Meaningful Data (XKCD)



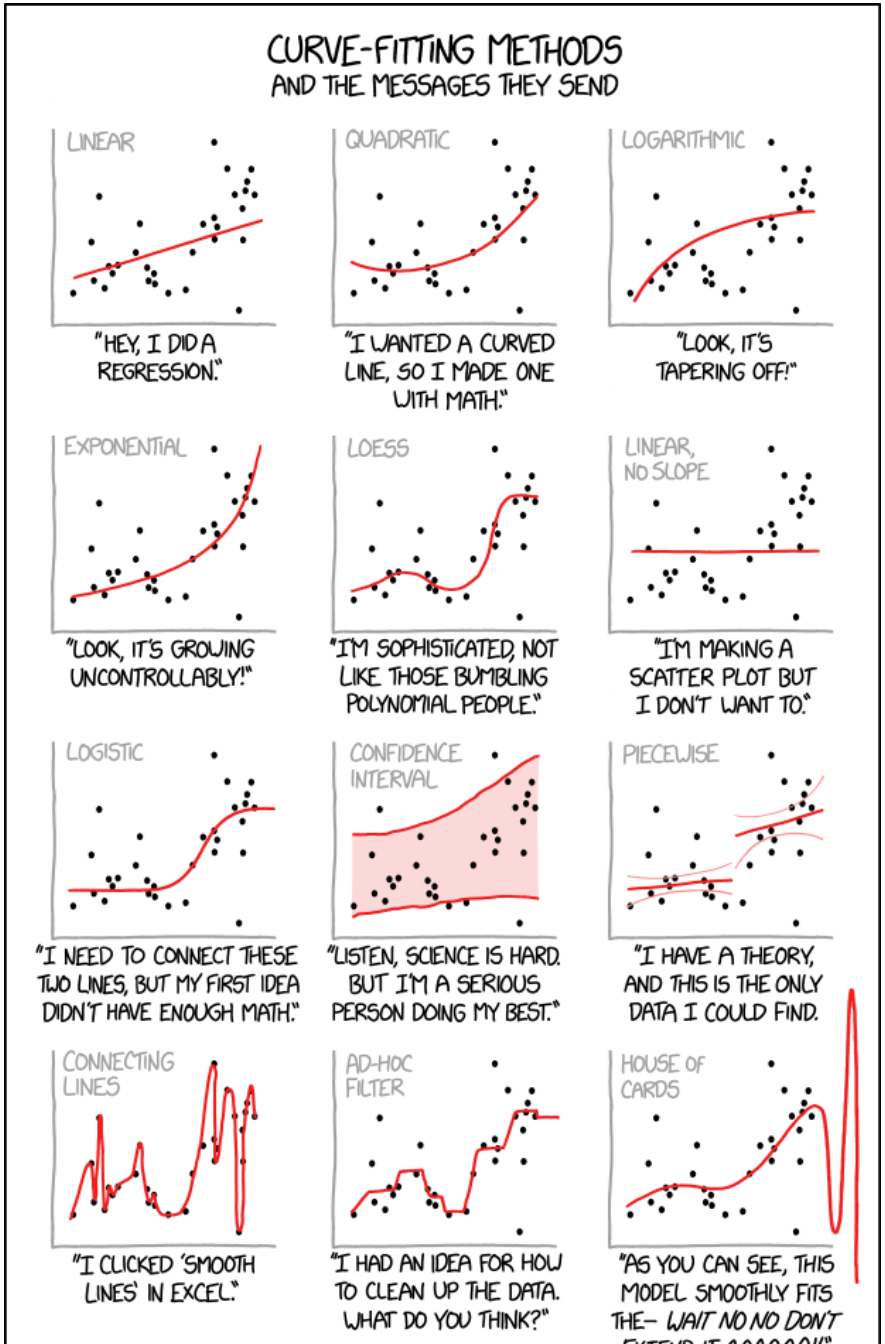




Use Related Data (XKCD)

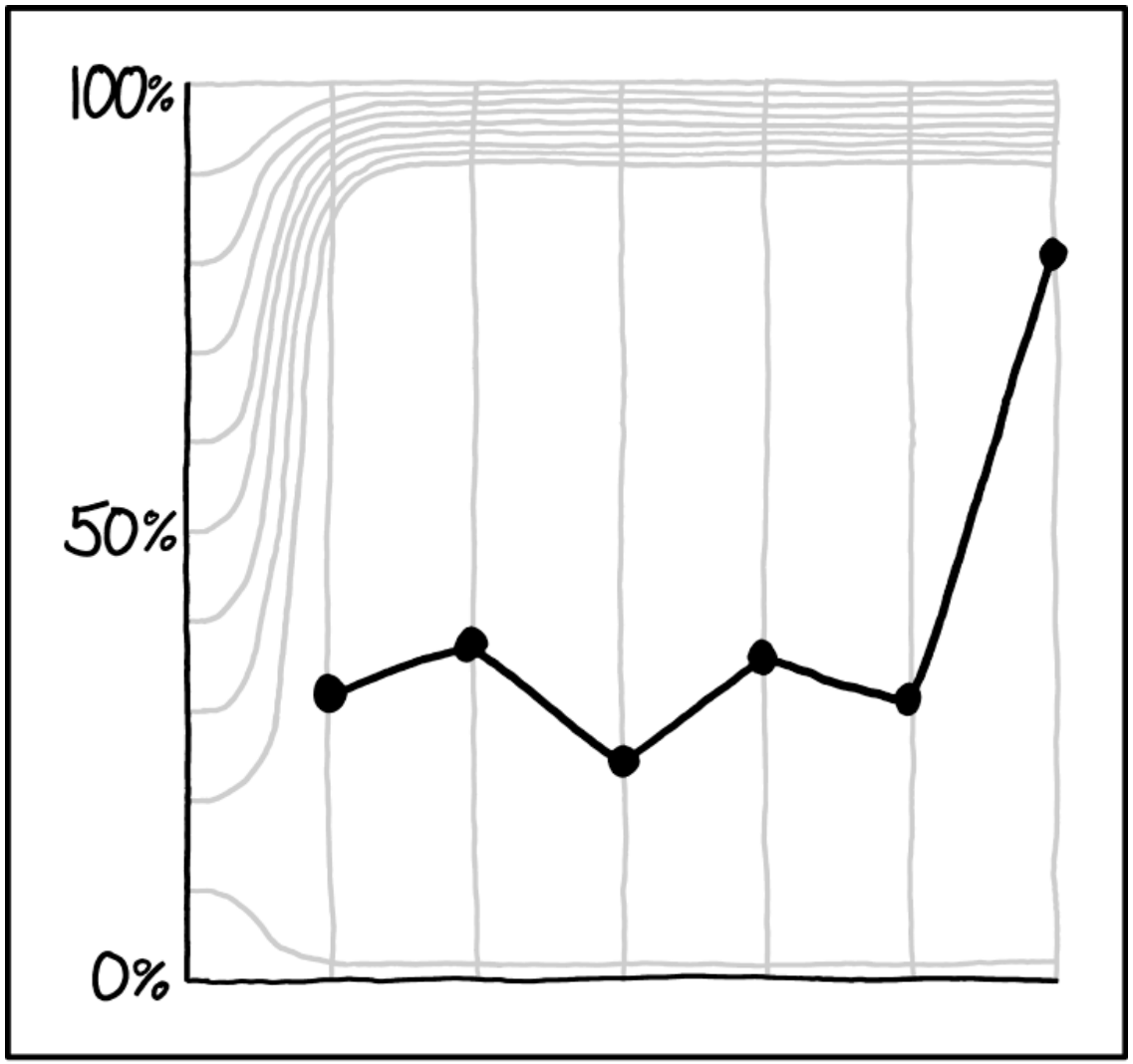


Do Not Mislead (XKCD)



MEMORIAL UNIVERSITY

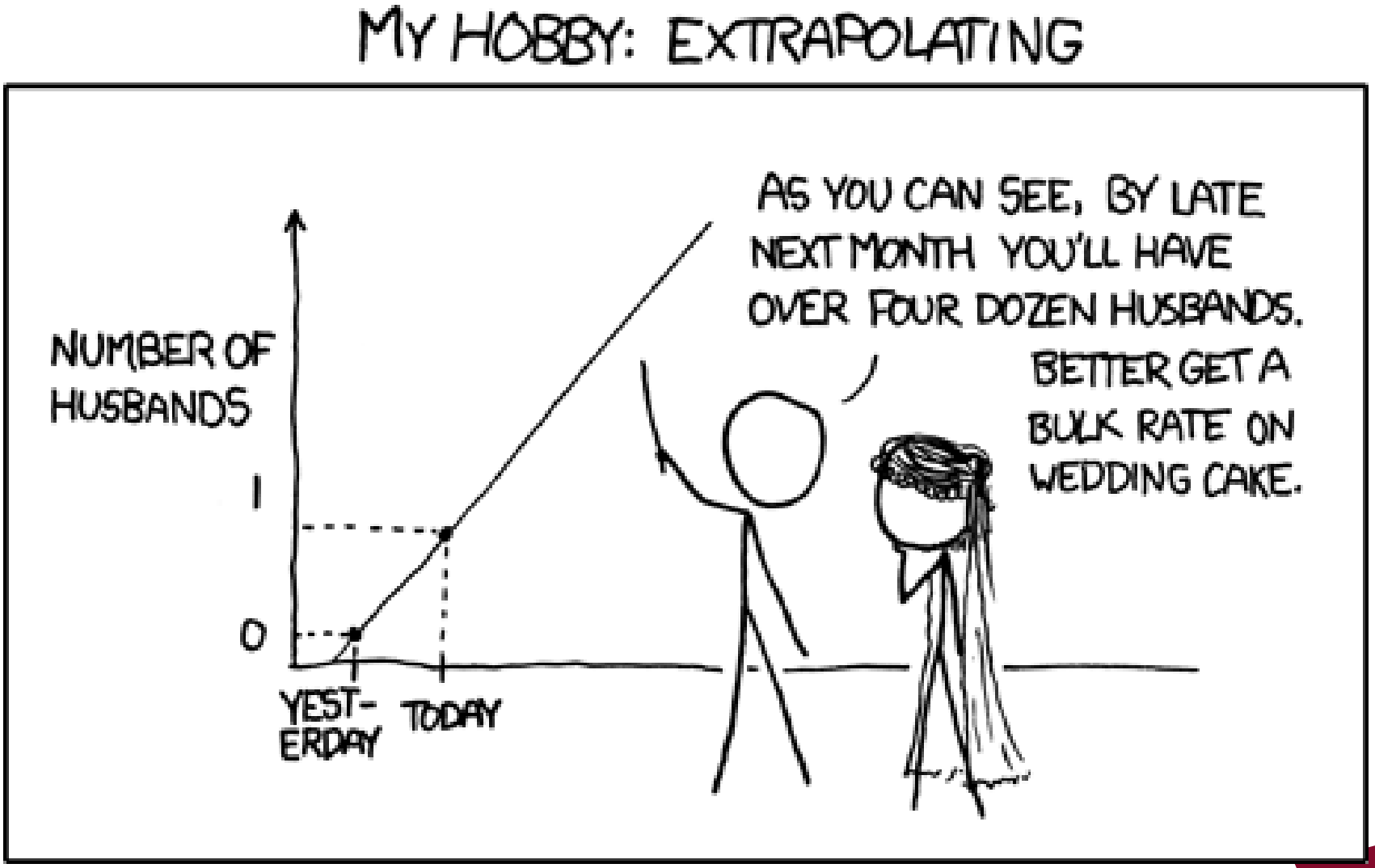
Choose Your Axes Meaningfully



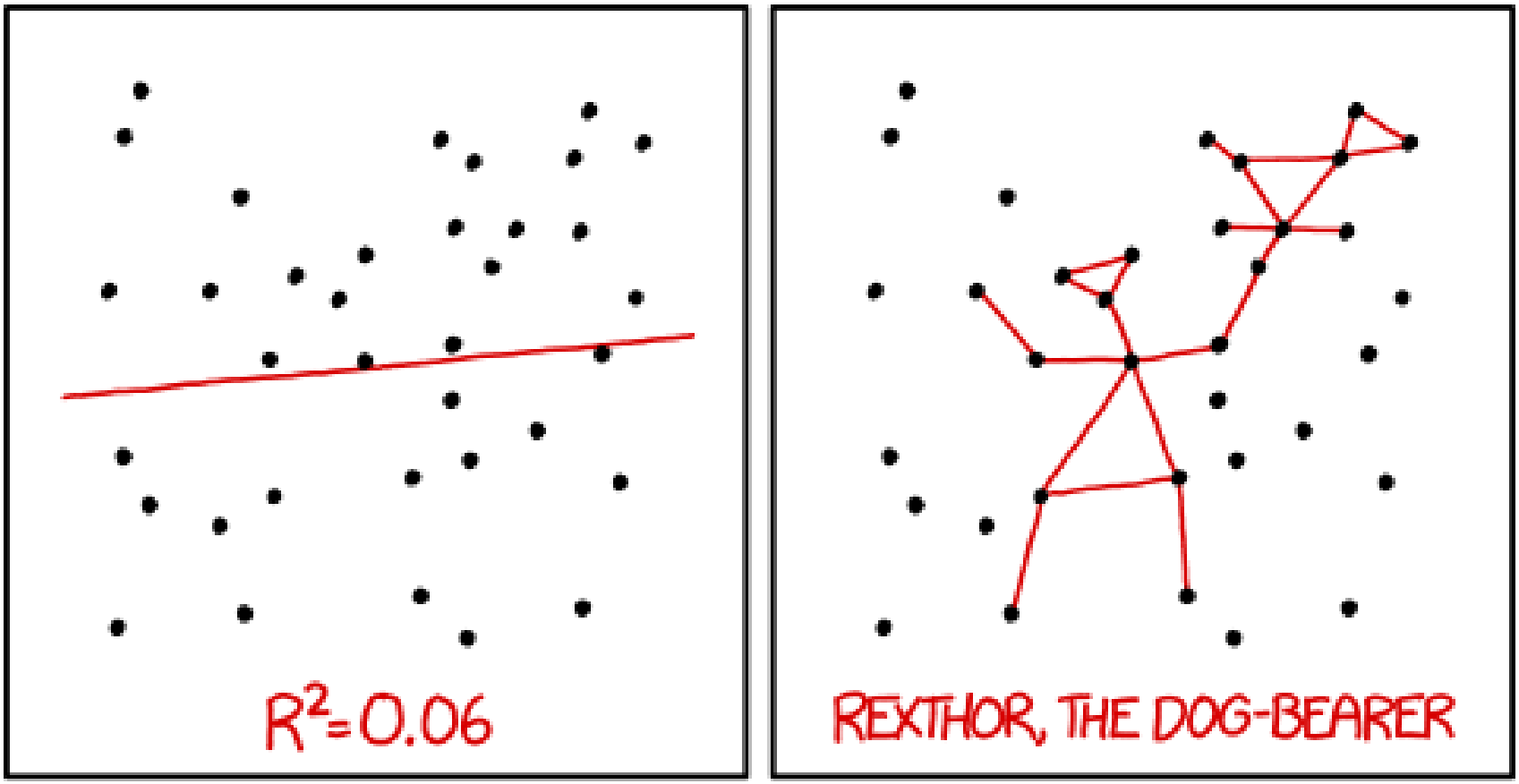
PEOPLE HAVE WISED UP TO THE "CAREFULLY CHOSEN Y-AXIS RANGE" TRICK, SO WE MISLEADING GRAPH MAKERS HAVE HAD TO GET CREATIVE.

MEMORIAL UNIVERSITY

Be Careful When Extrapolating (XKCD)



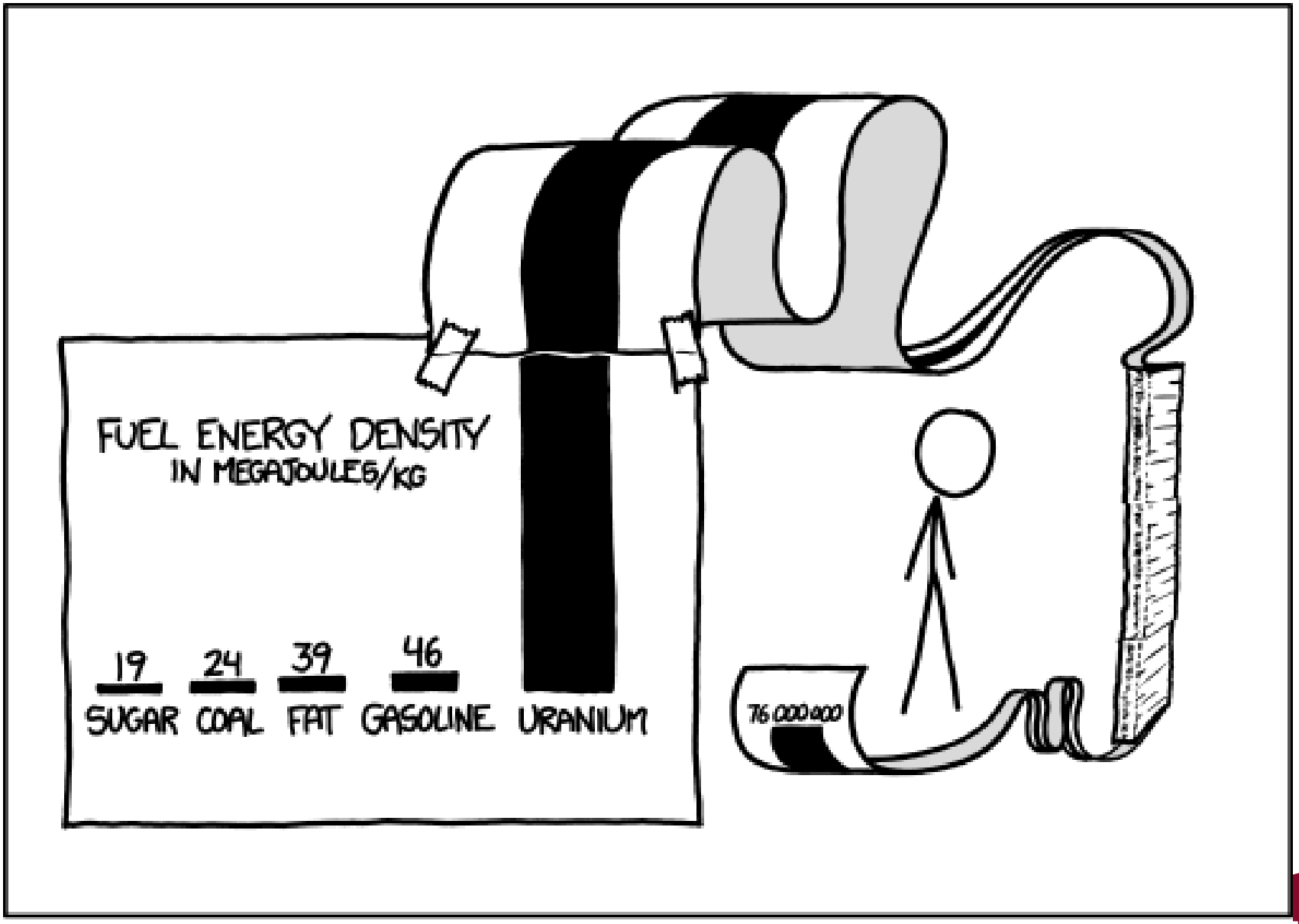
Verify Trends (XKCD)



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.

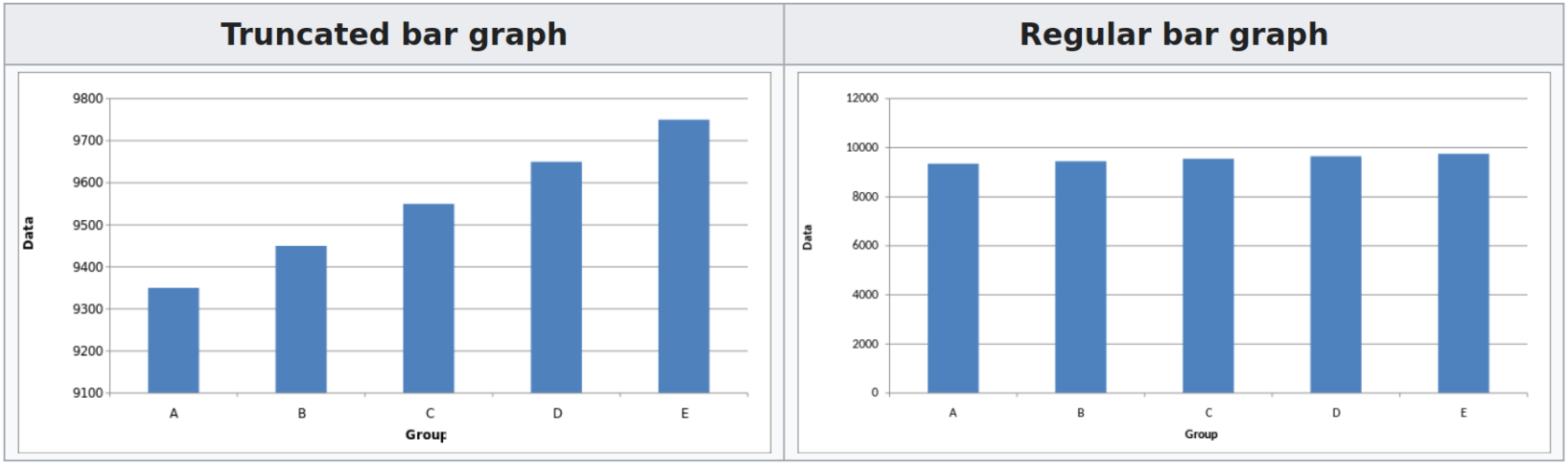
MEMORIAL UNIVERSITY

Use Appropriate Scales (XKCD)



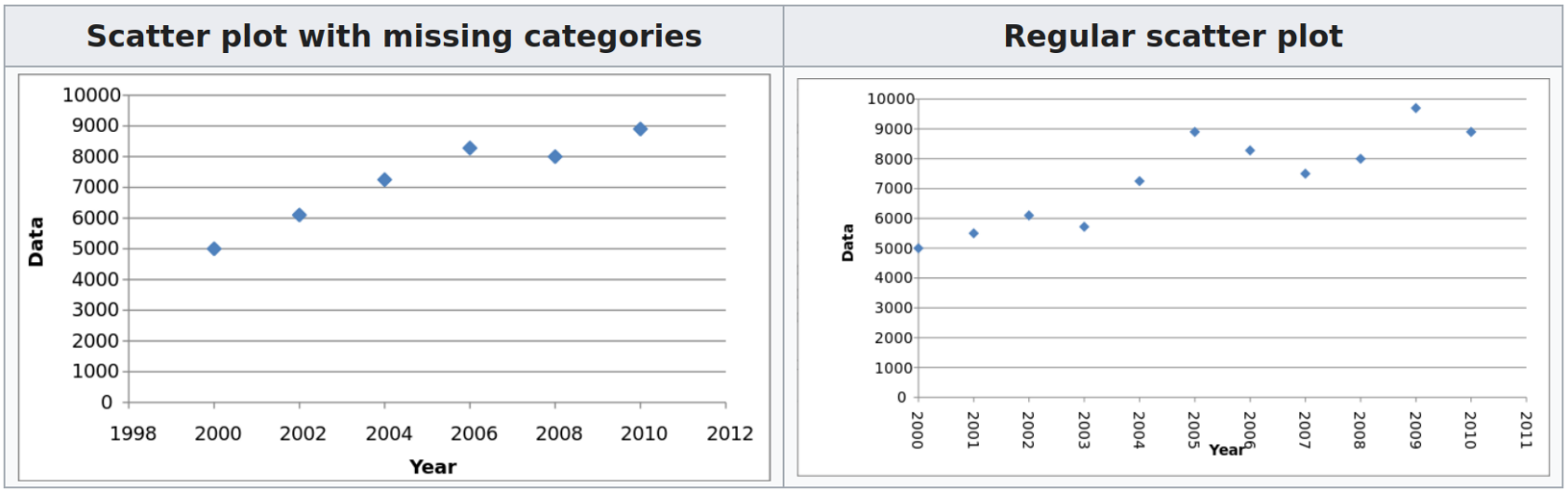
Don’t Lose Your Point

Dark Patterns – Truncated Axes



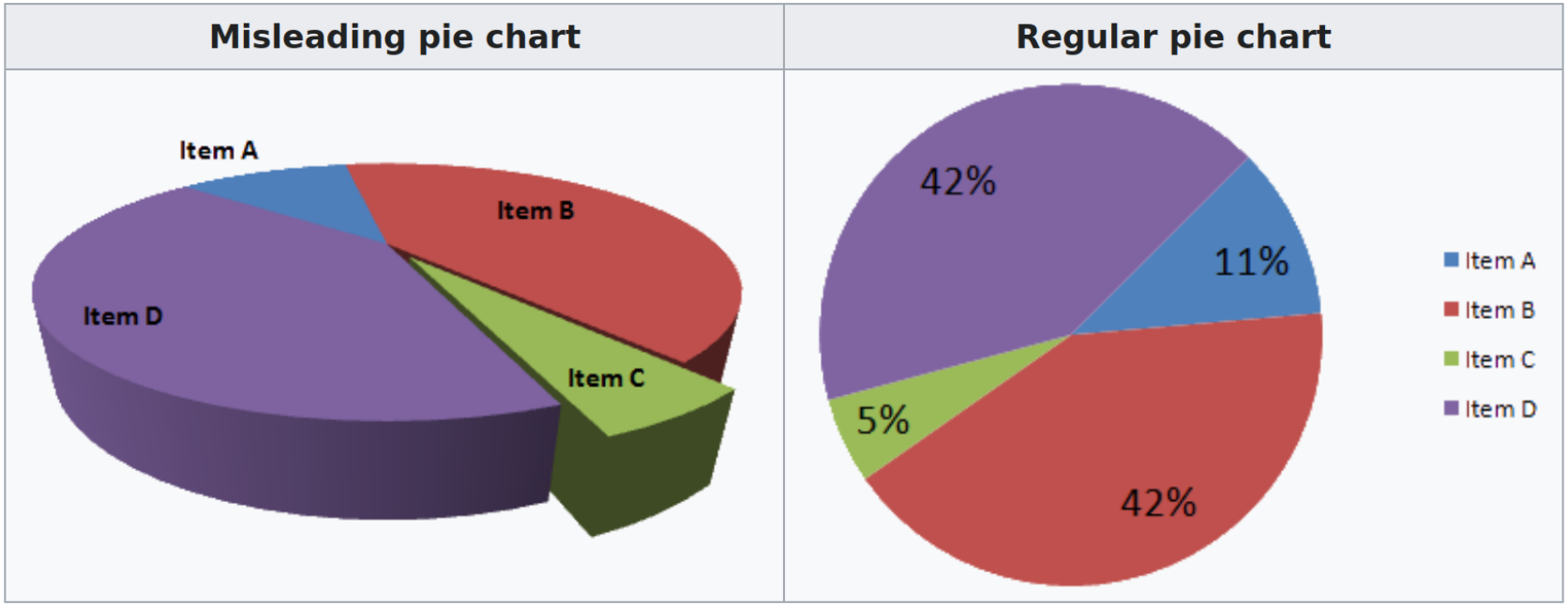
https://en.wikipedia.org/wiki/Misleading_graph

Dark Patterns – Omitted Data



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

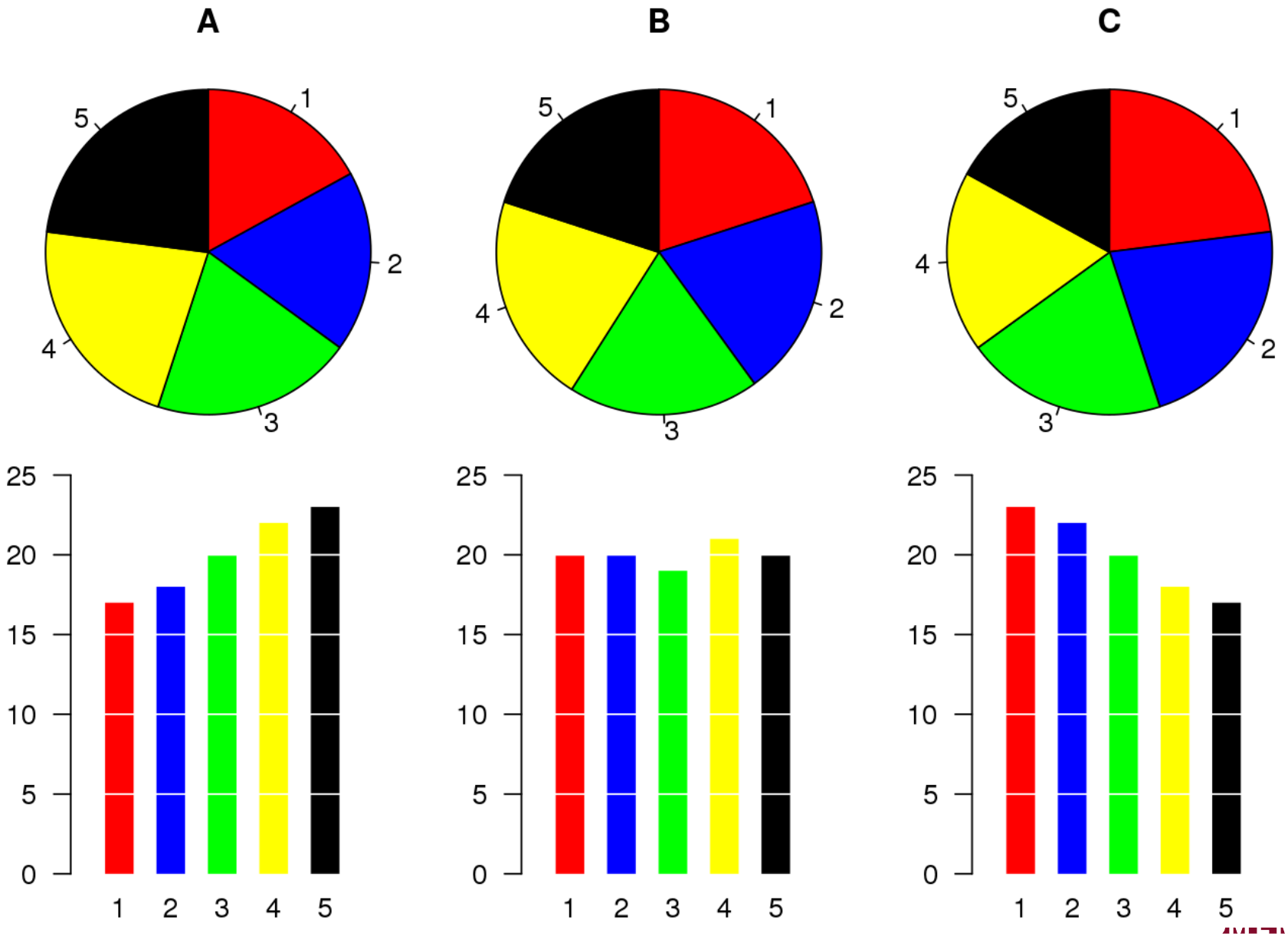
Dark Patterns – 3D Pie Charts



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

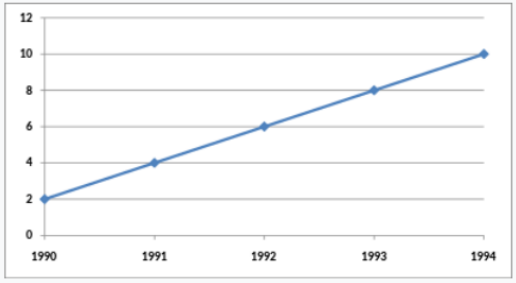
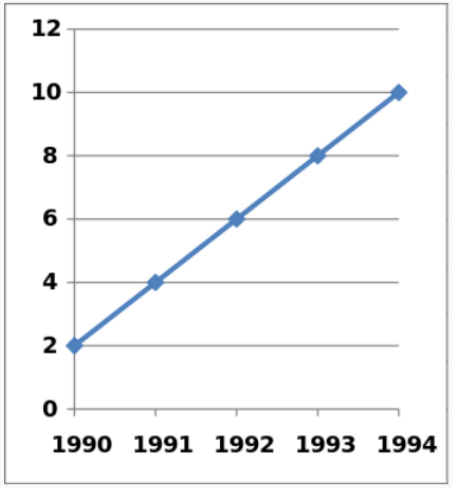
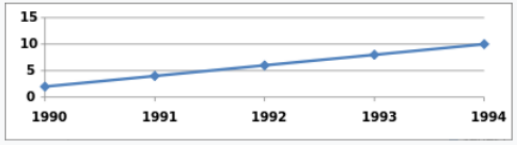
MEMORIAL UNIVERSITY

Dark Patterns – Comparing Pie Charts



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

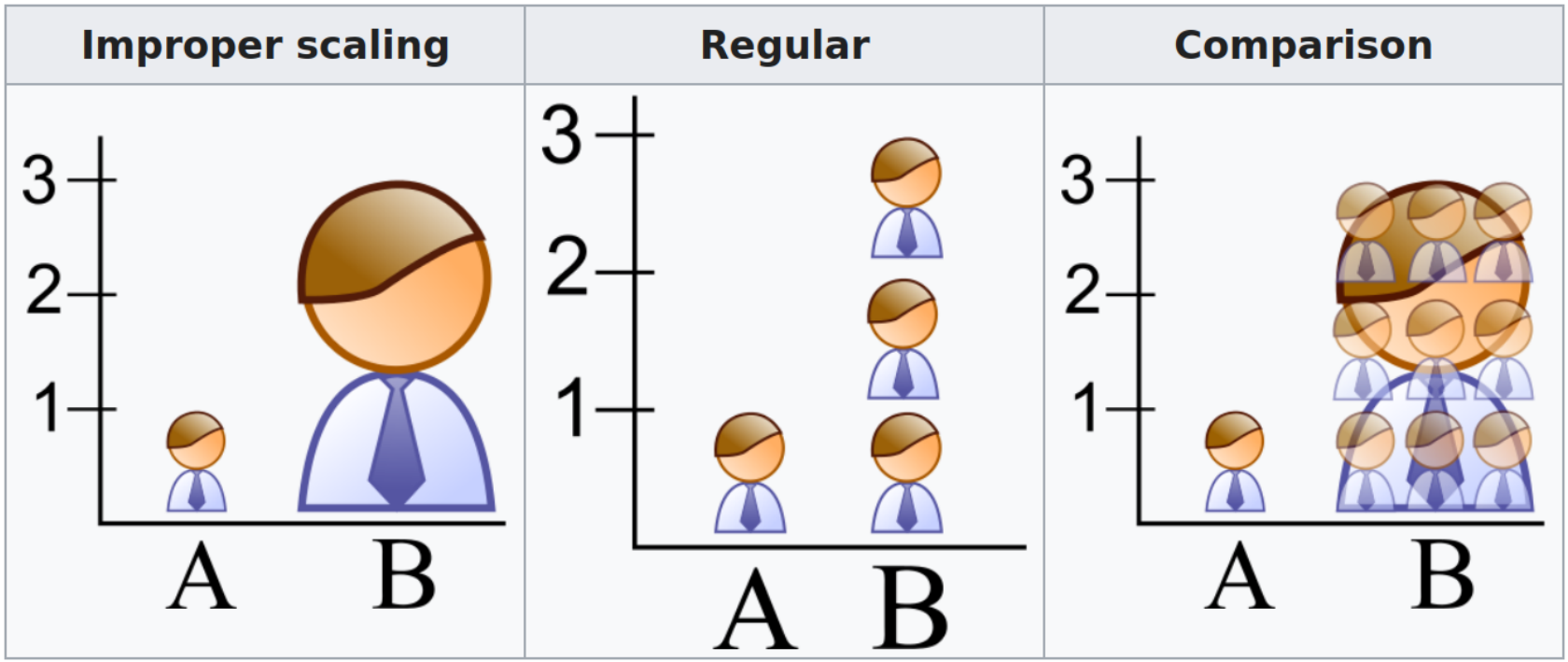
Dark Patterns – Scaling Axes and Aspect Ratios

Original graph	Half-width, twice the height	Twice width, half-height
		

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

MEMORIAL UNIVERSITY

Dark Patterns – Scaling Multiple Dimensions



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

Special Types of Data and Visual Analytics

- Streaming data
  - Continually changing
  - Limited buffers/windows
- Spatial, geographic, map data
  - Geo aware, irregular map boundaries, image overlays
- Network data
  - Vertices and vertex types, edges and edge types
- Text data
  - Unstructured text, e.g. from social media or web sites

MEMORIAL UNIVERSITY

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 (pint, line)

Other Plot Elements

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

Colour Palettes

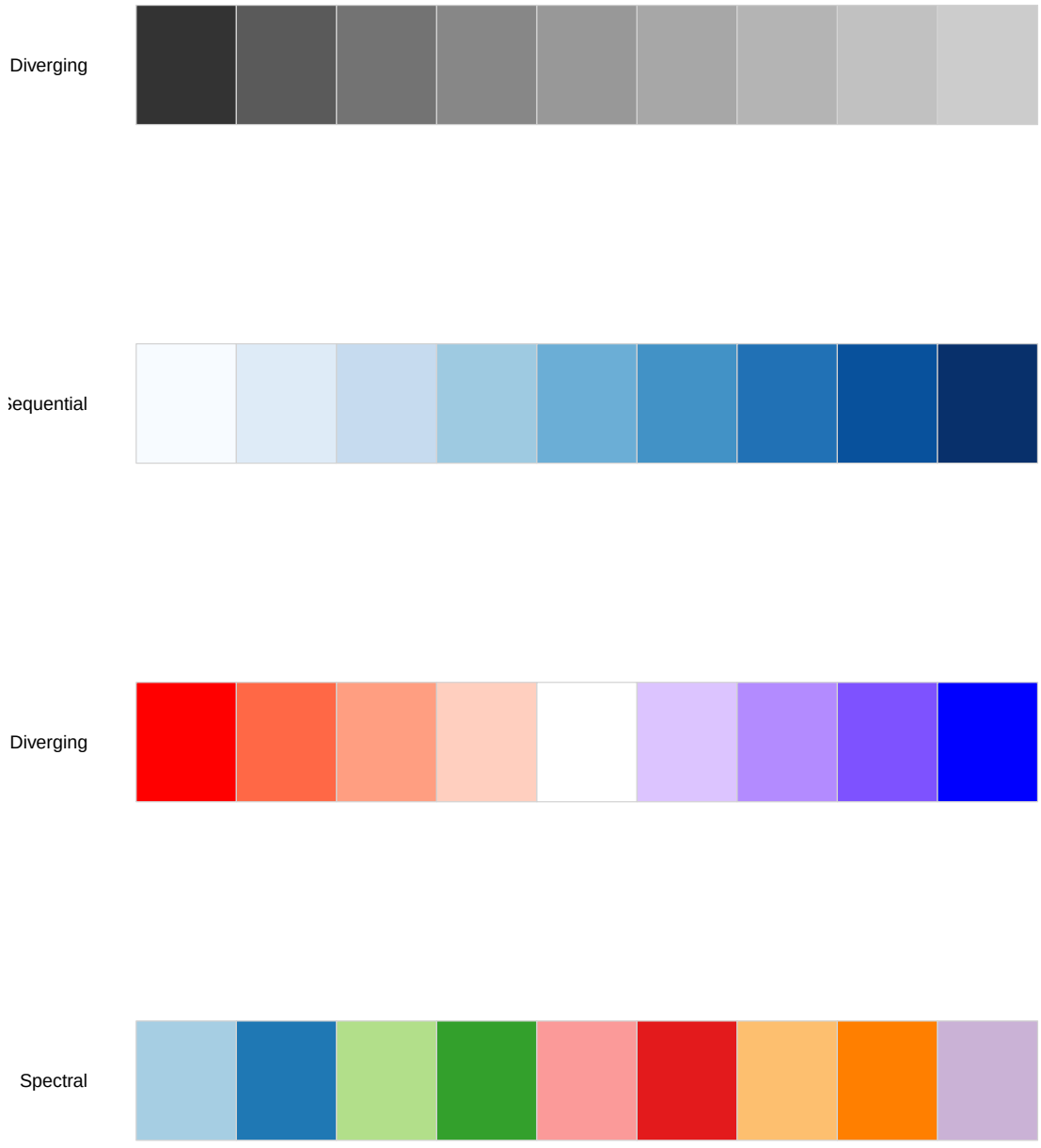
Desirable Characteristics

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

Typical of Colour Palettes

- **Monochrome/Sequential**, i.e. light to dark within a single colour
- **Divergent**, i.e. from one colour to another via white
- **Spectral**, uses a large number of colours
- **Bivariate**, e.g. combination or RGB and CMY

Colour palettes may be continuous, discrete, or categorical


Diverging
Sequential
Diverging
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

MEMORIAL UNIVERSITY

Original

MUN Faculty of Education Class Room

MEMORIAL UNIVERSITY

Simulated Colour Vision Deficiencies

Monochromatism

Protanopia

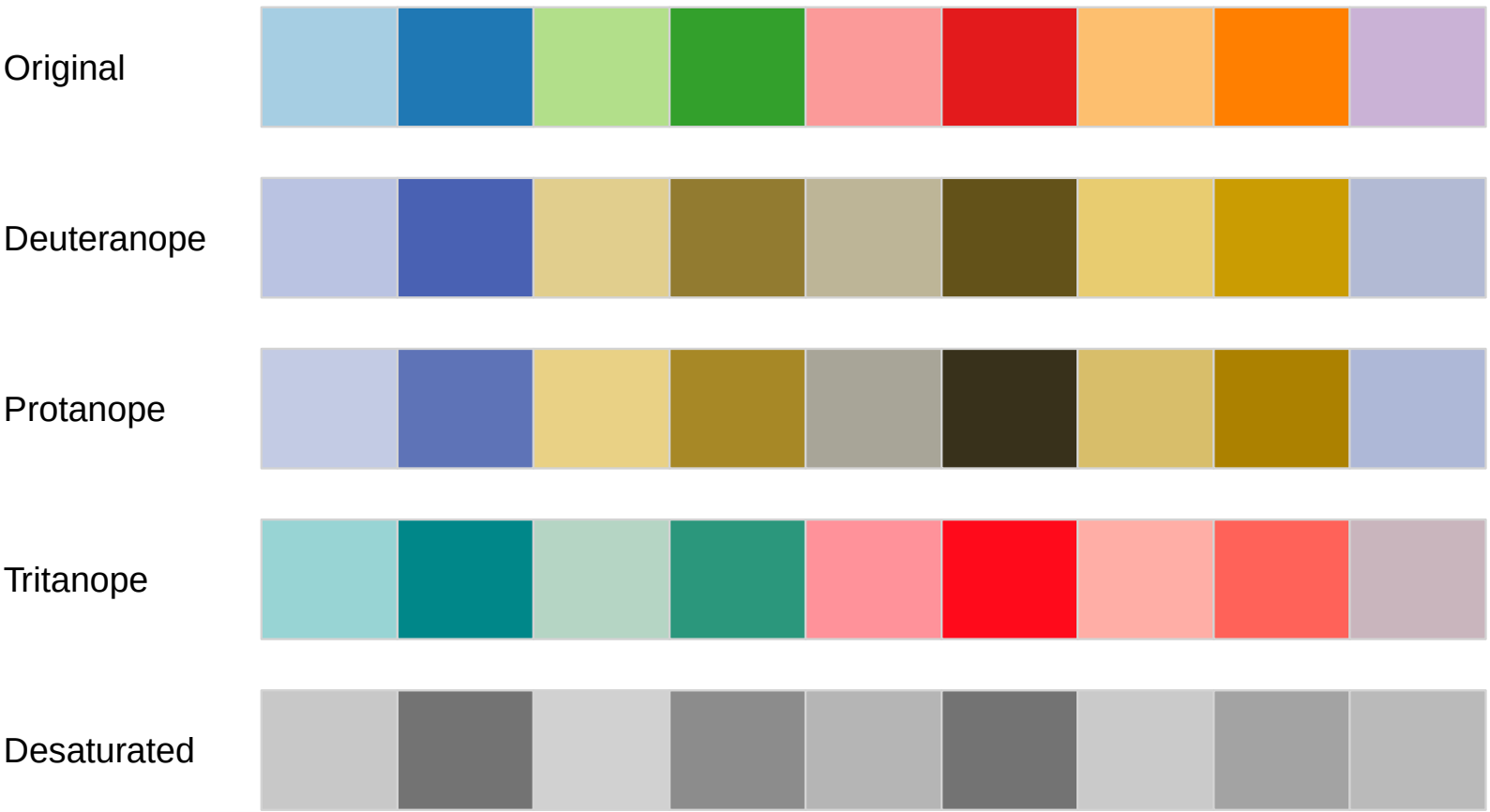
Deuteranopia

Tritanopia

MEMORIAL UNIVERSITY

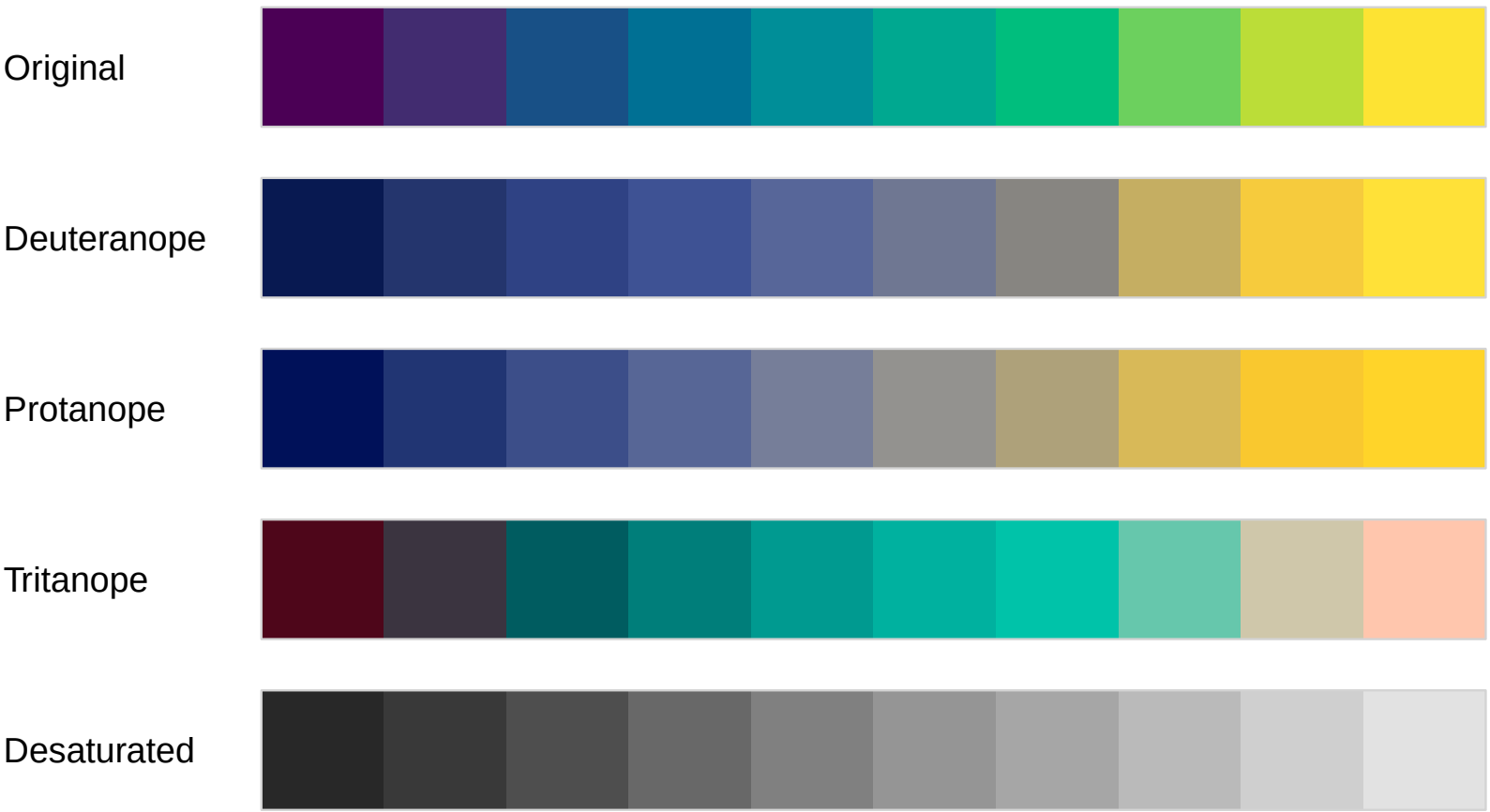
Example: Colourbrewer Palette ”Paired”

Brewer Paired



Viridis Colour Palette

Viridis Palette



Plots for One Variable

Continuous

- **Area**: Degree of change over time, or relationship of parts to aggregate
- **Density, Dot, Frequency, Histogram**: Show frequency distribution of data

Discrete

- **Bar**: Connections among individual things, compare items of different groups
- **Pie**: Relationships of parts to aggregate

MEMORIAL UNIVERSITY

Plots for Two Variables

Both Continuous

- **Point**: Connections among numeric values, show multiple groups of data
- **Lines, Local Regression**: Relationships/correlations among multiple data series or over time
- **Text / Label**: Frequency of labels in content/document

One Discrete, One Continuous

- **Column**: Correlations among things or information changes over time
- **Box, Dot, Violin**: Compare distributions between many groups, display spread and skew of data

MEMORIAL UNIVERSITY

Plots for Two Variables, cont'd

Both Discrete

- **Points/Counts**: Magnitude of counts
- **Jitter**: Plots of data points

Distributions

- **Bin2D, Density2D, Hex**: Shows frequency of values over two continuous variables

Plots for Three Variables

Continuous

- **Contour, Raster and Tile**: Shows relationships among three data series

Visualizing Errors and Uncertainty

Purpose

- Give a general idea of how precise a value is, or how far a value might be from the true value
- Used to augment a given visualization

Common Visualization Styles

- Crossbar
- Errorbar
- Range (line, point)

MEMORIAL UNIVERSITY

Selected Graphics Libraries and Frameworks

R

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

Python

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

Web & JS

- D3, ChartJS, GoogleCharts

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[1]
City	Numeric[2]
Hwy	Numeric
Comb	Numeric
Range	Numeric[3]

[1] Small, Midsize, Large, Pickup, SUV, Station Wagon, etc.
[2] Fuel consumption in l/100km equivalent
[3] Range in km

MEMORIAL UNIVERSITY

Read Data

```
library(tidyverse)

e <- read.csv('fuel.csv')

e$Year <- as.numeric(e$Year)
e$Category <- as.factor(e$Category)
e$Fuel <- as.factor(e$Fuel)
e$City <- as.numeric(e$City)
e$Hwy <- as.numeric(e$Hwy)
e$Comb <- as.numeric(e$Comb)
e$Range <- as.numeric(e$Range)
e$Annual <- as.numeric(e$Annual)

e.clean <- e
```

MEMORIAL UNIVERSITY

Load Graphics Libraries

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

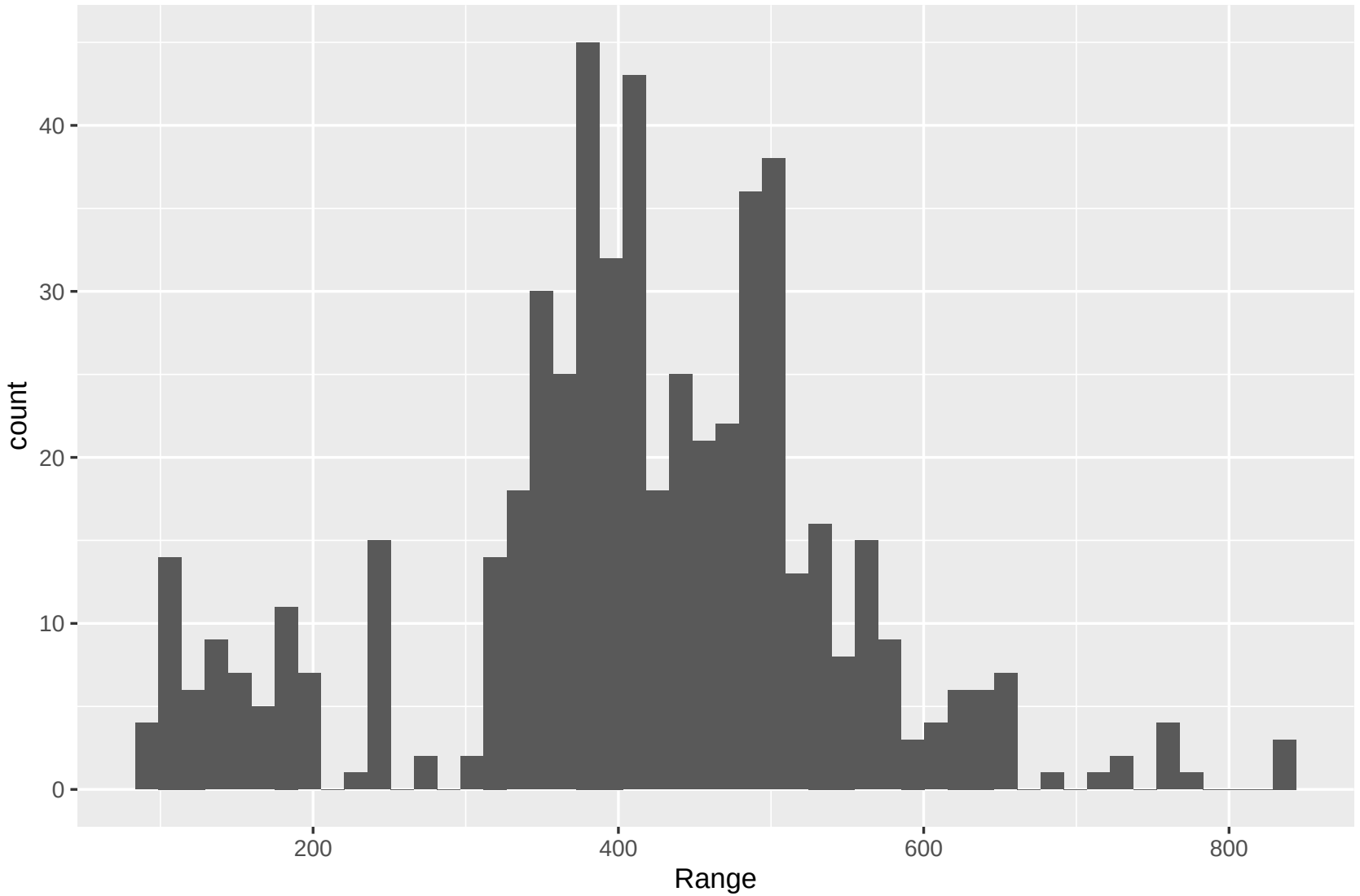
Histogram

```
e.clean |>
  ggplot(aes(x=Range)) +
    geom_histogram(bins=50)

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

- Aesthetic `aes()` determines mapping of data to plot elements
- Add plot functions as needed with their own additional aesthetics and options
- Save plot in different formats (PDF, PNG, JPEG, . . .)

MEMORIAL UNIVERSITY

Histogram



MEMORIAL UNIVERSITY

Density Plot

Prepare some summary statistics:

```
mean_v <- e.clean |>
   summarize(mean_v = mean(Range),
             median_v = median(Range),
             lower95=quantile(Range, .025),
             upper95=quantile(Range, .975),
             maxdensity = max(density(Range)$y))
```

Density Plot [cont'd]

```
e.clean |>
  ggplot(aes(Range)) +
    geom_density(kernel='gaussian',
                 fill='lightblue') +
    labs(x = 'Range (km)',
         y = 'Proportion of Vehicles',
         title='Density Plot - Electric Vehicle Range',
         subtitle='Years 2012 to 2024',
         caption='Lower and Upper 95 percentile, \
                  median and mean') +
    geom_vline(data=mean_v,
               aes(xintercept=mean_v),
               linetype='dashed') +
    geom_vline(data=mean_v,
               aes(xintercept=median_v),
               linetype='dotdash') +
    geom_vline(data=mean_v,
               aes(xintercept=lower95),
               linetype='dotted') +
```

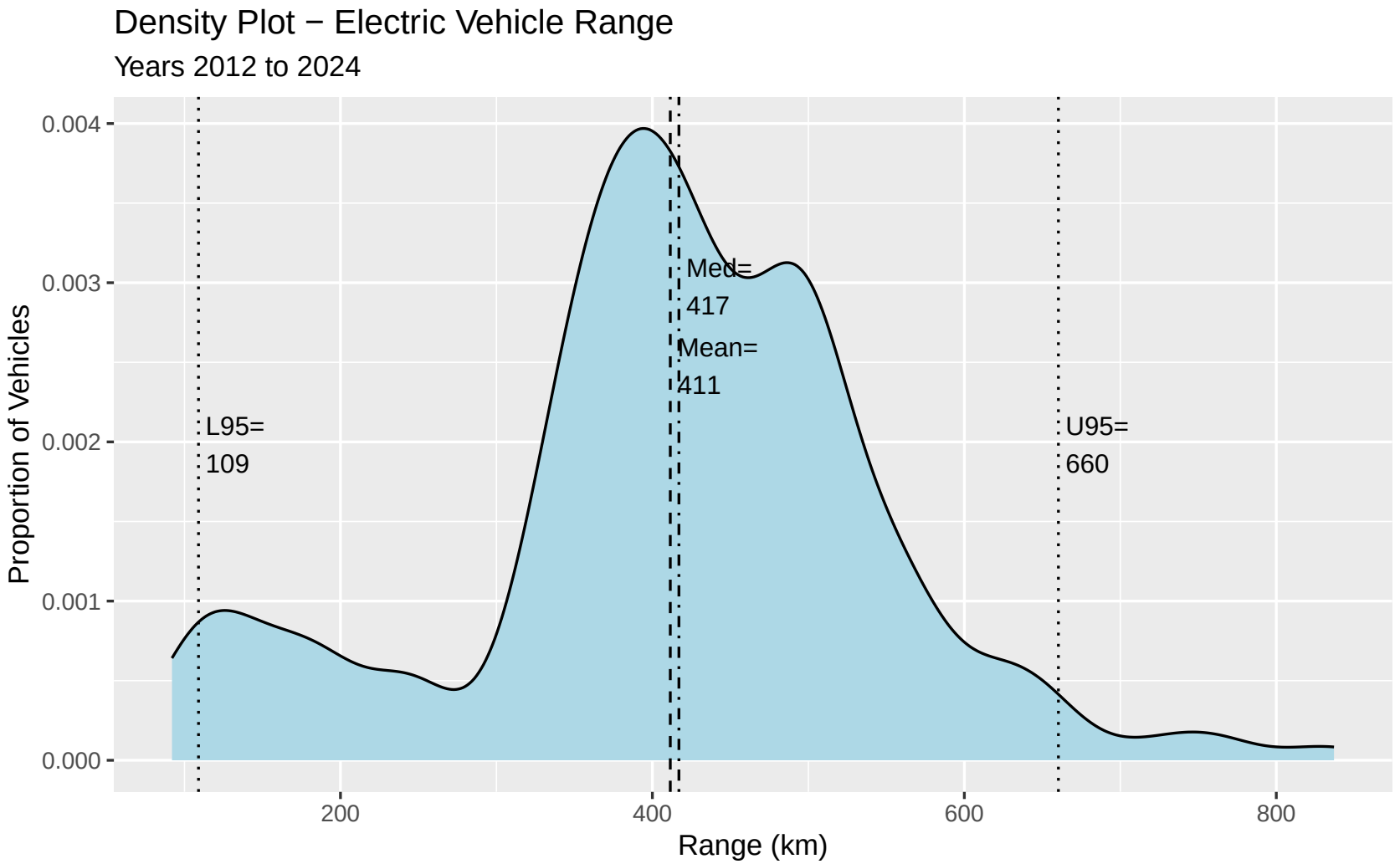
MEMORIAL UNIVERSITY

Density Plot [cont'd]

```
geom_vline(data=mean_v,
           aes(xintercept=upper95),
           linetype='dotted') +
annotate('text',
   label=paste(' L95=\n ',round(mean_v$lower95),sep=''),
   x = mean_v$lower95, y = mean_v$maxdensity/2,
   size=3.5, hjust=0) +
annotate('text',
   label=paste(' Med=\n ',round(mean_v$median_v),sep=''),
   x = mean_v$median_v, y = mean_v$maxdensity*3/4,
   size=3.5, hjust=0) +
annotate('text',
   label=paste(' Mean=\n ',round(mean_v$mean_v),sep=''),
   x = mean_v$mean_v, y = mean_v$maxdensity*5/8,
   size=3.5, hjust=0) +
annotate('text',
   label=paste(' U95=\n ',round(mean_v$upper95),sep=''),
   x = mean_v$upper95, y = mean_v$maxdensity/2,
   size=3.5, hjust=0)
```

MEMORIAL UNIVERSITY

Density Plot [cont'd]



Lower and Upper 95 percentile, median and mean

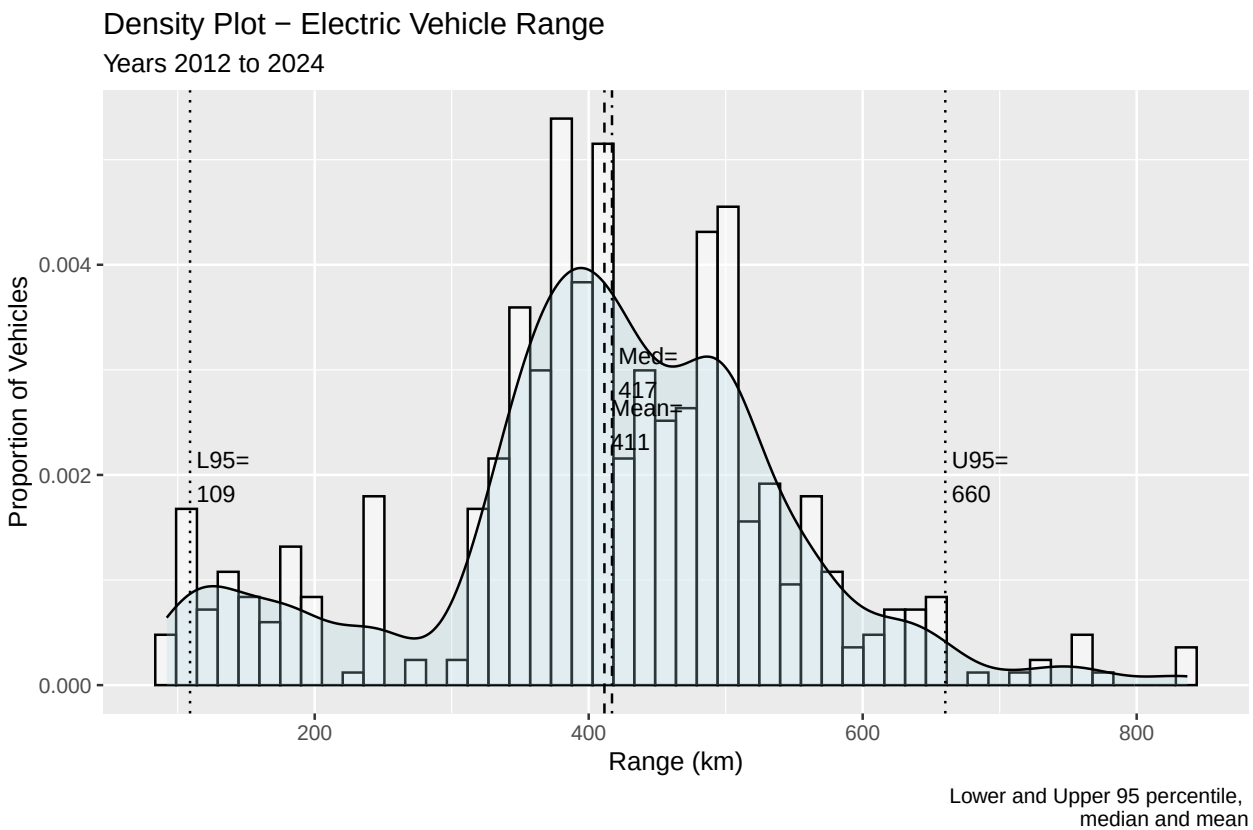
Density Plot [cont'd]

- Add additional elements to a plot with "+"
- Explicitly label plot and axes
- `geom_vline` and `annotate` are elements like `geom_density` and `geom_history` that can be added to plots
- `geom_vline` does not get its data from the pipe, but from the `mean_v` data frame
- Annotations can be freely placed in the plot in coordinate system determined by plot axes

MEMORIAL UNIVERSITY

Histogram

```
...
geom_histogram(aes(y=..density..), bins=50,
     alpha=0.5, fill='white', color='black', ) +
geom_density(kernel='gaussian',
     alpha=0.25, fill='lightblue') +
...
```

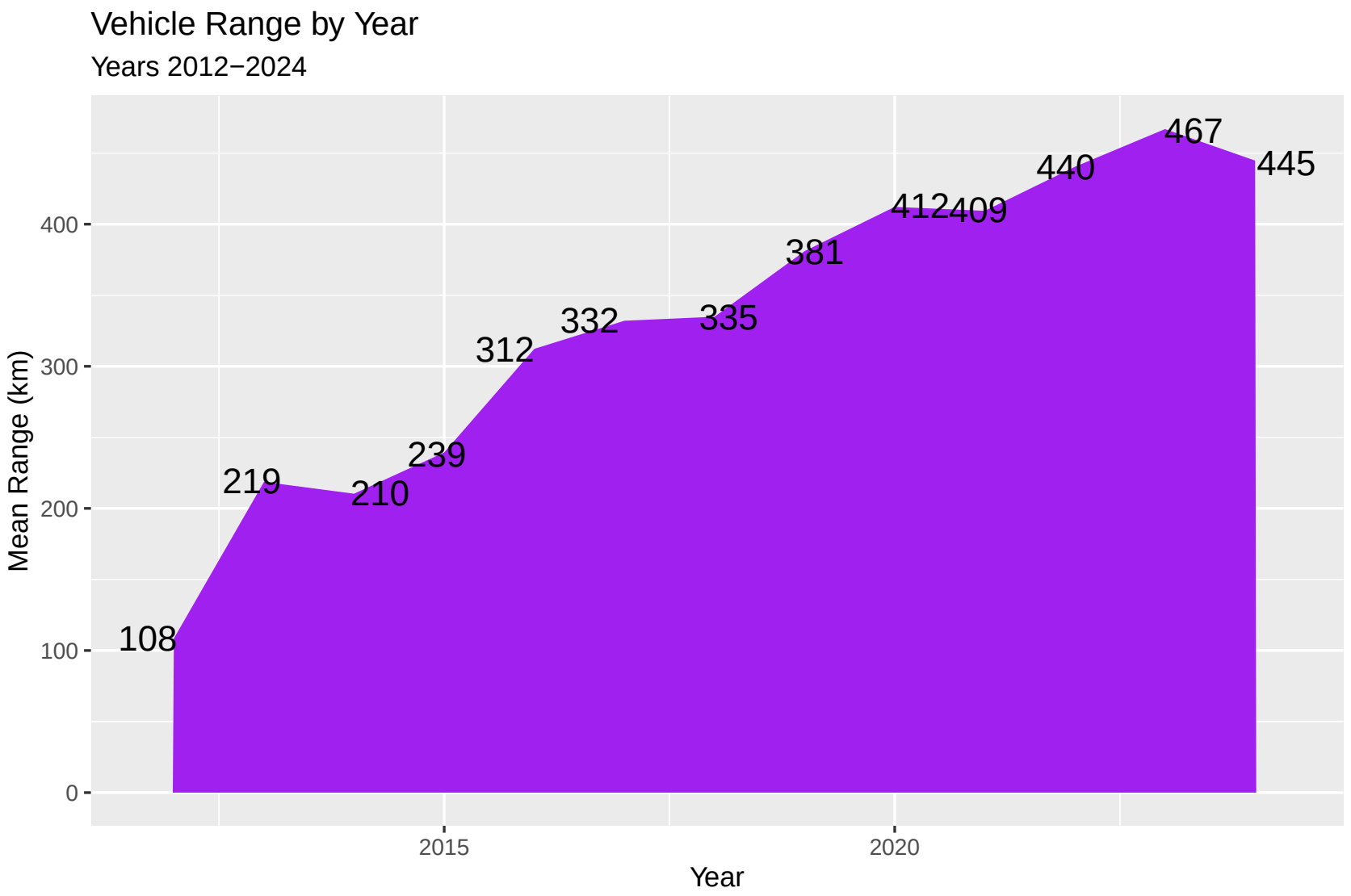


Area Plot

```
e.clean %>%
  group_by(Year) %>%
  summarize(meanRange = mean(Range)) %>%
  ungroup() %>%
  ggplot(aes(Year, meanRange)) +
    geom_area(fill='purple') +
    geom_text(aes(label=round(meanRange)),
              size=5, position='jitter') +
    labs(x='Year', y='Mean Range (km)',
         title='Vehicle Range by Year',
         subtitle='Years 2012-2024')
```

MEMORIAL UNIVERSITY

Area Plot

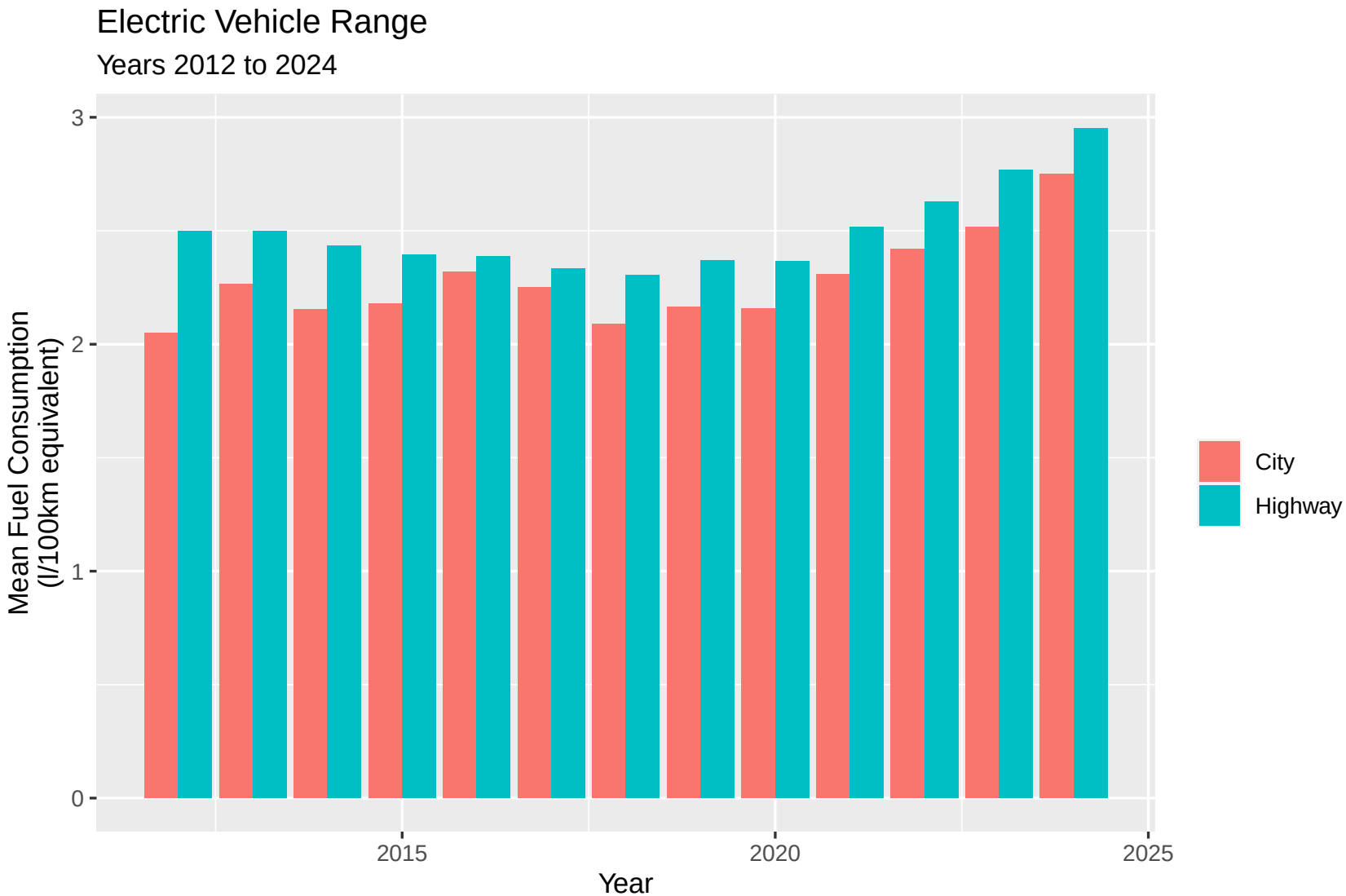

Vehicle Range by Year
Years 2012−2024
Mean Range (km)
0
100
200
300
400
108
219
210
239
312
332
335
381
412
409
440
467
445
2015
2020
Year

Column Chart

```
e.clean %>%
   group_by(Year) %>%
   summarize(meanCity = mean(City), meanHwy = mean(Hwy)) %>%
   ungroup() %>%
   pivot_longer(cols=c('meanCity', 'meanHwy'),
                names_to='metric',
                values_to='consumption') |>
   ggplot(aes(Year, consumption, fill=metric)) +
      geom_col(position='dodge') +
      scale_fill_brewer(palette="Paired") +
      scale_fill_discrete(labels=c("City", "Highway")) +
      labs(x = 'Year',
           y='Mean Fuel Consumption\n(l/100km equivalent)',
           fill='',
           title='Electric Vehicle Range',
           subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

Column Chart



Column Chart (with Patterns)

Prepare data

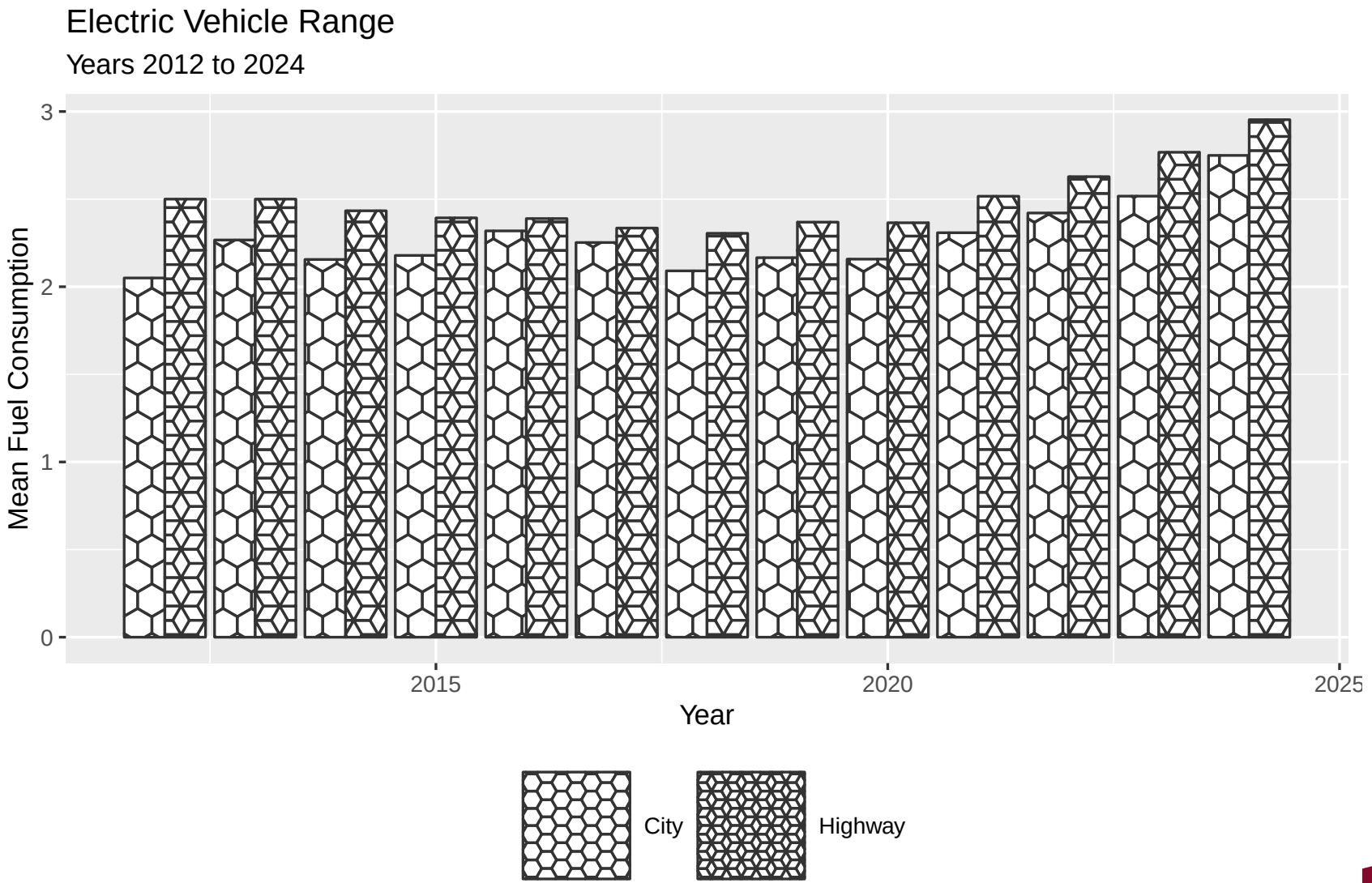
```
e.clean %>%
   group_by(Year) %>%
   summarize(meanCity = mean(City),
             meanHwy = mean(Hwy)) %>%
   ungroup() %>%
   pivot_longer(
        cols=c('meanCity', 'meanHwy'),
        names_to='metric',
        values_to='consumption') %>%
```

Column Chart (with Patterns) [cont'd]

Continued from previous slide ...

```
ggplot(aes(Year, consumption)) +
  geom_col_pattern(
         aes(pattern_type=metric, pattern_angle=metric),
     pattern='polygon_tiling',
     pattern_fill='white',
     pattern_scale=0.5,
     position='dodge',
     pattern_key_scale_factor=0.4) +
  scale_pattern_type_manual(
     values = c('hexagonal', 'rhombille', 'pythagorean',
                'truncated_square', 'rhombitrihexagonal',
                'truncated_trihexagonal'),
     labels=c("City", "Highway")) +
  labs(x = 'Year', y='Mean Fuel Consumption',
       pattern_type='',
       title='Electric Vehicle Range',
       subtitle='Years 2012 to 2024') +
  guides(pattern_angle=FALSE,
         pattern_type=guide_legend(nrow=1)) +
  theme(legend.key.size=unit(1.5, 'cm'),
                legend.position='bottom')
```

Column Chart (with Patterns)



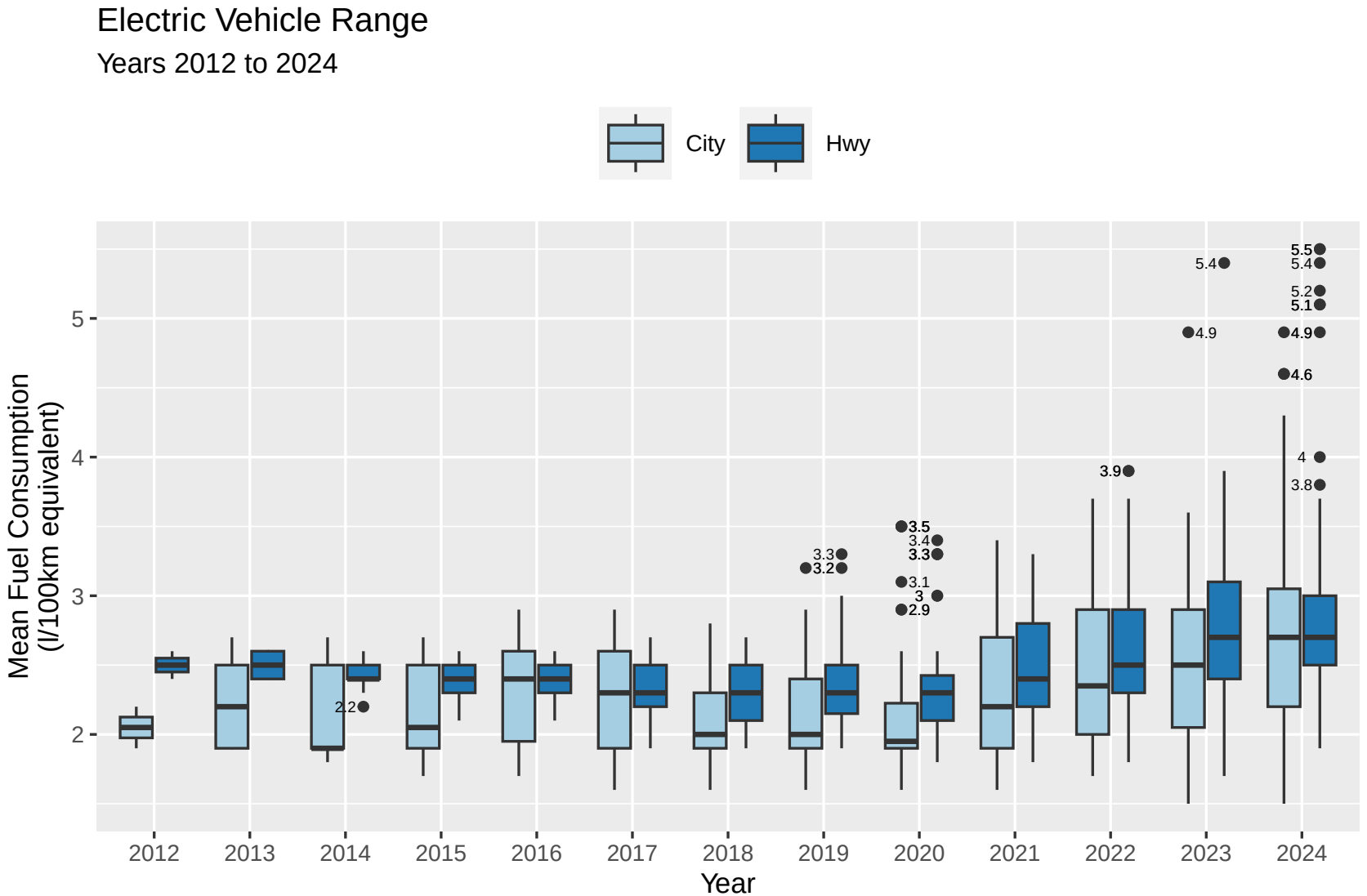
Box Plot

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

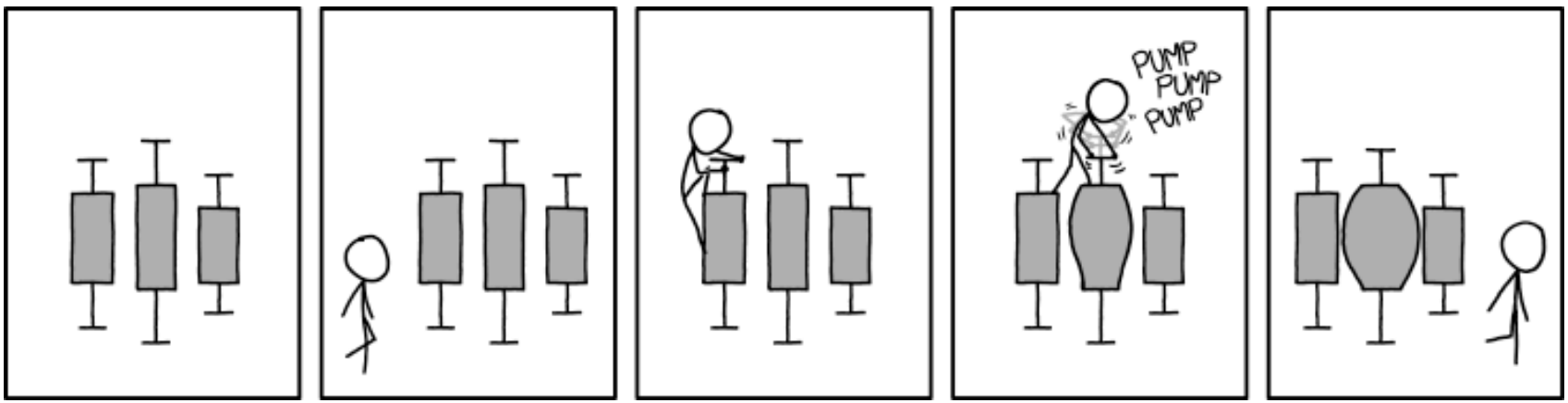
Box Plot

```
  stat_summary(
    aes(label = round(stat(y), 1)),
    geom = "text",
    size=2,
    fun.y = function(y) {
                  o<-boxplot.stats(y)$out;
              if(length(o)==0) NA else o}) +
  scale_fill_brewer(palette="Paired") +
  labs(x = 'Year',
    y='Mean Fuel Consumption\n(l/100km equivalent)',
    fill='',
    title='Electric Vehicle Range',
    subtitle='Years 2012 to 2024') +
  theme(legend.key.size=unit(1, 'cm'),
        legend.position='top')
```

Box Plot


Electric Vehicle Range
Years 2012 to 2024
City
Hwy
Mean Fuel Consumption (l/100km equivalent)
Year
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024

Boxplot (XKCD)

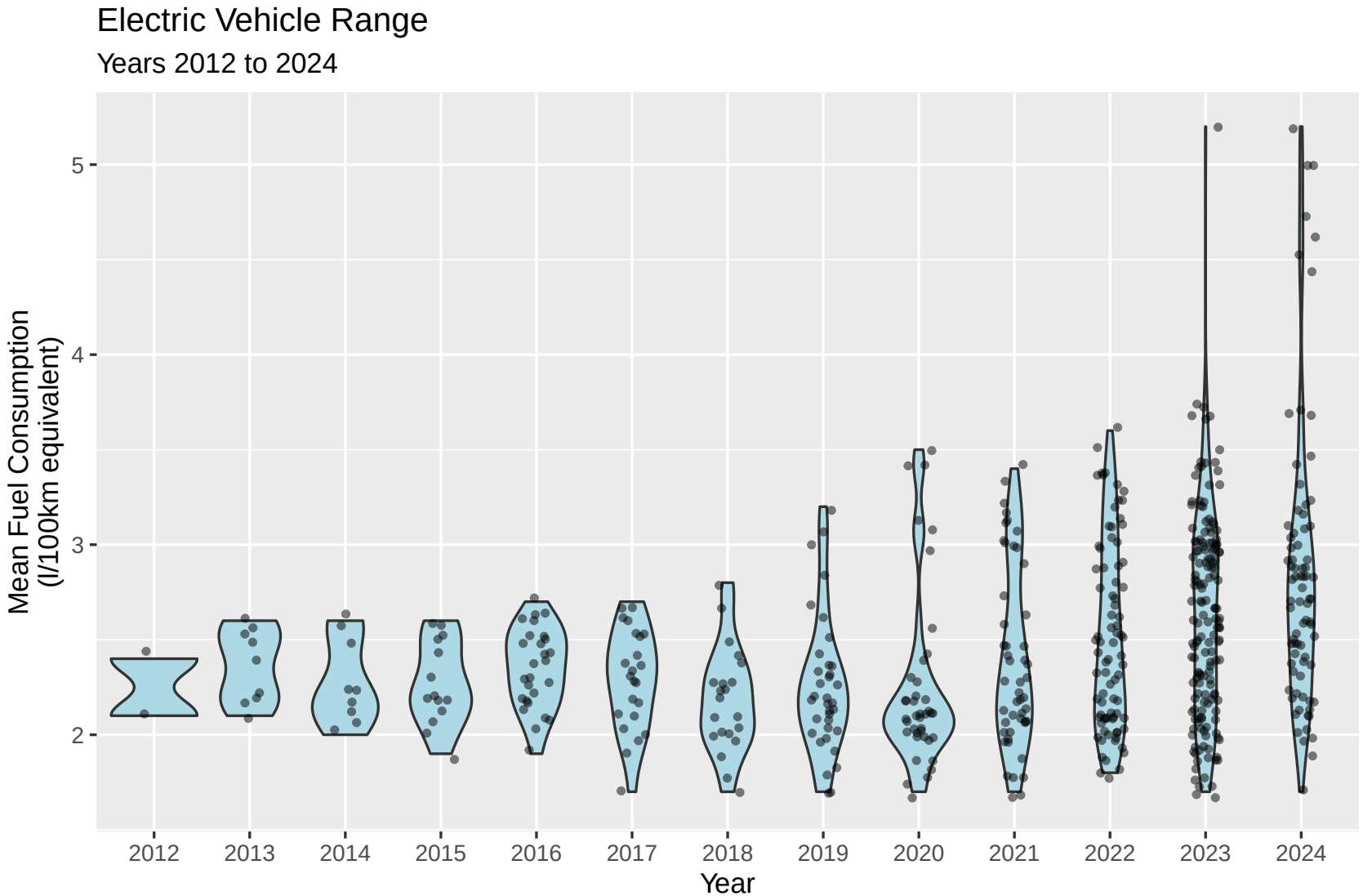


Violin Plot

```
e.clean %>%
  ggplot(aes(x=as.factor(Year), y=Comb)) +
    geom_violin(fill='lightblue') +
    geom_jitter(width=0.15, color='black',
                size=1, fill=NA, alpha=0.5) +
    scale_fill_brewer(palette="Paired") +
    labs(x = 'Year',
         y='Mean Fuel Consumption\n(l/100km equivalent)',
         fill='',
         title='Electric Vehicle Range',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

Violin Plot



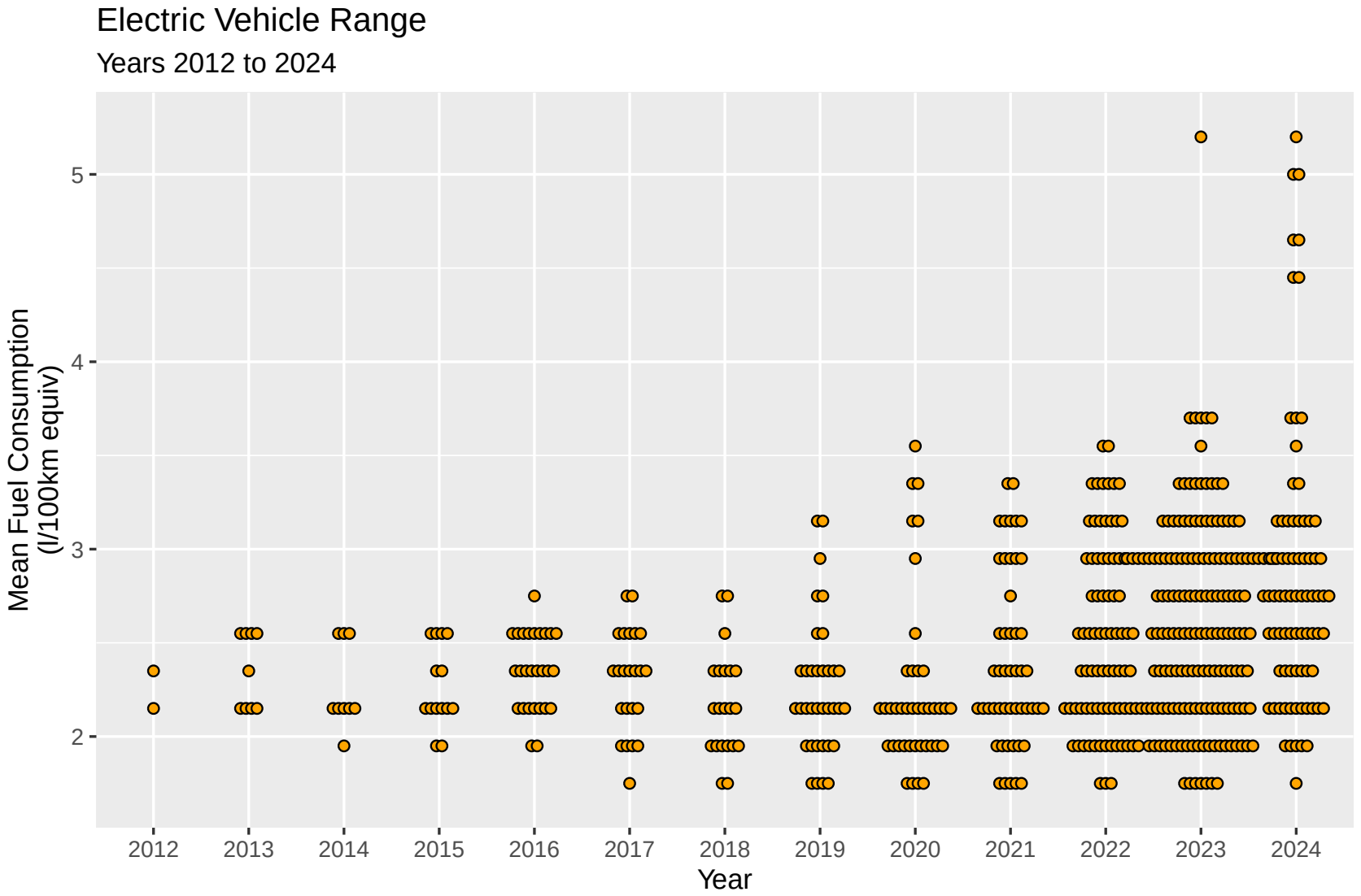
MEMORIAL UNIVERSITY

Dot Plot

```
e.clean %>%
  ggplot(aes(x=as.factor(Year), y=Comb)) +
    geom_dotplot(binaxis='y',
                 stackdir='center',
                 stackratio=0.5,
                 binpositions='all',
                 dotsize=0.5,
                 color='black',
                 fill='orange') +
    scale_fill_brewer(palette="Paired") +
    labs(x = 'Year',
         y='Mean Fuel Consumption\n(l/100km equiv)',
         fill='',
         title='Electric Vehicle Range',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

Dot Plot



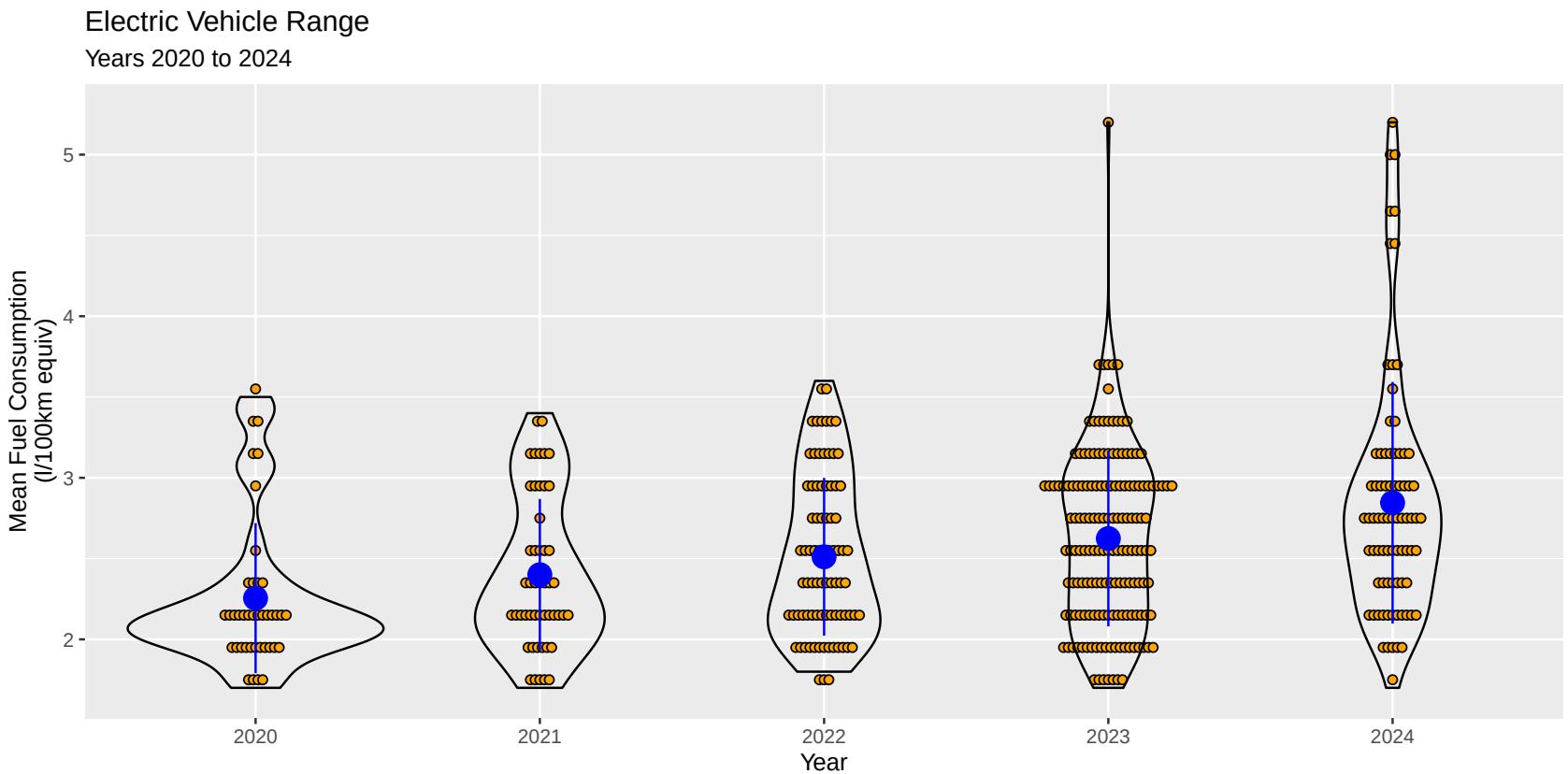
MEMORIAL UNIVERSITY

Dot Plot (with Violin and Range Summary)

```
e.clean %>%
  filter(Year > 2019) %>%
  ggplot(aes(x=as.factor(Year), y=Comb)) +
    geom_dotplot(binaxis='y',
                 stackdir='center', stackratio=0.5,
                 binpositions='all', dotsize=0.5,
                 color='black', fill='orange') +
    geom_violin(color='black', fill=NA) +
    stat_summary(fun.data=mean_sdl,
                 fun.args=list(mult=1),
                 size=1, color='blue',
                 geom="pointrange") +
    scale_fill_brewer(palette="Paired") +
    labs(x = 'Year',
         y='Mean Fuel Consumption\n(l/100km equiv)',
         fill='',
         title='Electric Vehicle Range',
         subtitle='Years 2020 to 2024') +
     theme(legend.position='none')
```

MEMORIAL UNIVERSITY

Dot Plot (with Violin and Range Summary)



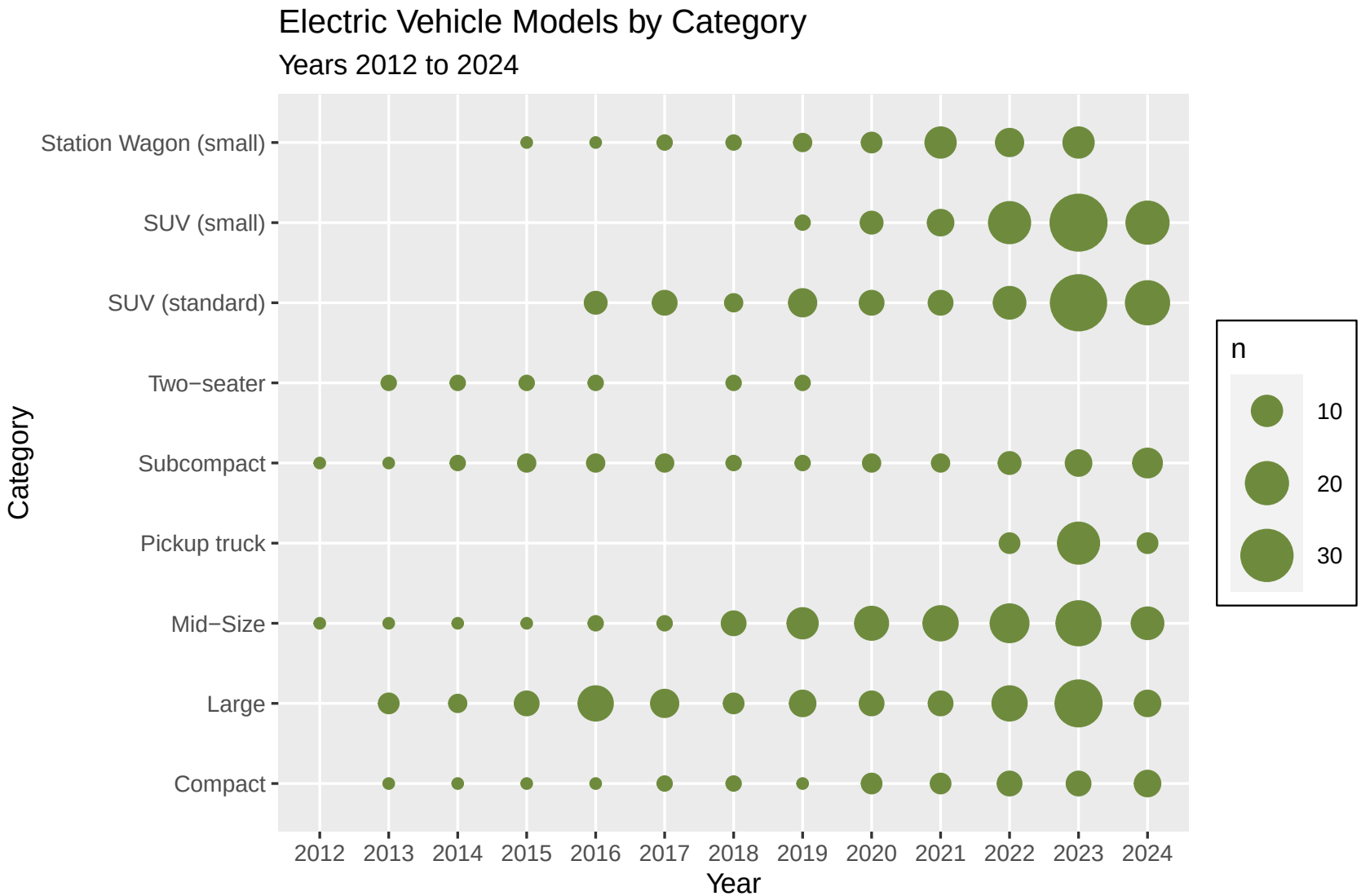
MEMORIAL UNIVERSITY

Count Plot

```
e.clean %>%
  ggplot(aes(as.factor(Year), as.factor(Category))) +
    geom_count(color='darkolivegreen4')+
    scale_size_area(max_size=10, n.breaks=6) +
    scale_color_brewer(palette="Paired") +
    scale_y_discrete(
      labels=c('Compact', 'Large', 'Mid-Size', 'Pickup truck',
               'Subcompact', 'Two-seater', 'SUV (standard)',
               'SUV (small)', 'Station Wagon (small)')) +
    guides(color=FALSE) +
    labs(x = 'Year',
         y='Category',
         fill='',
         title='Electric Vehicle Models by Category',
         subtitle='Years 2012 to 2024') +
    theme(legend.background=element_blank(),
          legend.box.background=element_rect(color='black',
                                             fill=NA),
          legend.key.size=unit(1, 'cm'))
```

MEMORIAL UNIVERSITY

Count Plot

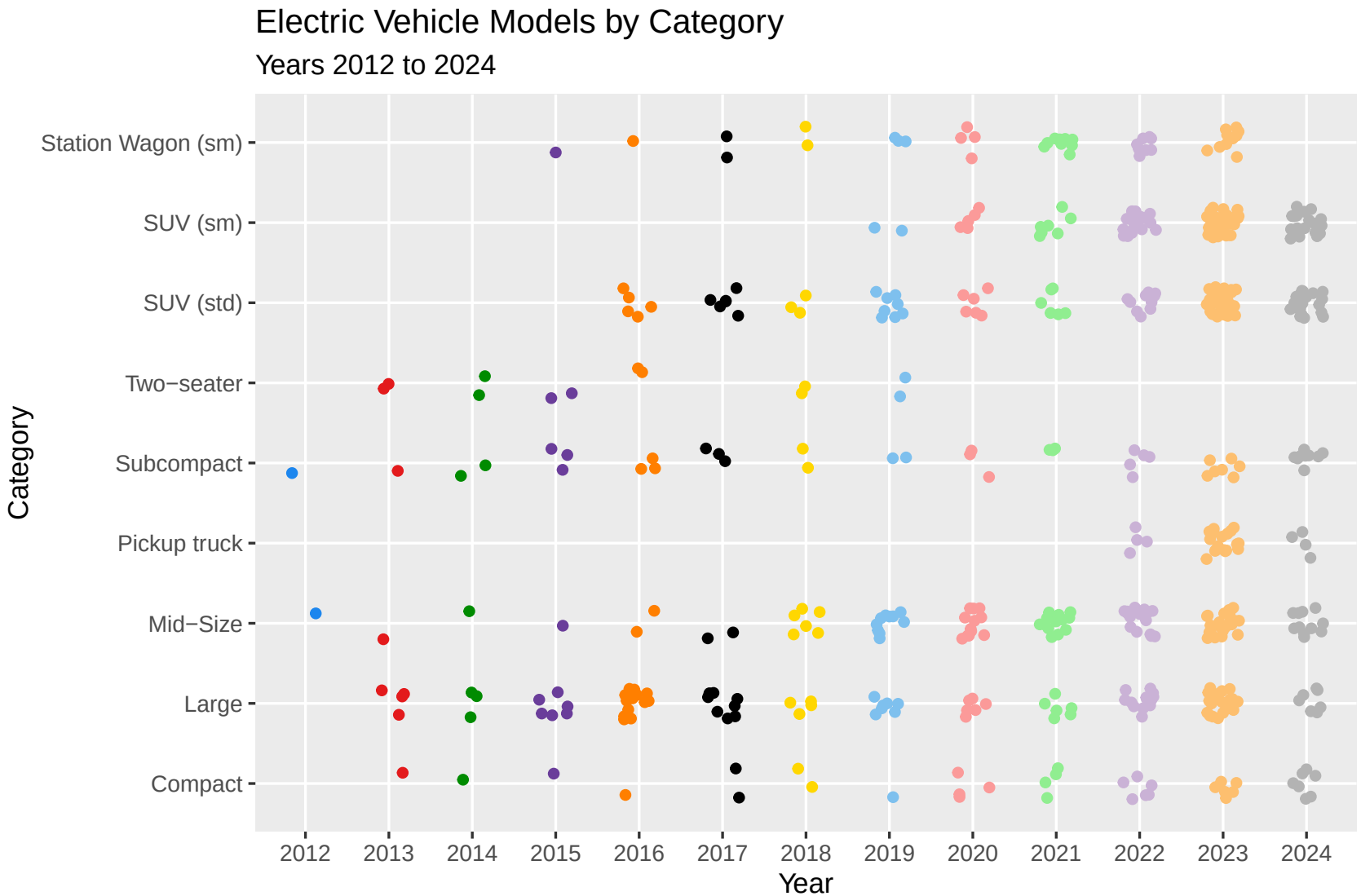

Electric Vehicle Models by Category
Years 2012 to 2024
Station Wagon (small)
SUV (small)
SUV (standard)
Two−seater
Subcompact
Pickup truck
Mid−Size
Large
Compact
Category
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Year
n
10
20
30

Jitter Plot

```
e.clean %>%
  ggplot(aes(x=as.factor(Year),
             y=as.factor(Category),
             color=as.factor(Year))) +
    geom_jitter(width=0.2, height=0.2) +
    scale_color_manual(values=c25) +
    scale_y_discrete(
      labels=c('Compact', 'Large', 'Mid-Size',
               'Pickup truck', 'Subcompact',
               'Two-seater', 'SUV (std)',
               'SUV (sm)', 'Station Wagon (sm)')) +
    guides(color=FALSE) +
    labs(x = 'Year',
         y='Category',
         fill='Make',
         title='Electric Vehicle Models by Category',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

Jitter Plot


Electric Vehicle Models by Category
Years 2012 to 2024
Category
Station Wagon (sm)
SUV (sm)
SUV (std)
Two−seater
Subcompact
Pickup truck
Mid−Size
Large
Compact
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
Year

Points Plot

```
e.clean %>%
  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)) +
  scale_color_tron() +
  scale_y_continuous(labels=scales::comma) +
  scale_color_discrete(
     labels=c('Compact', 'Large', 'Mid-Size',
              'Pickup truck', 'Subcompact',
              'Two-seater', 'SUV (dtd)',
              'SUV (sm)', 'Station Wagon (sm)')) +
```

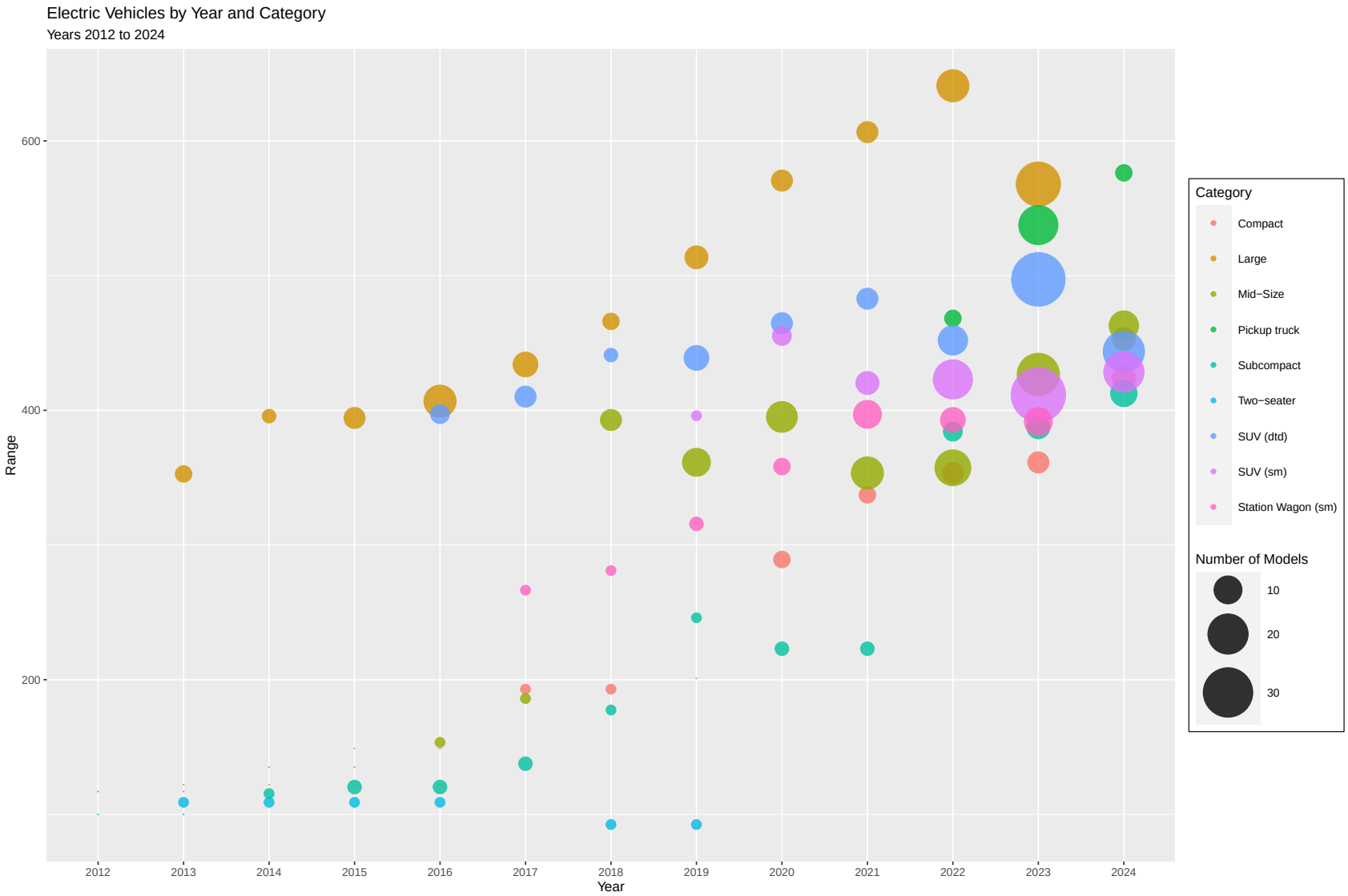
MEMORIAL UNIVERSITY

Points Plot [cont'd]

Continued from previous slide . . .

```
labs(x = 'Year', y='Range',
     fill='Make', size='Number of Models',
     title='Electric Vehicles by Year and Category',
     subtitle='Years 2012 to 2024', ) +
guides(color=guide_legend(position='bottom'),
       size=guide_legend(position='right')) +
theme(legend.background=element_blank(),
      legend.box.background=element_rect(color='black',
                                         fill=NA),
      legend.key.size=unit(1, 'cm'))
```

Points Plot



MEMORIAL UNIVERSITY

Lines and Points Plot

```
e.clean %>%
  filter(Year >= 2022 & Year <= 2023) %>%
  filter(Comb <= 4) %>%
  filter(Category != 'PL') %>%
  filter(Category != 'T') %>%
ggplot(aes(Comb, Range,
           color=Category,
           shape=Category,
           linetype=Category)) +
  geom_line(size=1) +
  geom_point(size=4) +
```

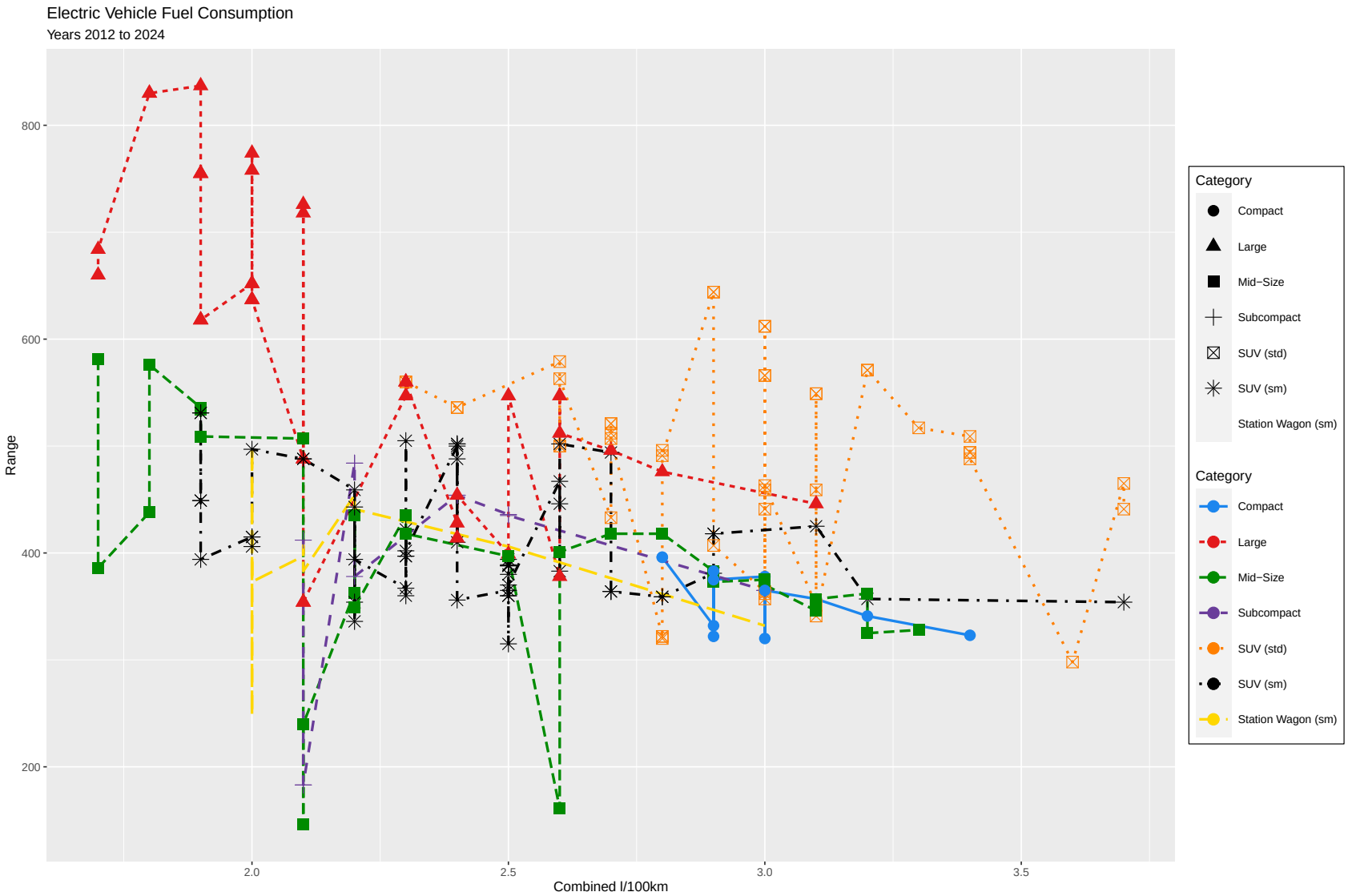
Lines and Points Plot [cont'd]

Continued from previous slide . . .

```
  scale_color_manual(values=c25,
    labels=c('Compact', 'Large', 'Mid-Size',
             'Subcompact', 'SUV (std)',
             'SUV (sn)', 'Station Wagon (sm)')) +
  scale_linetype(
    labels=c('Compact', 'Large', 'Mid-Size',
             'Subcompact', 'SUV (std)',
             'SUV (sm)', 'Station Wagon (sm)')) +
  scale_shape(
    labels=c('Compact', 'Large', 'Mid-Size',
             'Subcompact', 'SUV (std)',
             'SUV (small)', 'Station Wagon (sm)')) +
...
```

MEMORIAL UNIVERSITY

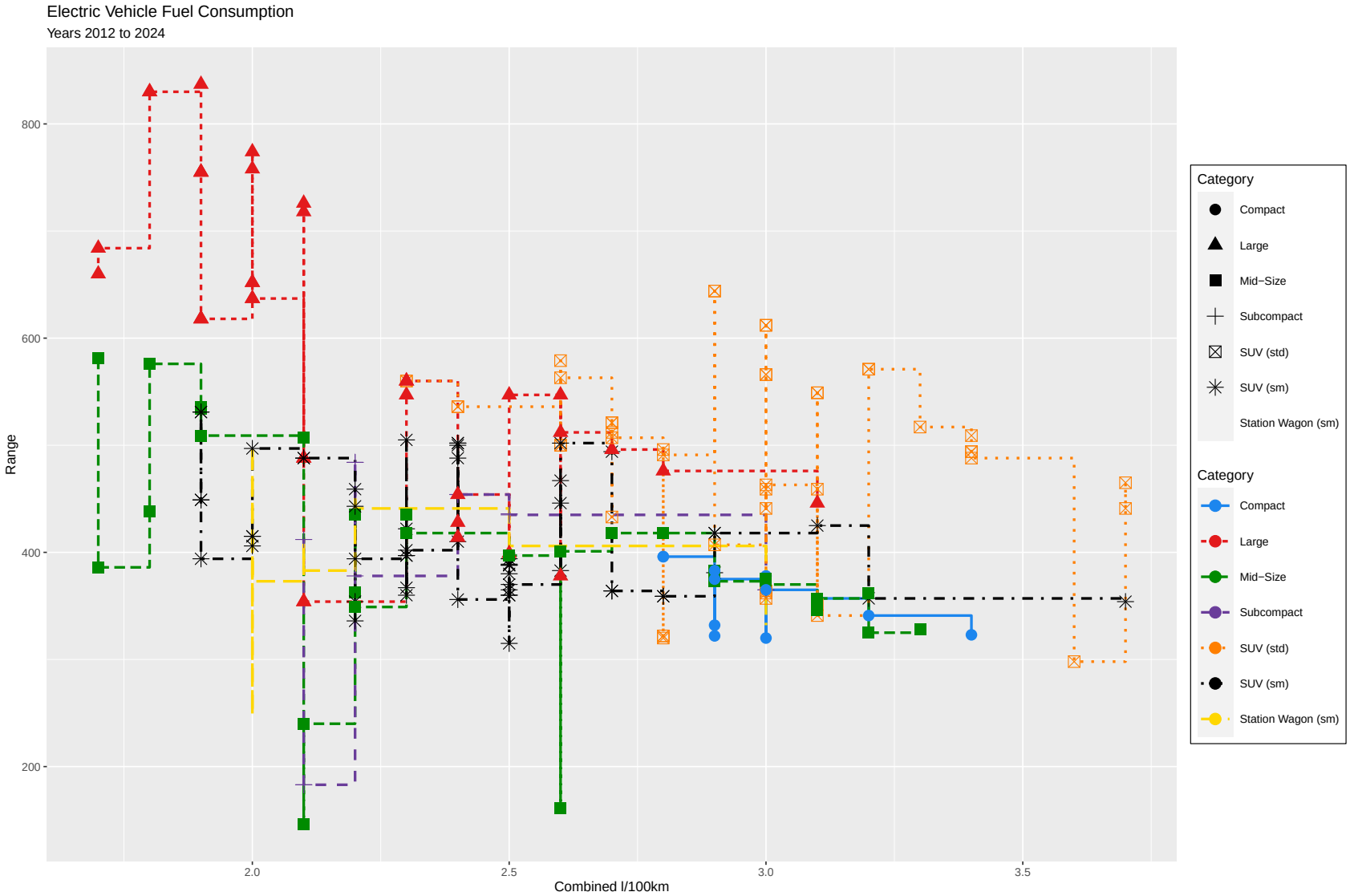
Lines and Points Plot



Stepped Lines Plot

```
...
      geom_step(size=1) +
...
```

Stepped Lines Plot

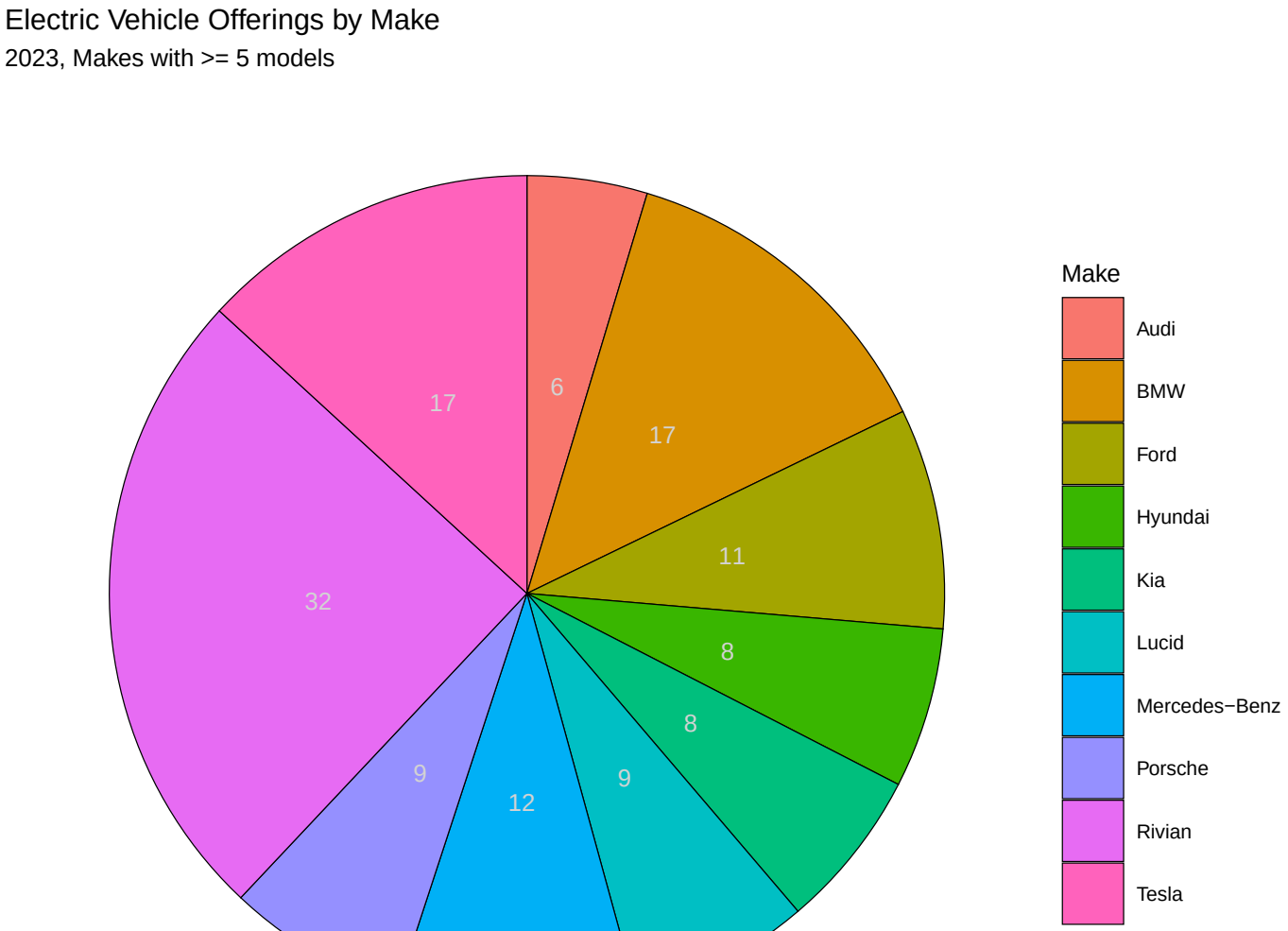


MEMORIAL UNIVERSITY

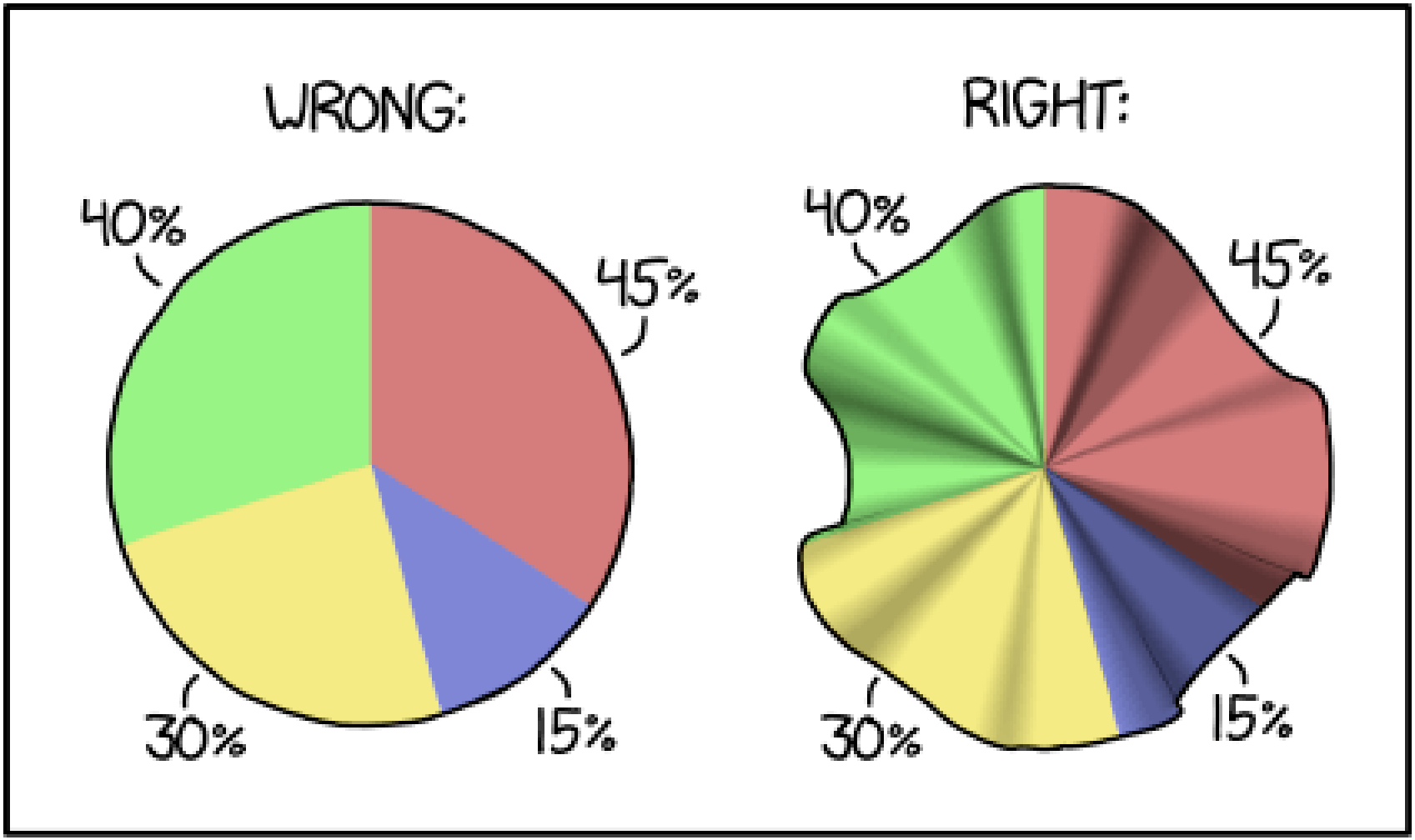
Pie Chart

```
e.clean %>%
  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,'')),
     color='lightgrey',
     position = position_stack(vjust=0.5)) +
  scale_y_continuous(labels=NULL) +
  scale_color_brewer(palette="Paired") +
  labs(x = '', y = '',  fill='Make',
       title='Electric Vehicle Offerings by Make',
       subtitle='2023, Makes with >= 5 models') +
  theme_void() +
  theme(legend.key.size=unit(1, 'cm'))
```

Pie Chart


Electric Vehicle Offerings by Make
2023, Makes with >= 5 models
6
17
11
8
8
9
12
9
32
17
Make
Audi
BMW
Ford
Hyundai
Kia
Lucid
Mercedes−Benz
Porsche
Rivian
Tesla

Pie Charts (XKCD)



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

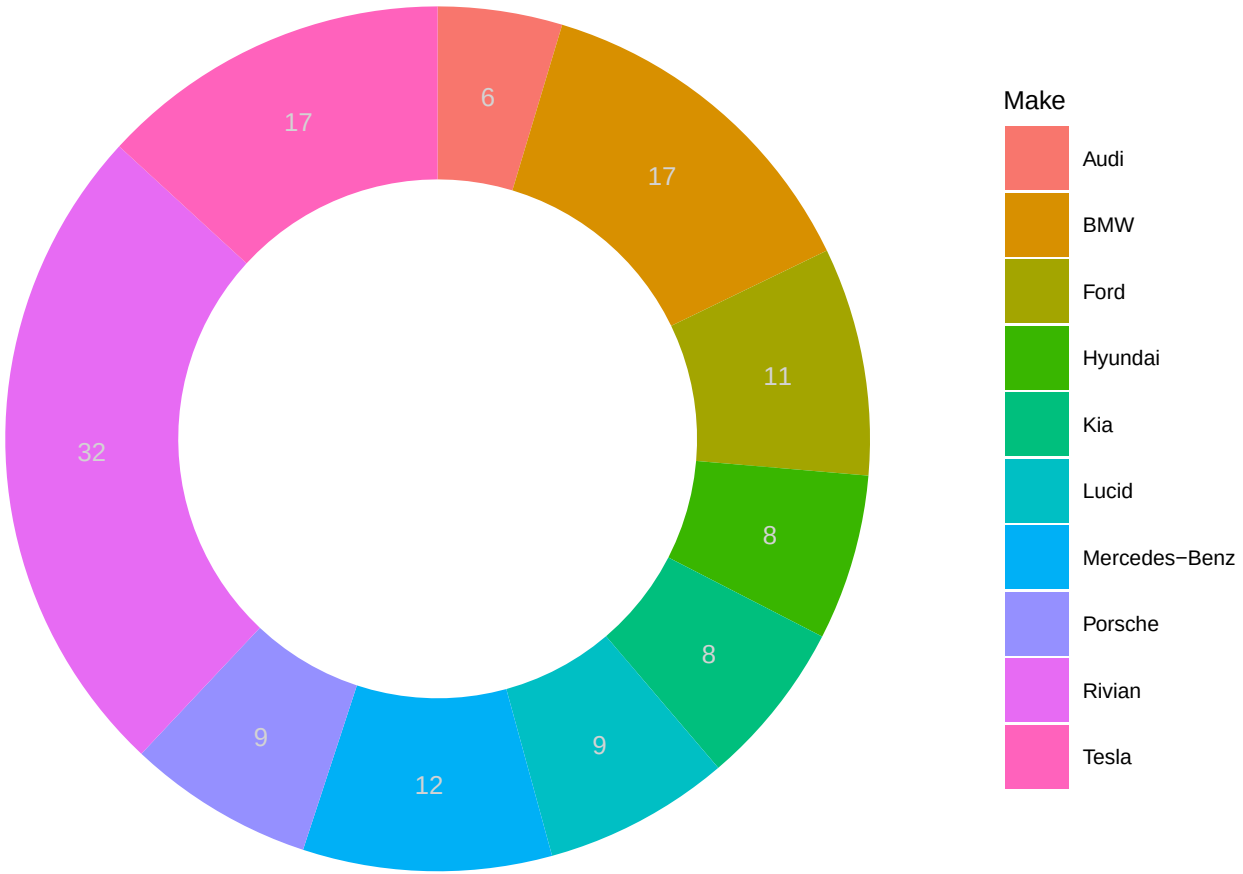
Donut Chart

```
holesize <- 2

....

 ggplot(aes(x=holesize, y=totalcount, fill=Make)) +
   geom_col() +
   xlim(c(0.2, holesize+0.5)) +

...
```

Donut Chart

Electric Vehicle Offerings by Make

2023, Makes with >= 5 models

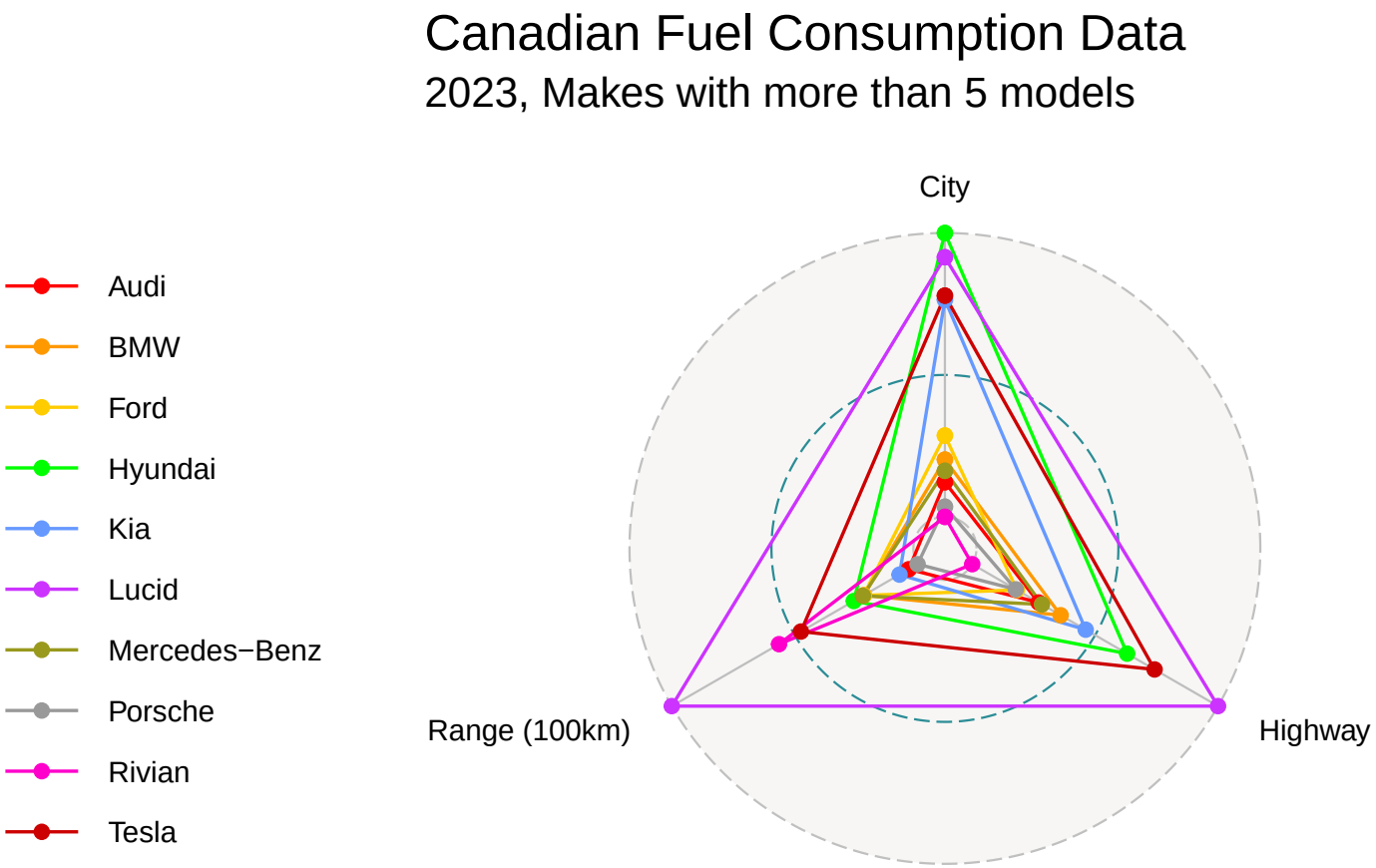


MEMORIAL UNIVERSITY

Radar Plot

```
e.clean %>%
  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_at(vars(-Make), rescale) %>%
  ggradar(axis.labels=
              c('City', 'Highway', 'Range (100km)'),
          values.radar='',
          group.line.width=0.75,
          group.point.size=3) +
     scale_color_ucscgb() +
  labs(x = '', y = '',  fill='Make',
       title='Canadian Fuel Consumption Data',
       subtitle='2023, Makes with more than 5 models')
```

Radar Plot


Canadian Fuel Consumption Data
2023, Makes with more than 5 models
City
Highway
Range (100km)
Audi
BMW
Ford
Hyundai
Kia
Lucid
Mercedes−Benz
Porsche
Rivian
Tesla

Lines with Multiple Axes

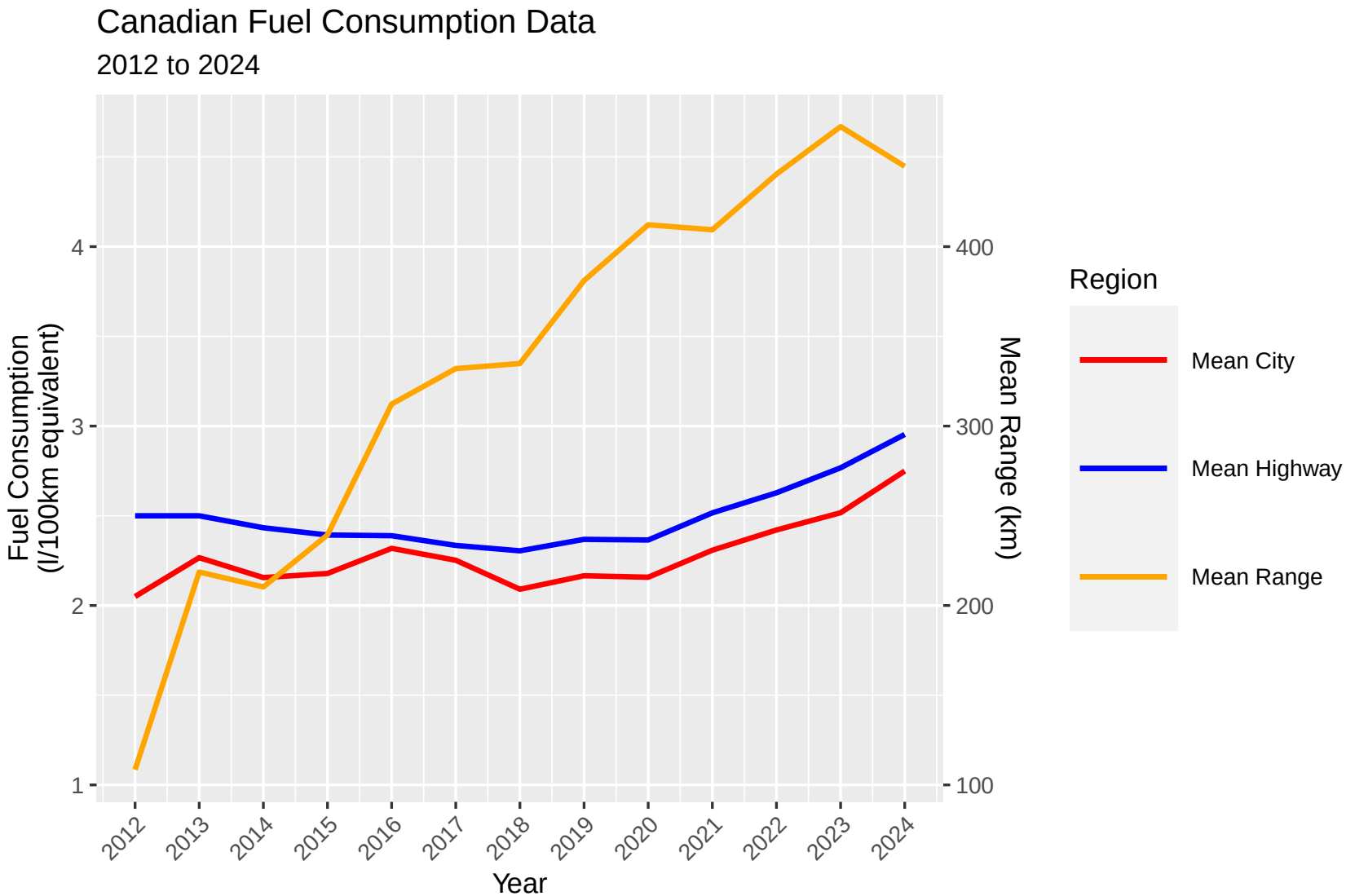
```
e.clean %>%
   group_by(Year) %>%
   summarize(meanCity = mean(City),
             meanHwy = mean(Hwy),
             meanRange = mean(Range)) %>%
   ungroup() %>%
   mutate(meanRange2 = meanRange/100) %>%
```

Lines with Multiple Axes

```
ggplot(aes(x=Year)) +
  scale_color_manual(name='Region',
     values=c('Mean City' = 'red',
              'Mean Highway' = 'blue',
              'Mean Range' = 'orange')) +
  geom_line(aes(y=meanCity, color='Mean City')) +
  geom_line(aes(y=meanHwy, color='Mean Highway')) +
  geom_line(aes(y=meanRange2, color='Mean Range')) +
  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)) +
  labs(x = 'Year', color='',
       title='Canadian Fuel Consumption Data',
       subtitle='2012 to 2024') +
  theme(legend.key.size=unit(1.5, 'cm'),
        axis.text.x = element_text(angle=45, hjust=1))
```

MEMORIAL UNIVERSITY

Lines with Multiple Axes

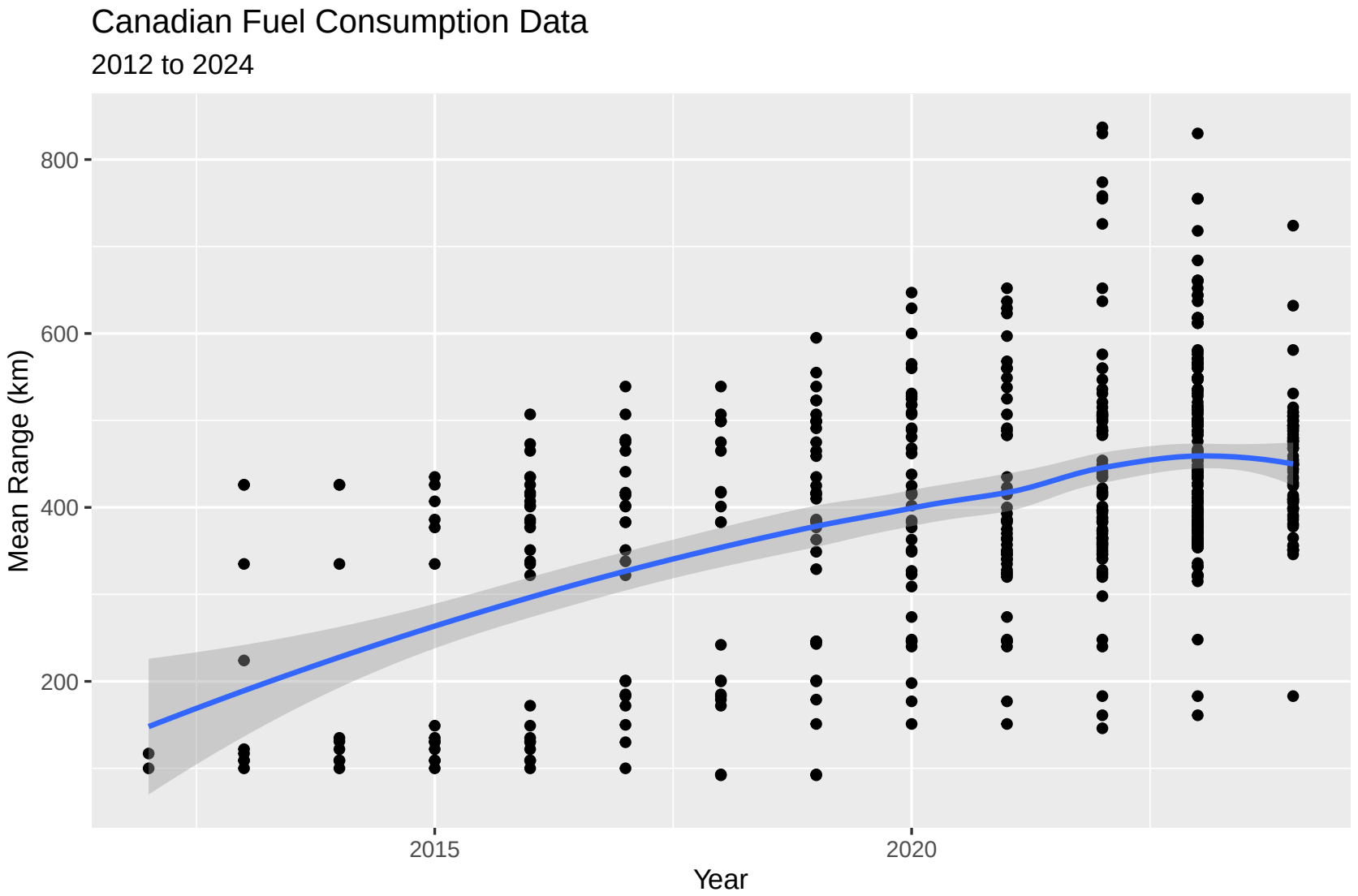


Local Regression Smoothing Plot

```
e.clean %>%
  ggplot(aes(Year, Range)) +
    geom_point() +
    geom_smooth() +
    scale_y_continuous(labels=scales::comma) +
    labs(x = 'Year', color='', y = 'Mean Range (km)',
         title='Canadian Fuel Consumption Data',
         subtitle='2012 to 2024')
```

MEMORIAL UNIVERSITY

Local Regression Smoothing Plot

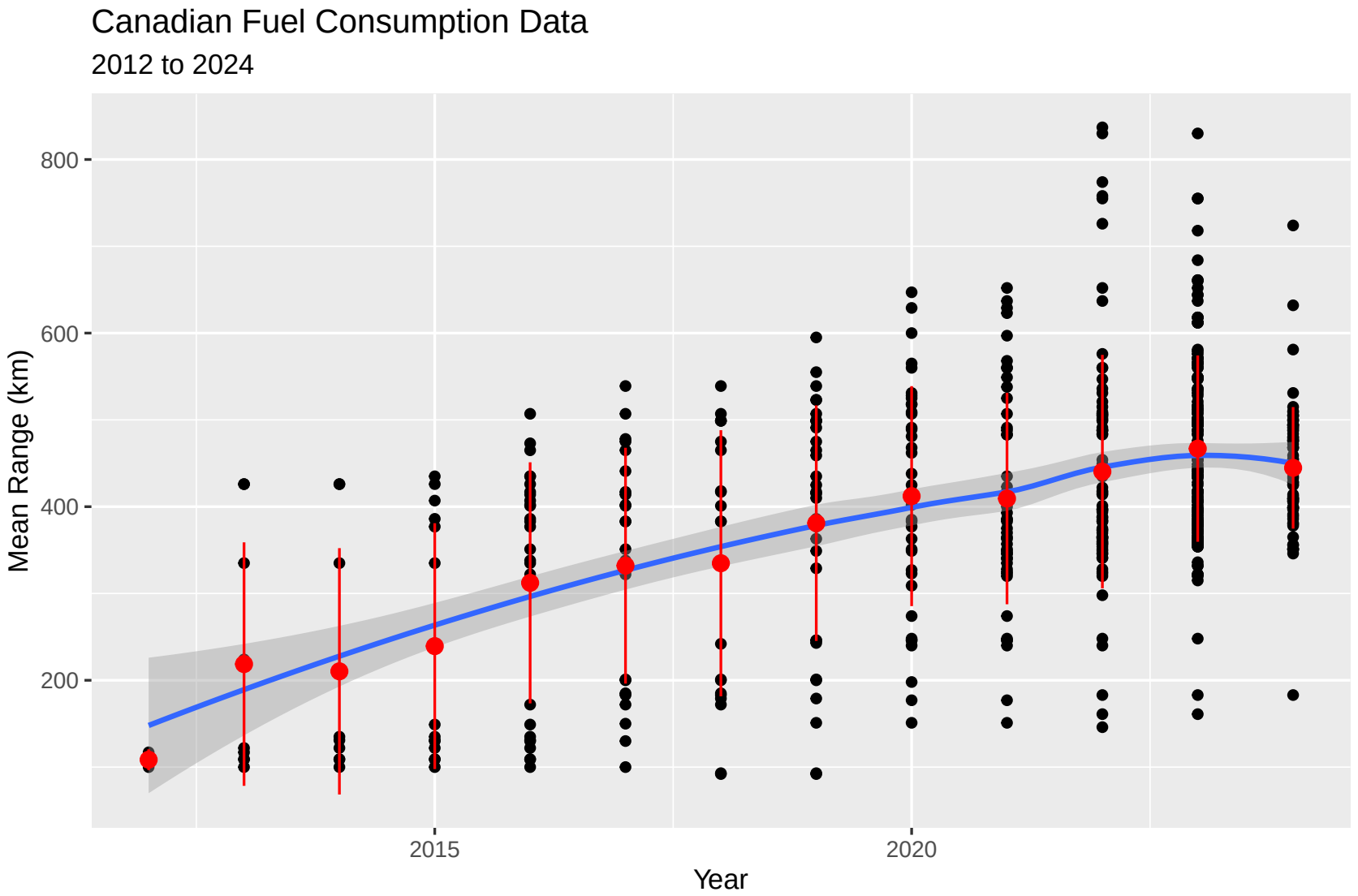


Local Regression Smoothing Plot (with range bar)

```
e.clean %>%
  ggplot(aes(Year, Range)) +
    geom_point() +
    geom_smooth() +
    stat_summary(
       fun.data=mean_sdl,
       fun.args=list(mult=1),
       color='red',
       geom="pointrange") +
    scale_y_continuous(labels=scales::comma) +
    labs(x = 'Year', color='', y = 'Mean Range (km)',
    title='Canadian Fuel Consumption Data',
    subtitle='2012 to 2024')
```

MEMORIAL UNIVERSITY

Local Regression Smoothing Plot (with range bar)

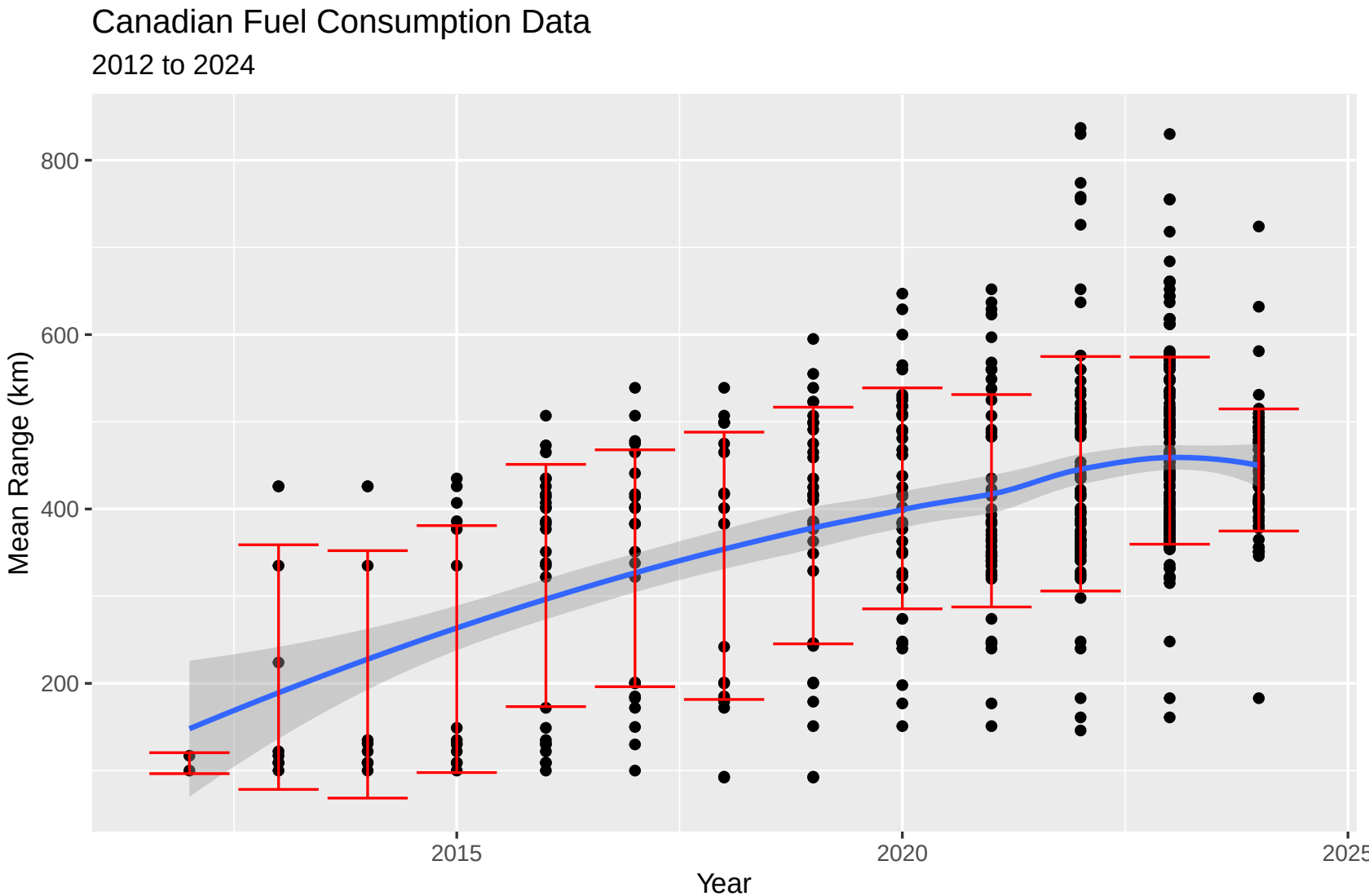


Local Regression Smoothing Plot (with error bars)

```
e.clean %>%
  ggplot(aes(Year, Range)) +
    geom_point() +
    geom_smooth() +
    stat_summary(
        fun.data=mean_sdl,
        fun.args=list(mult=1),
        color='red',
        geom="errorbar") +
    scale_y_continuous(labels=scales::comma) +
    labs(x = 'Year', color='', y = 'Mean Range (km)',
    title='Canadian Fuel Consumption Data',
    subtitle='2012 to 2024')
```

MEMORIAL UNIVERSITY

Local Regression Smoothing Plot (with error bars)



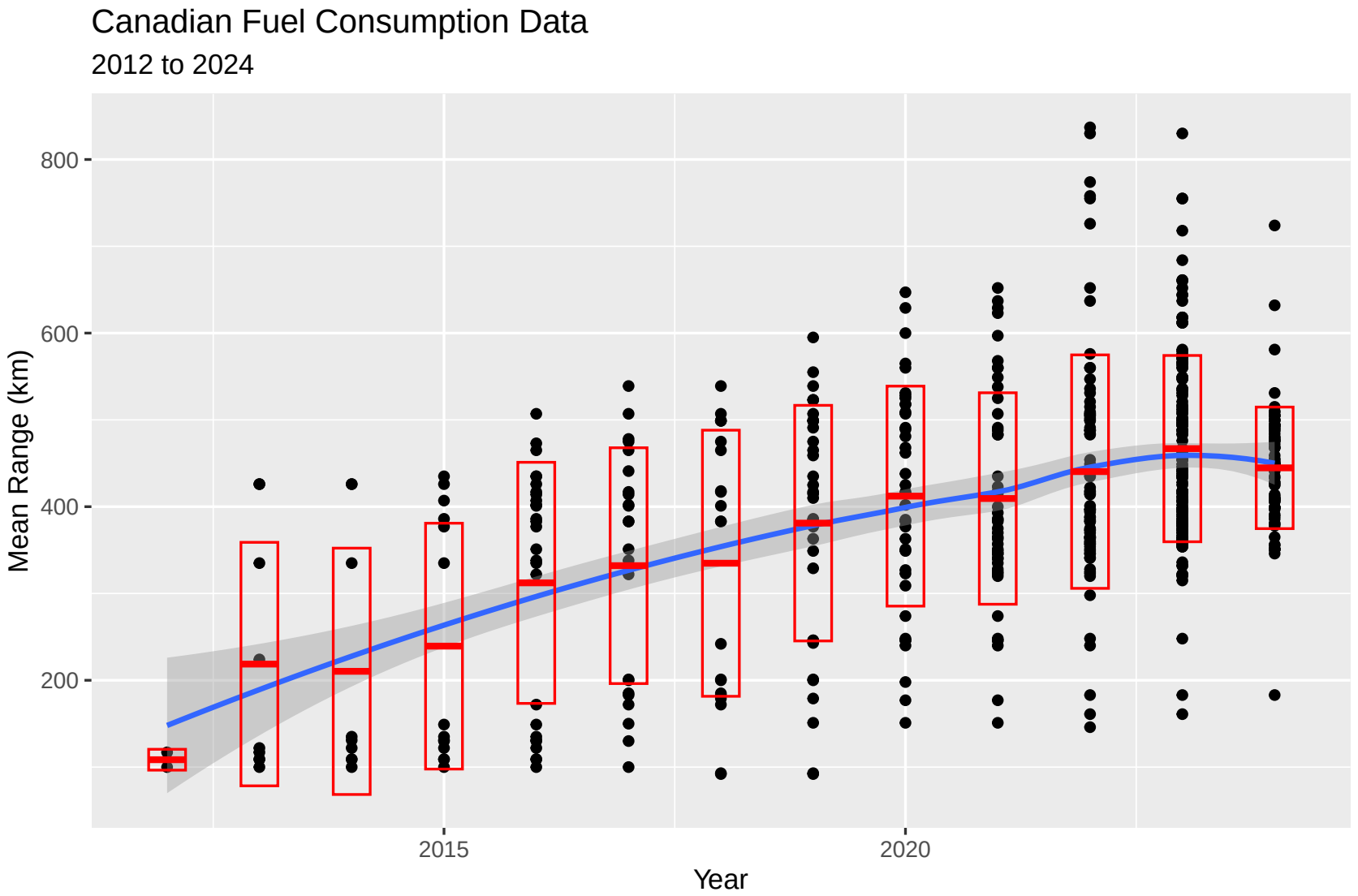
MEMORIAL UNIVERSITY

Local Regression Smoothing Plot (with cross bars)

```
e.clean %>%
  ggplot(aes(Year, Range)) +
    geom_point() +
    geom_smooth() +
    stat_summary(
        fun.data=mean_sdl,
        fun.args=list(mult=1),
        color='red',
        geom="crossbar",
        width=0.4) +
    scale_y_continuous(labels=scales::comma) +
    labs(x = 'Year', color='', y = 'Mean Range (km)',
    title='Canadian Fuel Consumption Data',
    subtitle='2012 to 2024')
```

MEMORIAL UNIVERSITY

Local Regression Smoothing Plot (with cross bars)

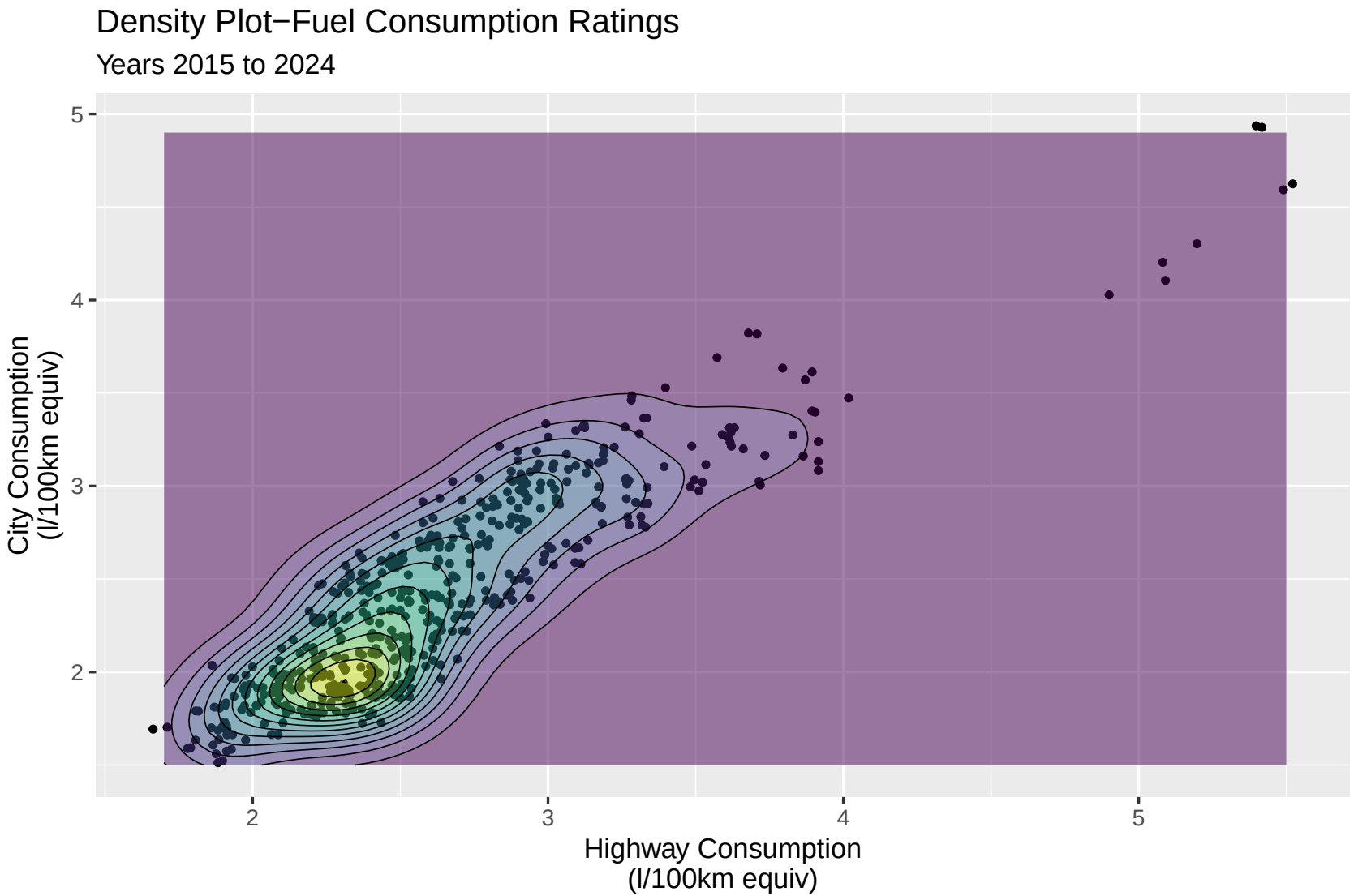


2D Density Plot

```
e.clean %>%
  ggplot(aes(x=Hwy, y=City)) +
    geom_point(color="black", size=1,
               position='jitter') +
    geom_density_2d_filled(alpha=0.5) +
    geom_density_2d(linewidth=0.25, colour='black') +
    scale_x_continuous(labels=scales::comma) +
    labs(x = 'Highway Consumption\n(l/100km equiv)',
         y = 'City Consumption\n(l/100km equiv)',
         title='Density Plot-Fuel Consumption Ratings',
         subtitle='Years 2015 to 2024') +
    theme(legend.position='none')
```

MEMORIAL UNIVERSITY

2D Density Plot

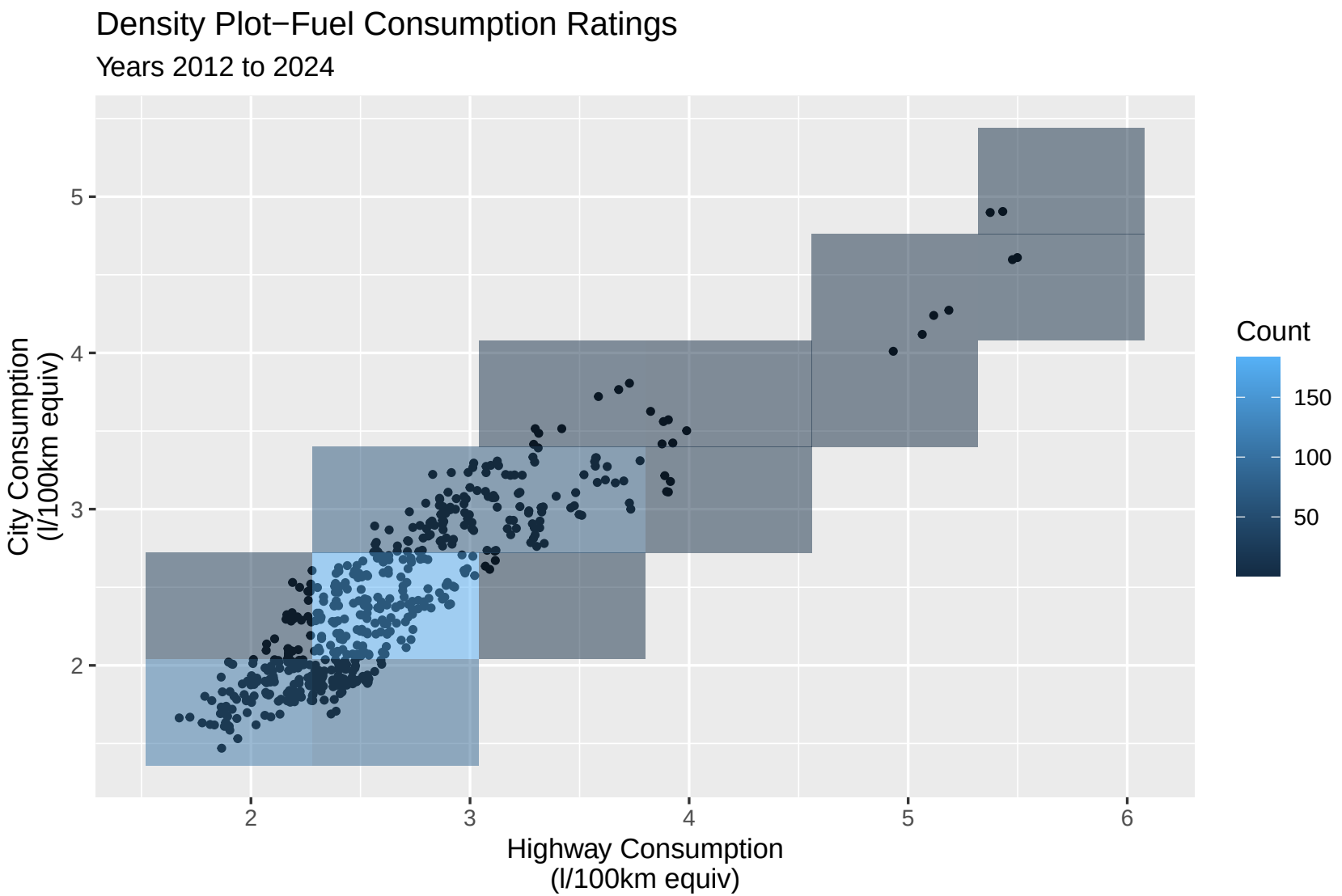


2D Bin Plot

```
e.clean %>%
  ggplot(aes(x=Hwy, y=City)) +
    geom_point(color="black", size=1,
               position='jitter') +
    geom_bin2d(alpha=0.5, bins=5) +
    scale_x_continuous(labels=scales::comma) +
    labs(x = 'Highway Consumption\n(l/100km equiv)',
         y = 'City Consumption\n(l/100km equiv)',
         fill='Count',
         title='Density Plot-Fuel Consumption Ratings',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

2D Bin Plot

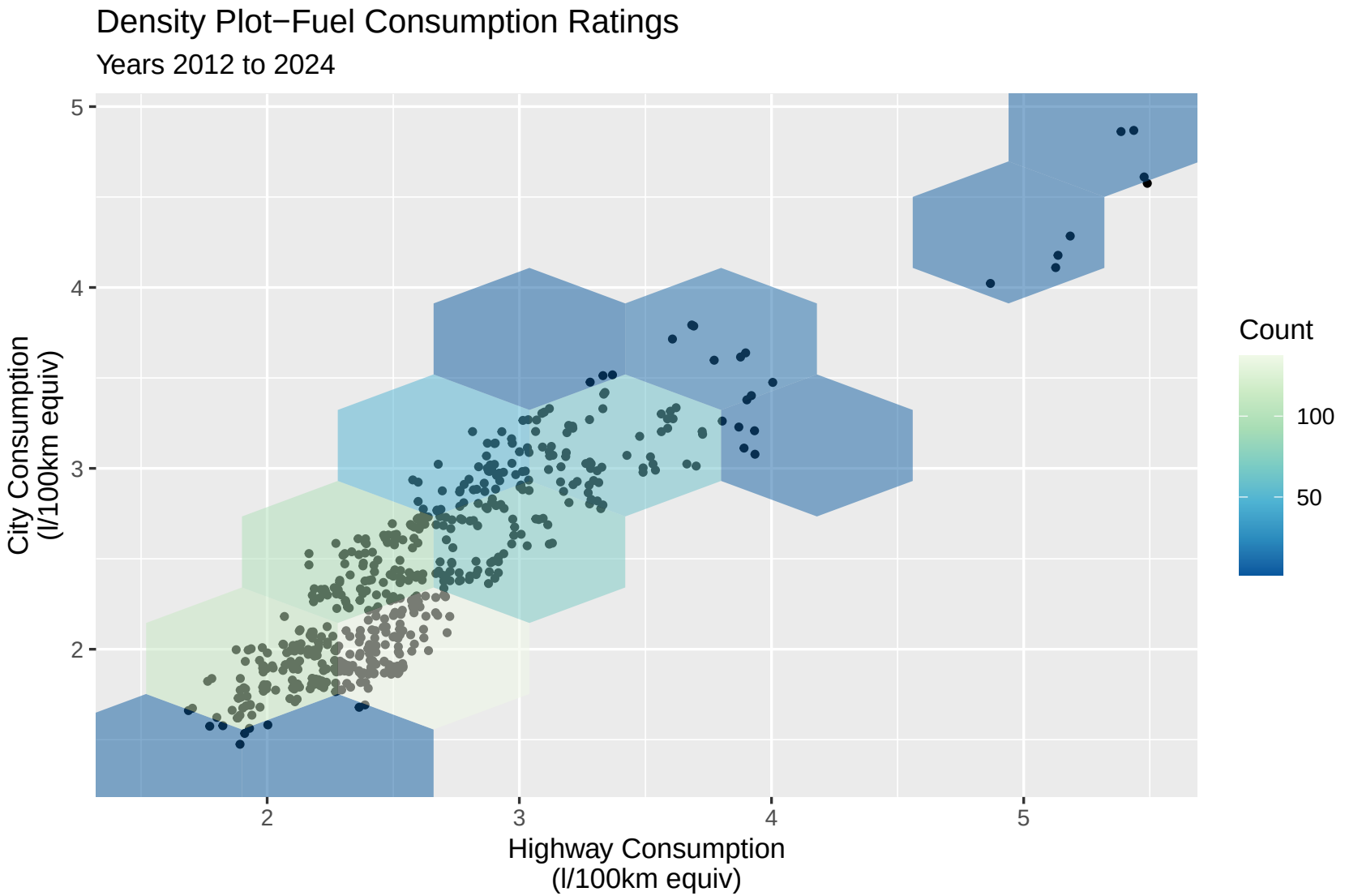

Density Plot−Fuel Consumption Ratings
Years 2012 to 2024
City Consumption (l/100km equiv)
Highway Consumption (l/100km equiv)
2
3
4
5
6
Count
150
100
50

2D Hex Plot

```
e.clean %>%
  ggplot(aes(x=Hwy, y=City)) +
    geom_point(color="black", size=1,
               position='jitter') +
    geom_hex(alpha=0.5, bins=5) +
    scale_fill_distiller(palette=4, direction=-1) +
    scale_x_continuous(labels=scales::comma) +
    labs(x = 'Highway Consumption\n(l/100km equiv)',
         y = 'City Consumption\n(l/100km equiv)',
         fill='Count',
         title='Density Plot-Fuel Consumption Ratings',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

2D Hex Plot

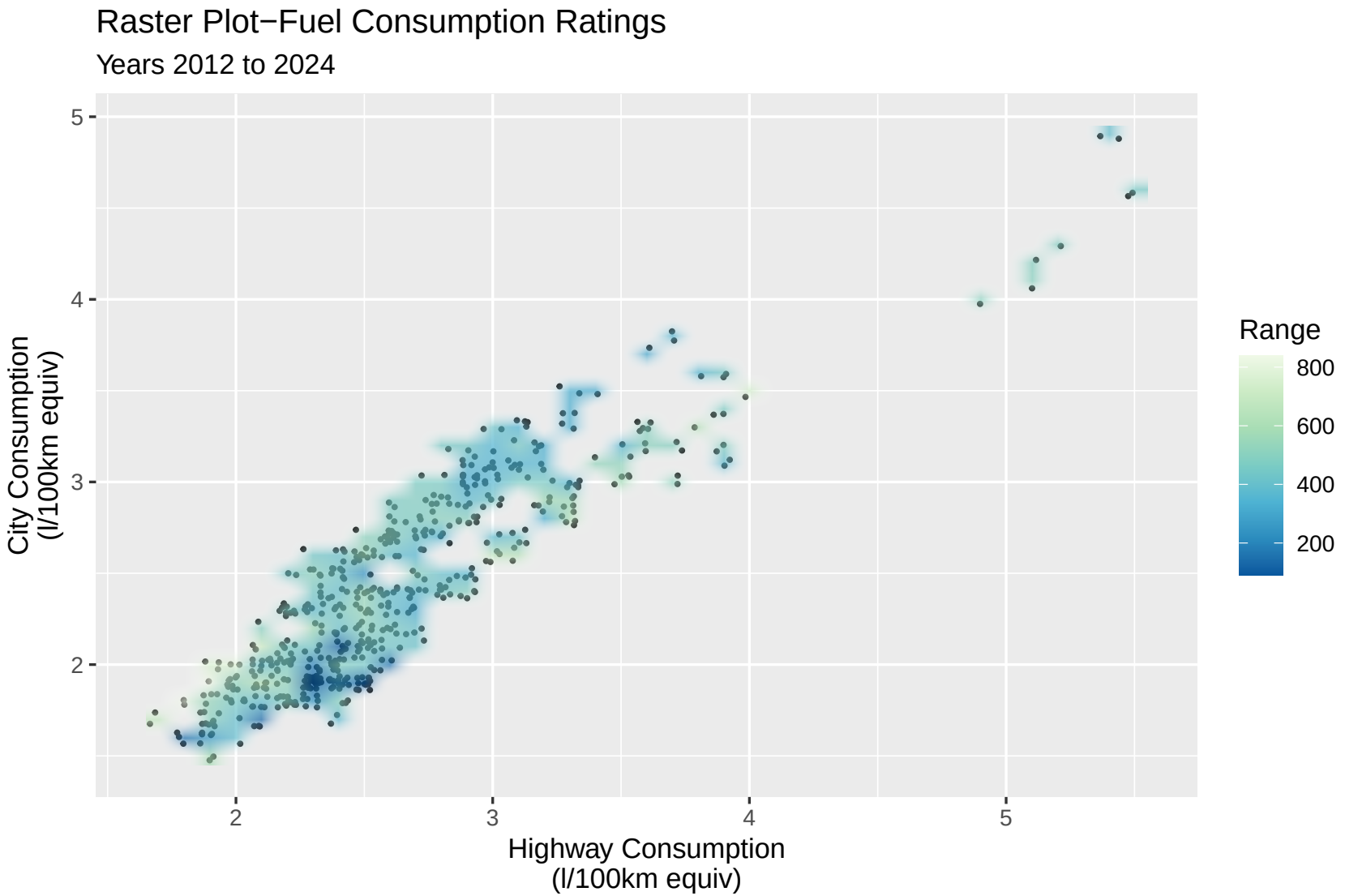

Density Plot−Fuel Consumption Ratings
Years 2012 to 2024
City Consumption (l/100km equiv)
Highway Consumption (l/100km equiv)
Count
100
50
2
3
4
5

3D Raster Plot

```
e.clean %>%
  ggplot(aes(x=Hwy, y=City)) +
    geom_point(color="black", size=0.5,
               position='jitter') +
    geom_raster(aes(fill=Range), alpha=0.7,
                interpolate=TRUE) +
    scale_fill_distiller(palette=4, direction=-1) +
    scale_x_continuous(labels=scales::comma) +
    labs(x = 'Highway Consumption\n(l/100km equiv)',
         y = 'City Consumption\n(l/100km equiv)',
         fill='Range',
         title='Raster Plot-Fuel Consumption Ratings',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

3D Raster Plot

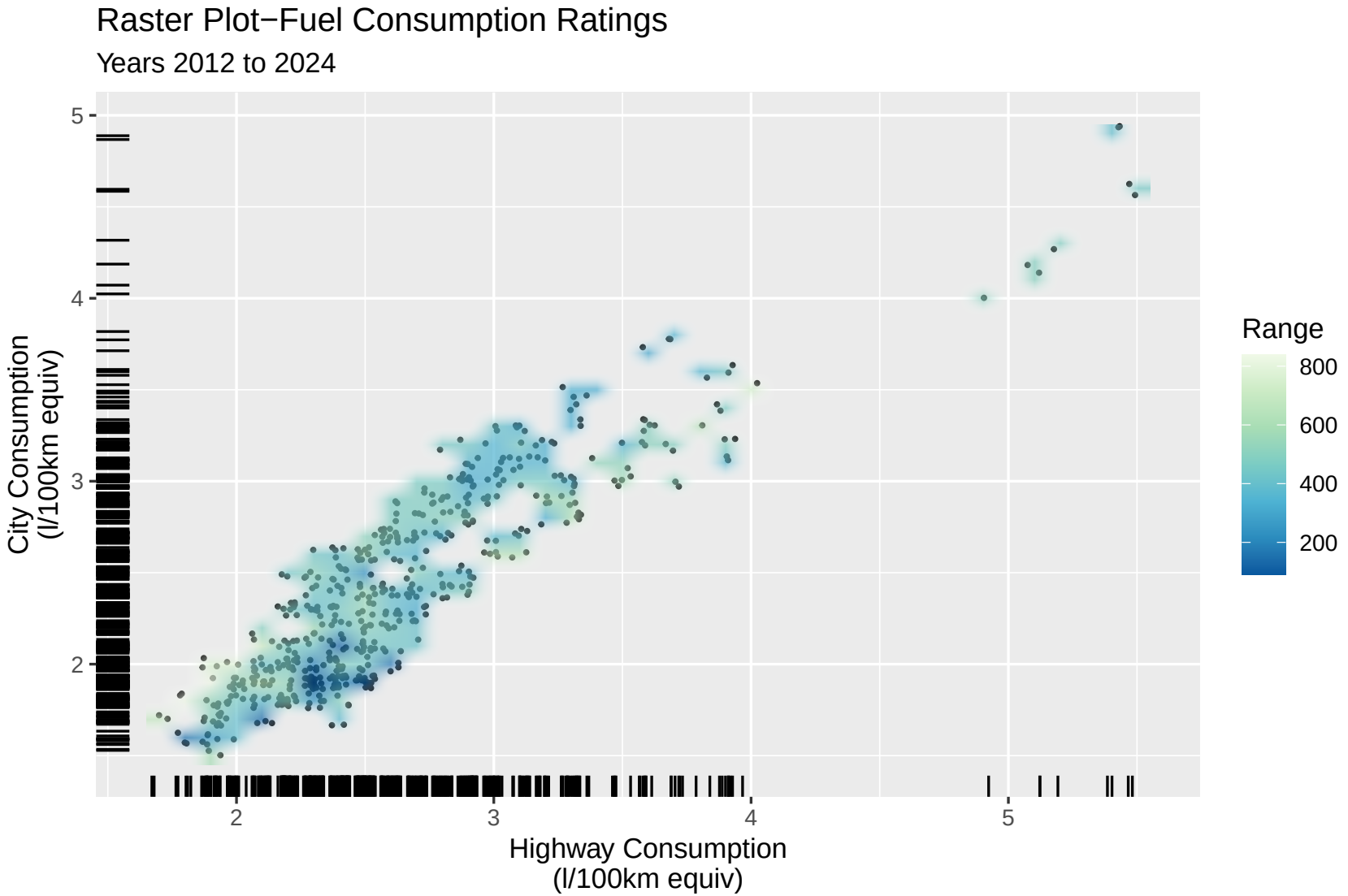


3D Raster Plot with Rug

```
e.clean %>%
  ggplot(aes(x=Hwy, y=City)) +
    geom_point(color="black", size=0.5,
               position='jitter') +
    geom_raster(aes(fill=Range), alpha=0.7,
                interpolate=TRUE) +
    geom_rug(position='jitter') +
    scale_fill_distiller(palette=4, direction=-1) +
    scale_x_continuous(labels=scales::comma) +
    labs(x = 'Highway Consumption\n(l/100km equiv)',
         y = 'City Consumption\n(l/100km equiv)',
         fill='Range',
         title='Raster Plot-Fuel Consumption Ratings',
         subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

3D Raster Plot (with rug)



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
   - Add error bars to this chart
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:

- The `read.csv()` function can read from a URL
- The data is de-normalized, use the `unique()` function to get accurate film counts for exercise 1
- Use the `year()` and `week()` functions from the `lubridate` package (another package of the Tidyverse set)

MEMORIAL UNIVERSITY