Data Visualization

Joerg Evermann

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

November 3, 2023

MEMORIAL UNIVERSITY

Legal Stuff

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!)

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

Visualization Process

1. Define objective and target audience
2. Acquire data
   - Identify sources
3. Parse data
   - Convert into appropriate format
4. Filter data
   - Select relevant information
5. Mine data
   - Apply statistical or data mining strategies for trends, correlations, etc.
6. Represent findings
   - Select a visualization type
7. Refine representation
   - Enhance for clarity and information content
8. Interact with representation
   - Provide user-controllable tools

Honesty in Visualization

General Guidelines

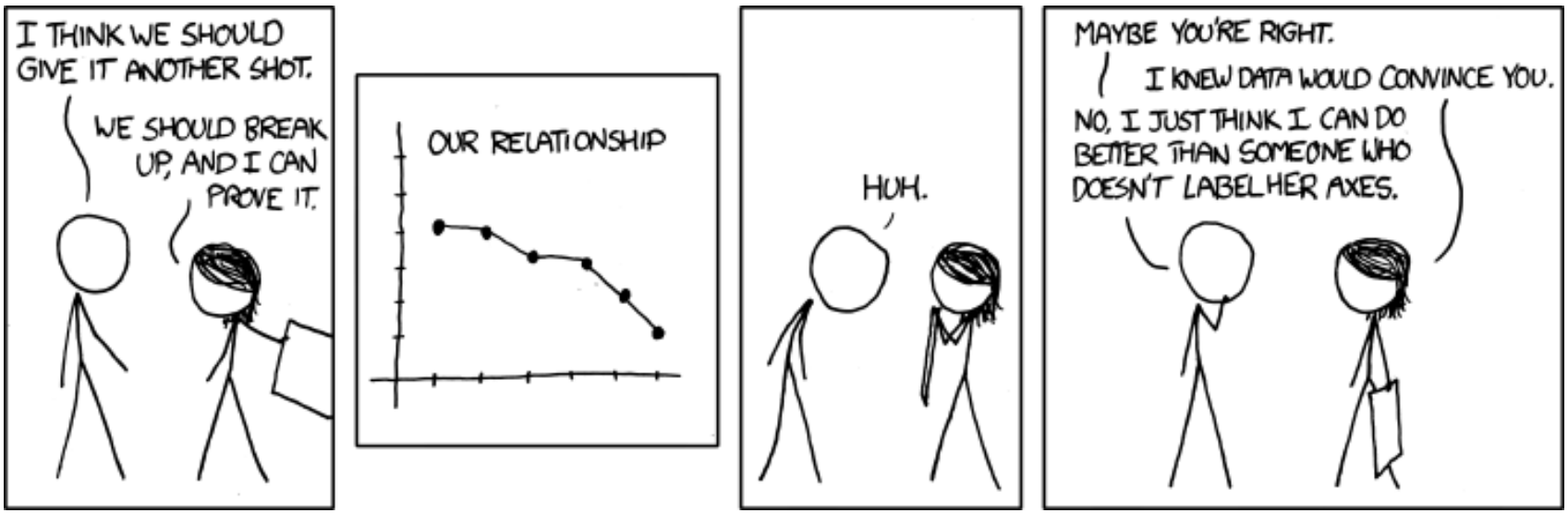
- Do not deceive your target audience
- Do not diminish or hide relationships or trends
- Do not exaggerate relationships or trends
- Do not confuse or obfuscate

Honesty in Visualization

Specific ”no-nos”

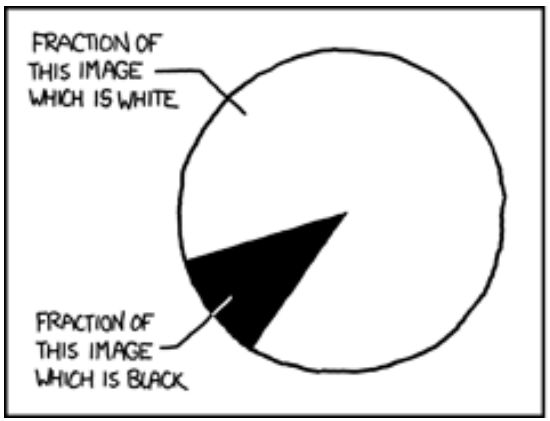
- Graph unrelated data to suggest non-existent relationships
- Scale multiple vertical axes to suggest correlations
- Truncate axes to hide or exaggerate trend
- Plot cumulative growth to hide trend
- Use maps for non-geographic data
- Use incomplete data (”cherry-picking”)
- Use invalid data

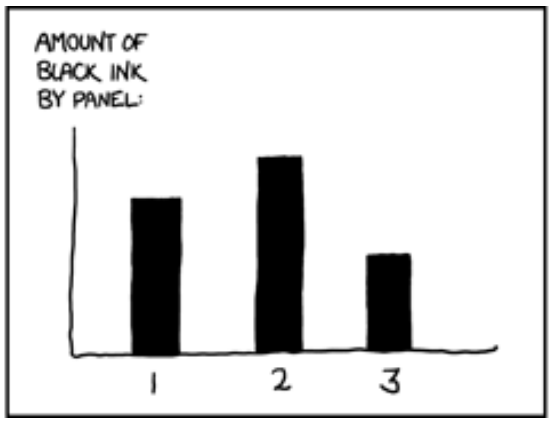
Label your Axes (XKCD)

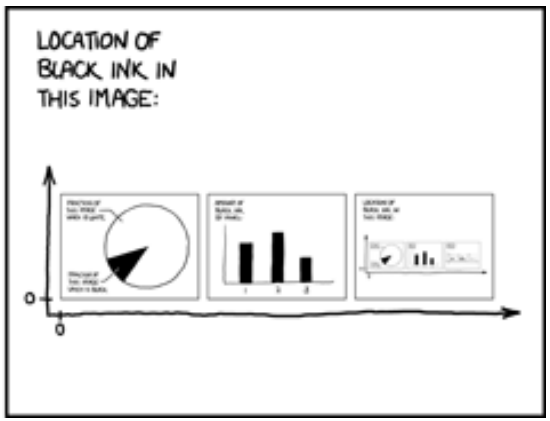


MEMORIAL UNIVERSITY

Use Meaningful Data (XKCD)

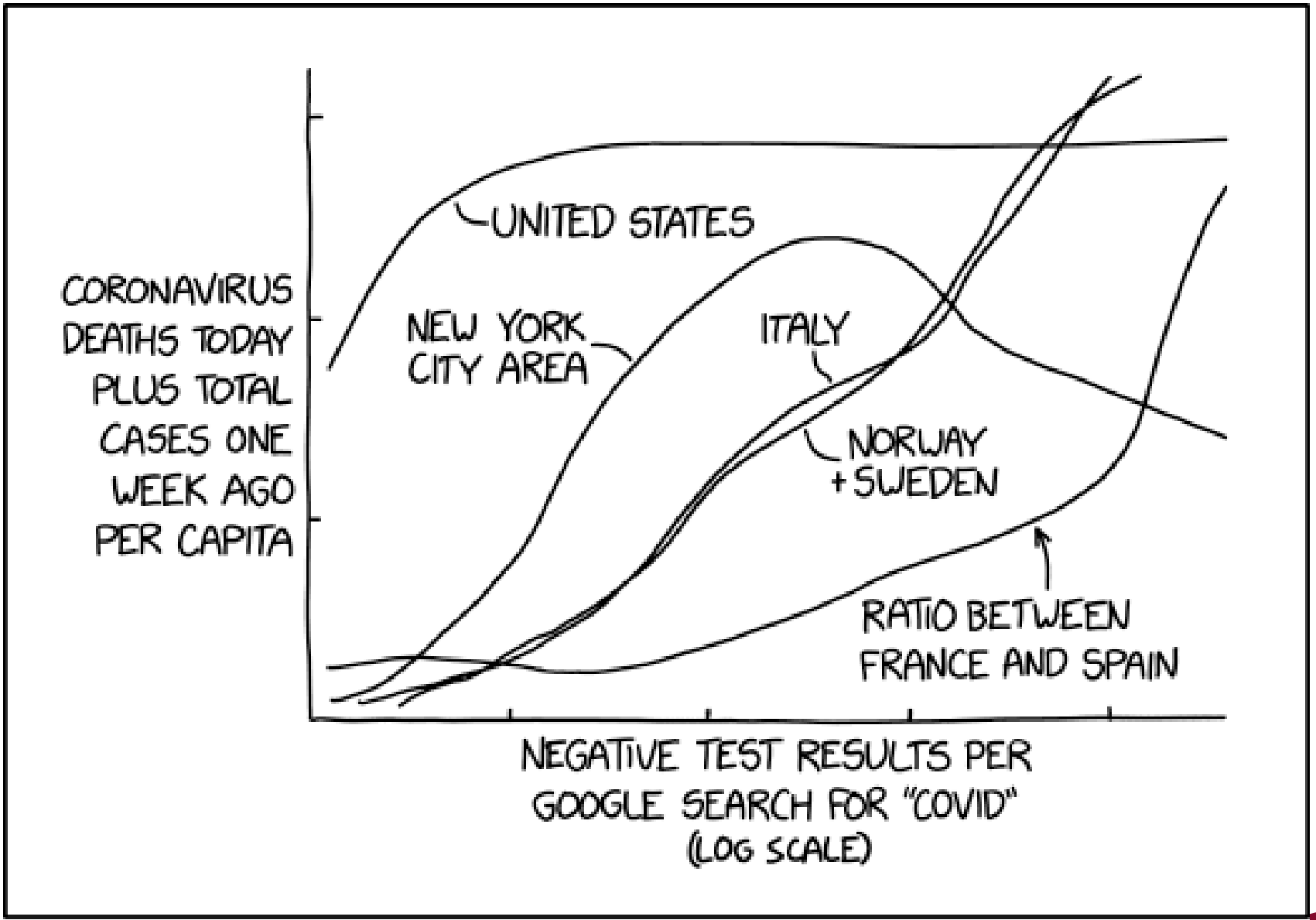






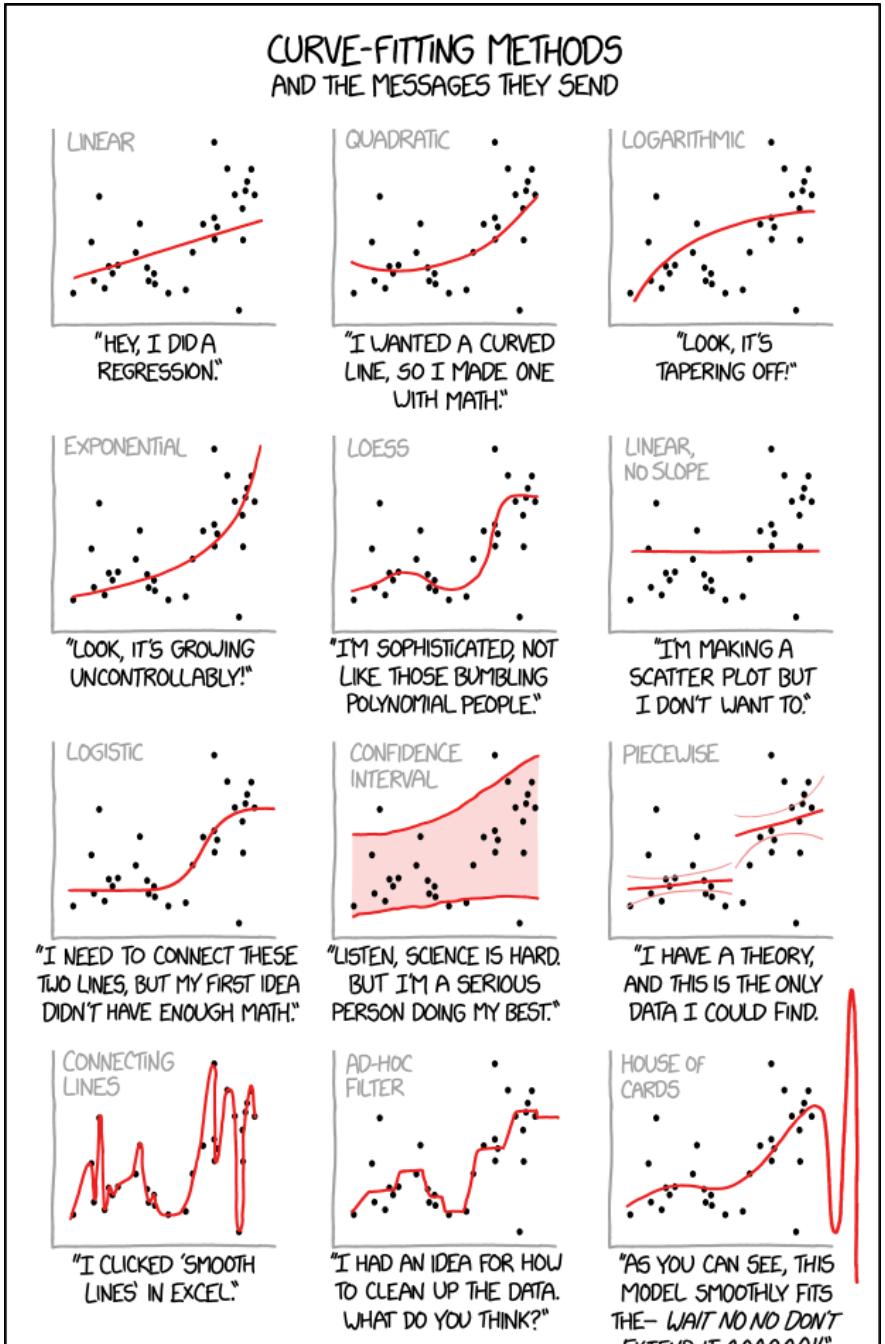
MEMORIAL UNIVERSITY

Use Related Data (XKCD)



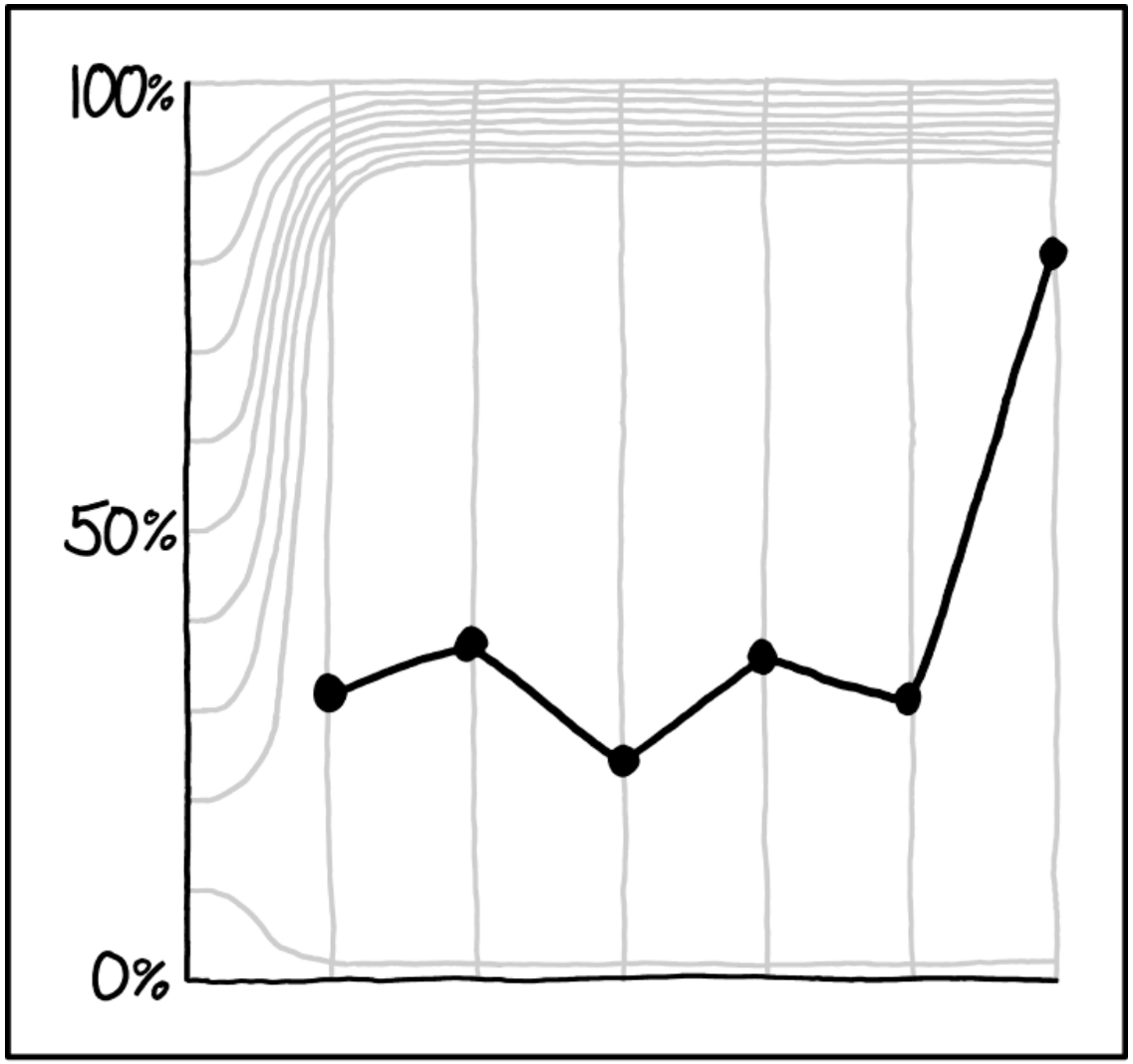
I'M A HUGE FAN OF WEIRD GRAPHS, BUT EVEN I ADMIT SOME OF THESE CORONAVIRUS CHARTS ARE LESS THAN HELPFUL.

Do Not Mislead (XKCD)



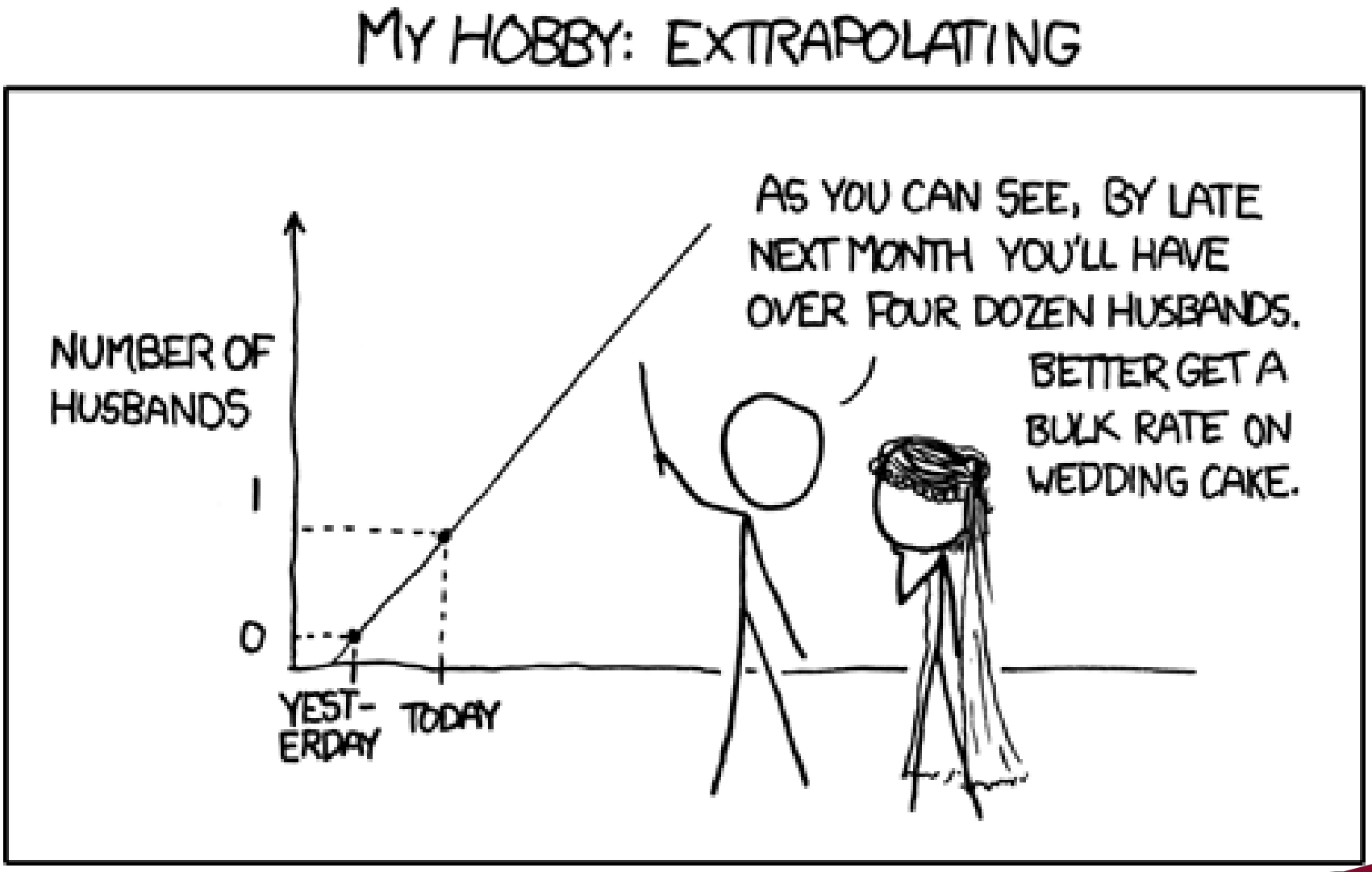
MEMORIAL UNIVERSITY

Choose Your Axes Meaningfully

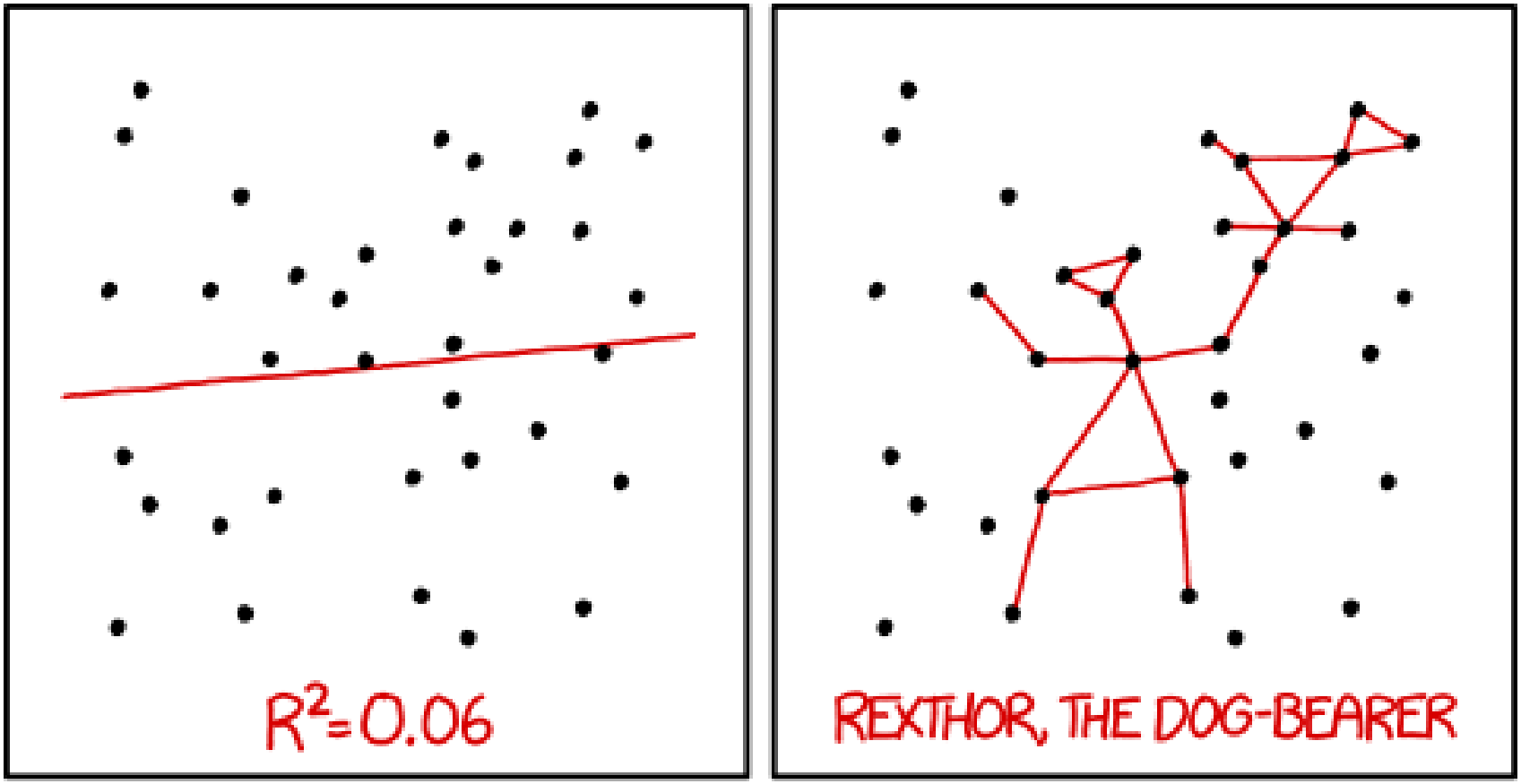


PEOPLE HAVE WISED UP TO THE "CAREFULLY CHOSEN Y-AXIS RANGE" TRICK, SO WE MISLEADING GRAPH MAKERS HAVE HAD TO GET CREATIVE.

Be Careful When Extrapolating (XKCD)

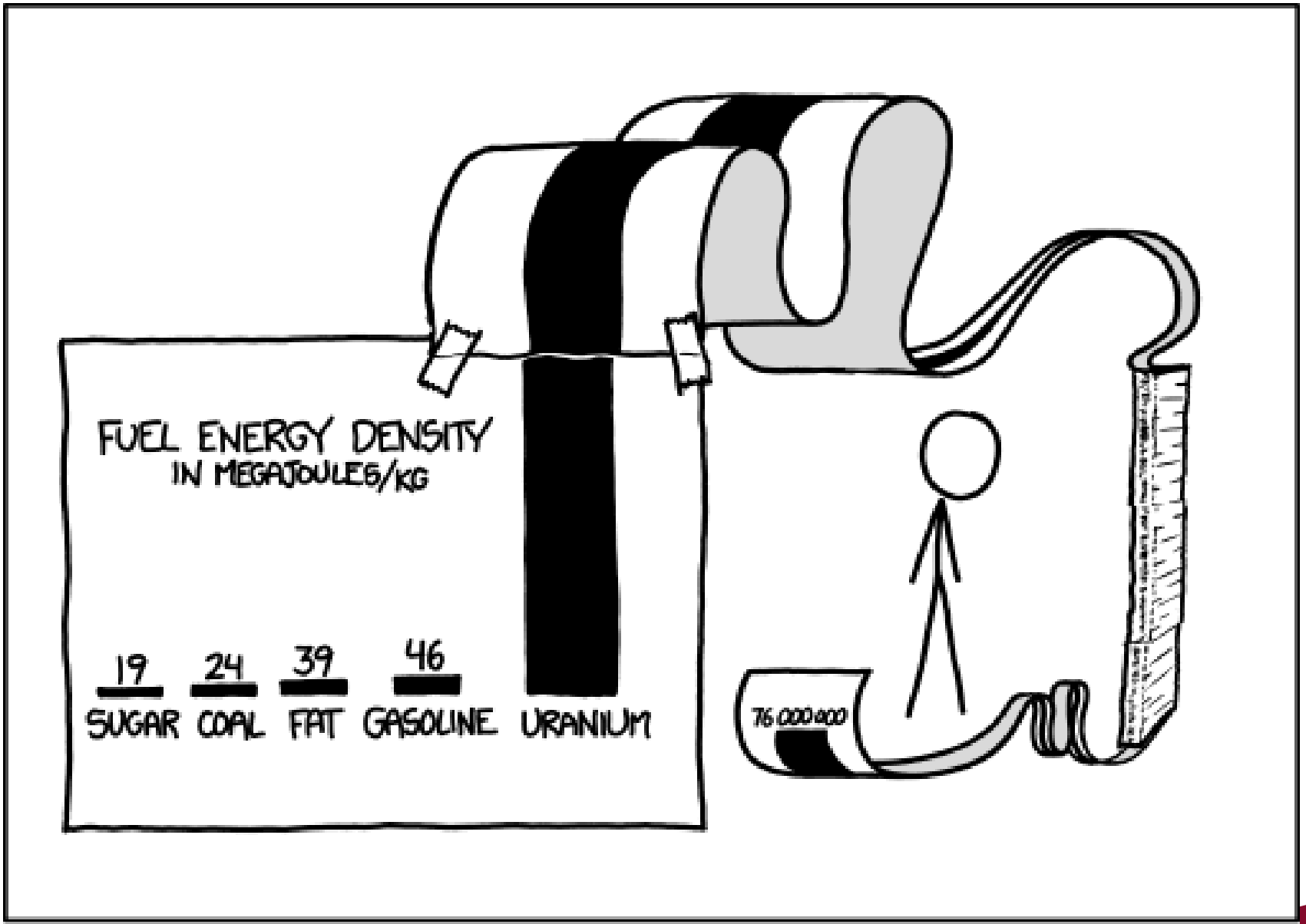


Verify Trends (XKCD)



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.

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

MEMORIAL UNIVERSITY

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

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

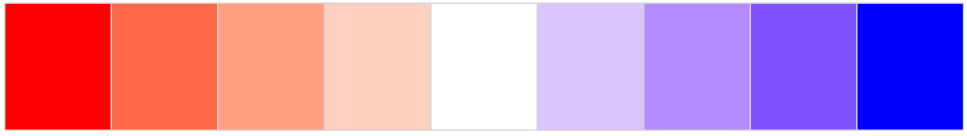
- **Monochrome/Sequential**, i.e. light to dark within a single colour
- **Divergent**, i.e. from one colour to another via white
- **Spectral**, uses a large number of colours
- **Bivariate**, e.g. combination or RGB and CMY

Colour palettes may be continuous, discrete, or categorical

Diverging

Sequential

Diverging



Spectral

MEMORIAL UNIVERSITY

CVD (Colour Vision Deficiency)

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

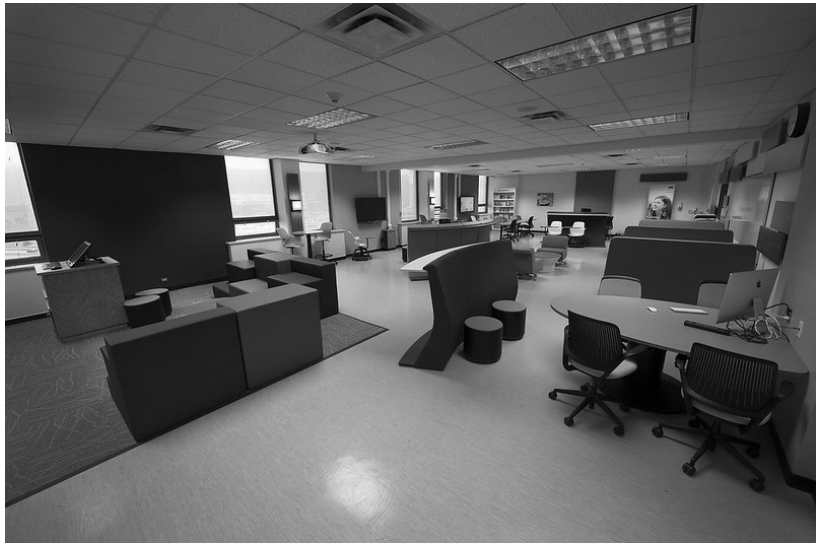
1 in 12 men have CVD
1 in 200 women have CVD
2.6 million Canadians are colour blind

Original

MUN Faculty of Education Class Room

MEMORIAL UNIVERSITY

Simulated Colour Vision Deficiencies



Monochromatism

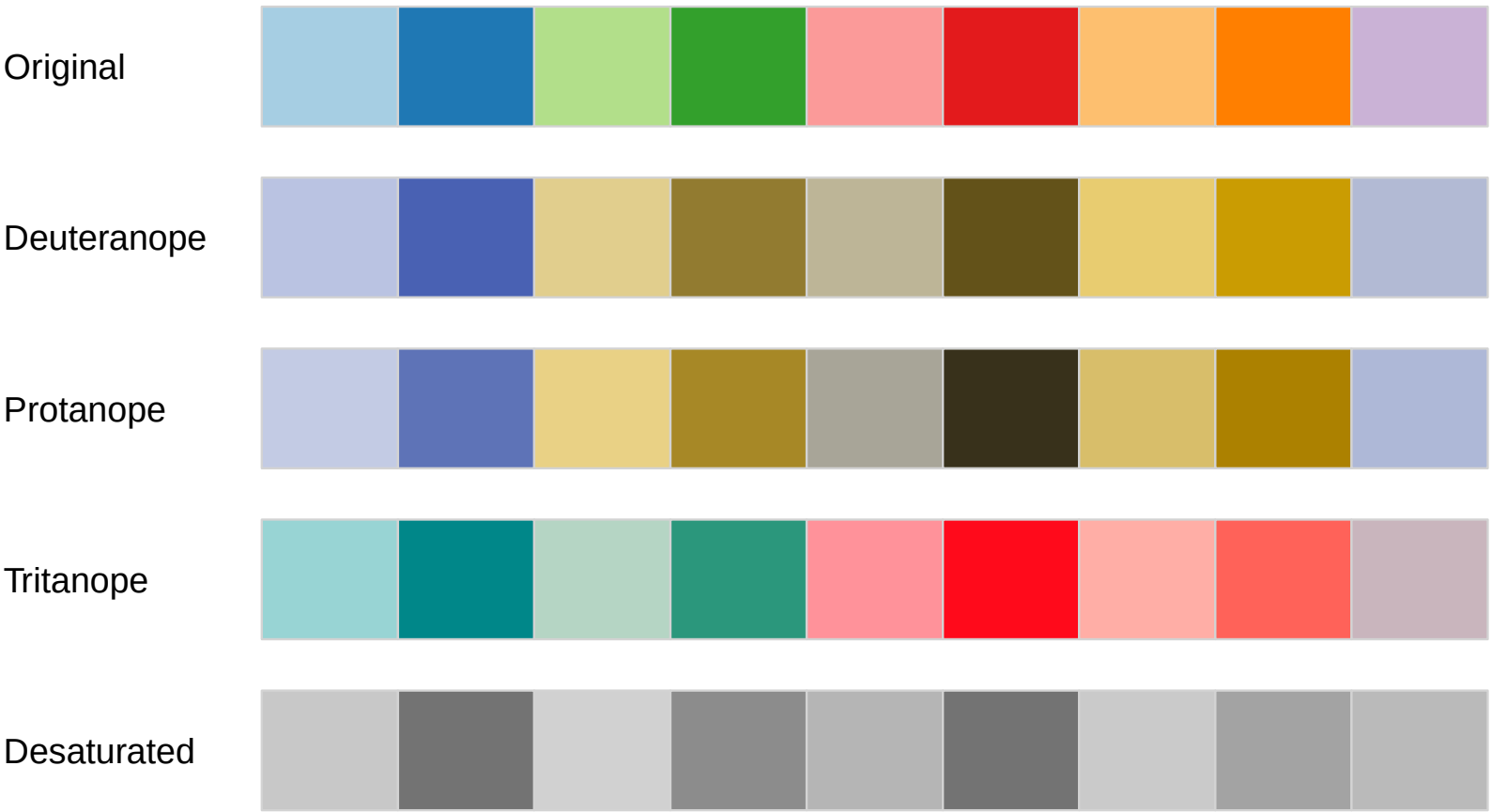
Protanopia

Deuteranopia

Tritanopia

Example: Colourbrewer Palette ”Paired”

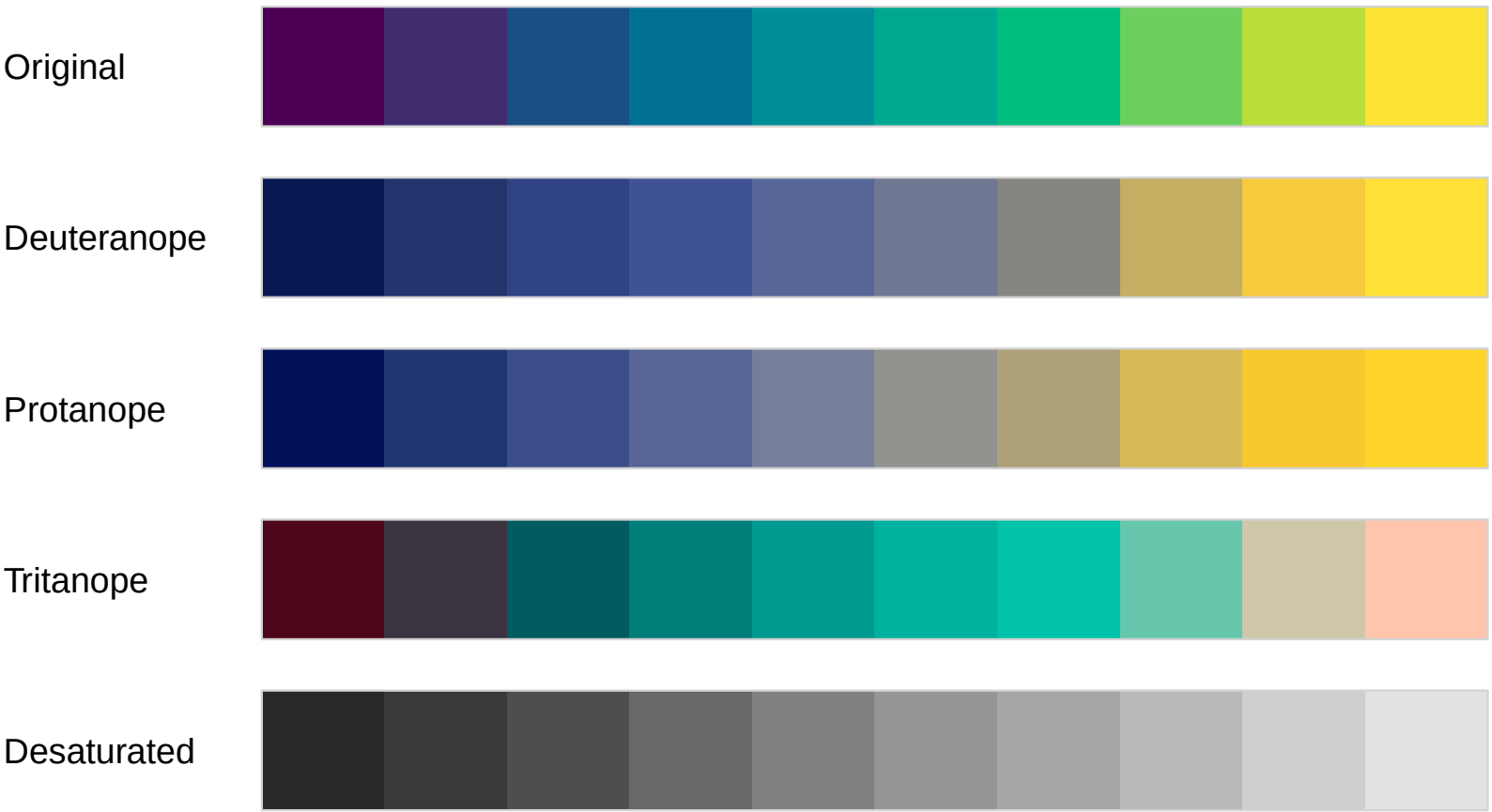
Brewer Paired



MEMORIAL UNIVERSITY

Viridis Colour Palette

Viridis Palette



MEMORIAL UNIVERSITY

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

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

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)

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 1 (”Contracts”)

- Government of Canada, Open Government Portal
- Proactive Publication – Contracts – Contracts over $10,000
- Last updated Sep 26, 2023
- https://open.canada.ca/data/en/dataset/d8f85d91-7dec-4fd1-8055-483b77225d8b/resource/fac950c0-00d5-4ec1-a4d3-9cbebf98a305

Column	Data Type
contract_date	Date
contract_value	Numeric
commodity_type	Discrete[1]
country_of_vendor	Discrete
province_of_vendor	Discrete
number_of_bids	Numeric
solicitation_procedure	Discrete

[1] Construction, Goods, Services

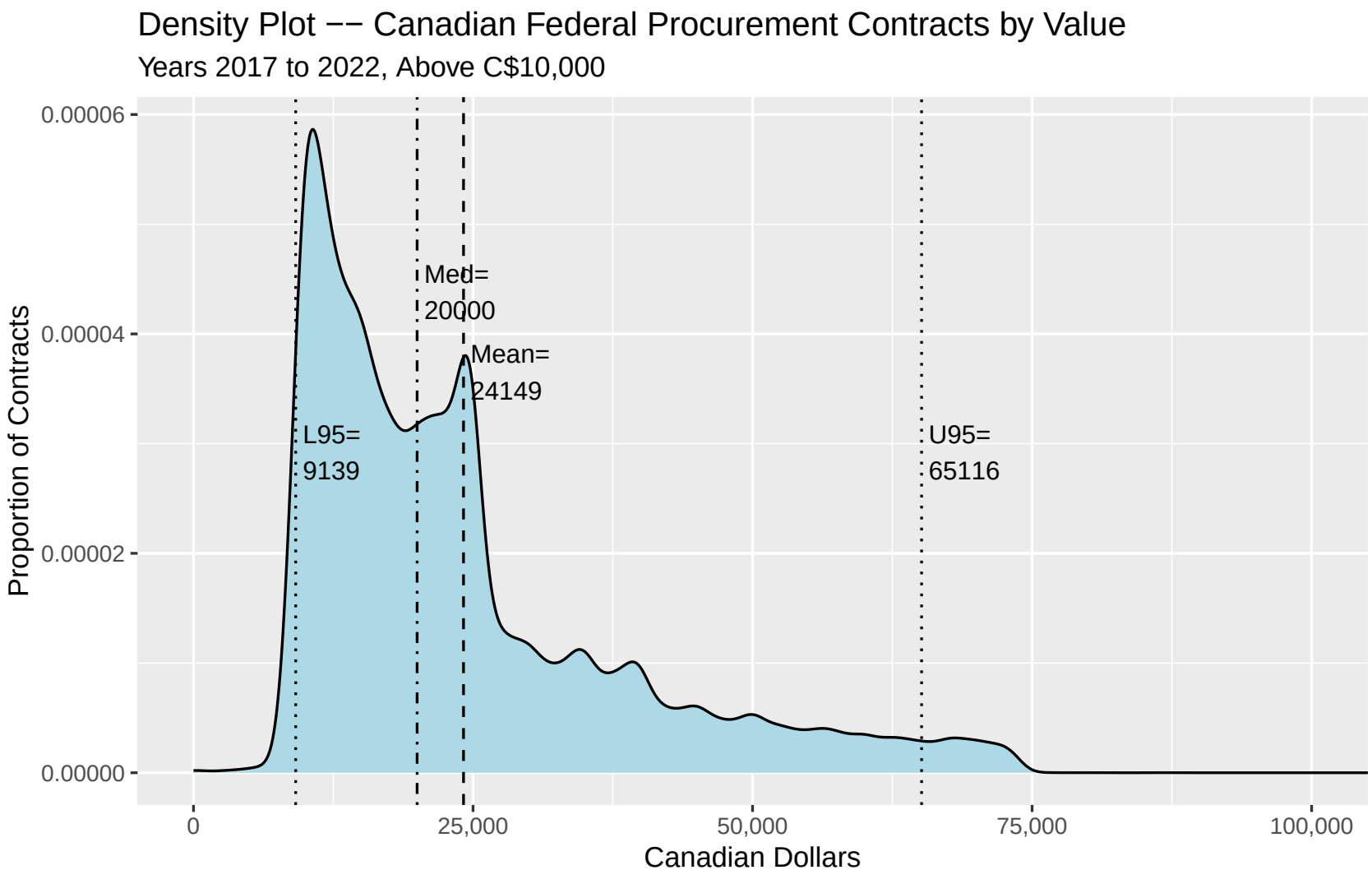
MEMORIAL UNIVERSITY

Data Preparation

```
d.clean <- read.csv('contracts.cleaned.csv')

d.clean$contract_date <- as.Date(d.clean$contract_date)
d.clean$contract_period_start <- as.Date(d.clean$contract_period_start)
d.clean$economic_object_code <- as.factor(d.clean$economic_object_code)
d.clean$commodity_type <- as.factor(d.clean$commodity_type)
d.clean$commodity_code <- as.factor(d.clean$commodity_code)
d.clean$country_of_vendor <- as.factor(d.clean$country_of_vendor)
d.clean$solicitation_procedure <- as.factor(d.clean$solicitation_procedure)
d.clean$limited_tendering_reason <- as.factor(d.clean$limited_tendering_reason)
d.clean$trade_agreement_exceptions <- as.factor(d.clean$trade_agreement_exceptions)
d.clean$award_criteria <- as.factor(d.clean$award_criteria)

d.clean <- d.clean %>% mutate(vendor_province = case_when(
    str_starts(vendor_postal_code, 'A') ~ 'NL',
    str_starts(vendor_postal_code, 'B') ~ 'NS',
    str_starts(vendor_postal_code, 'C') ~ 'PE',
    str_starts(vendor_postal_code, 'E') ~ 'NB',
    str_starts(vendor_postal_code, 'G|J') ~ 'QC',
    str_starts(vendor_postal_code, 'L|M|N|K|P') ~ 'ON',
    str_starts(vendor_postal_code, 'R') ~ 'MB',
    str_starts(vendor_postal_code, 'S') ~ 'SK',
    str_starts(vendor_postal_code, 'T') ~ 'AB',
    str_starts(vendor_postal_code, 'V') ~ 'BC',
    str_starts(vendor_postal_code, 'X') ~ 'NT',
    str_starts(vendor_postal_code, 'Y') ~ 'YK',
    TRUE ~ 'UNKWN'))
d.clean$vendor_province <- as.factor(d.clean$vendor_province)
```

MEMORIAL UNIVERSITY

Density Chart – 1 Variable, 1 Series



MEMORIAL UNIVERSITY

Density Chart – 1 Variable, 1 Series

```
mean_v <- d.clean %>%
   filter(original_value < quantile(original_value, .90)) %>%
   filter(contract_value < quantile(contract_value, .90)) %>%
   pivot_longer(  c(original_value, contract_value),
                  names_to="value_type",
                  values_to="value") %>%
   summarize(
       mean_v = mean(value),
       median_v = median(value),
       lower95=quantile(value, .025),
       upper95=quantile(value, .975),
       maxdensity = max(density(value)$y))
```

MEMORIAL UNIVERSITY

Density Chart – 1 Variable, 1 Series

```
d.clean %>%
   filter(original_value < quantile(original_value, .90)) %>%
   filter(contract_value < quantile(contract_value, .90)) %>%
   pivot_longer(  c(original_value, contract_value),
                  names_to="value_type",
                  values_to="value") %>%
   ggplot(aes(value)) +
     geom_density(kernel='gaussian', n=4096, fill='lightblue') +
     scale_x_continuous(labels=scales::comma) +
     scale_y_continuous(labels=scales::comma) +
     labs(
         x = 'Canadian Dollars',
         y = 'Proportion of Contracts',
         title='Density Plot -- Canadian Federal Procurement Contracts
         subtitle='Years 2017 to 2022, Above C$10,000',
         caption='Lower and Upper 95 percentile, median and mean') +
     coord_cartesian(xlim=c(0, 100000)) +
```

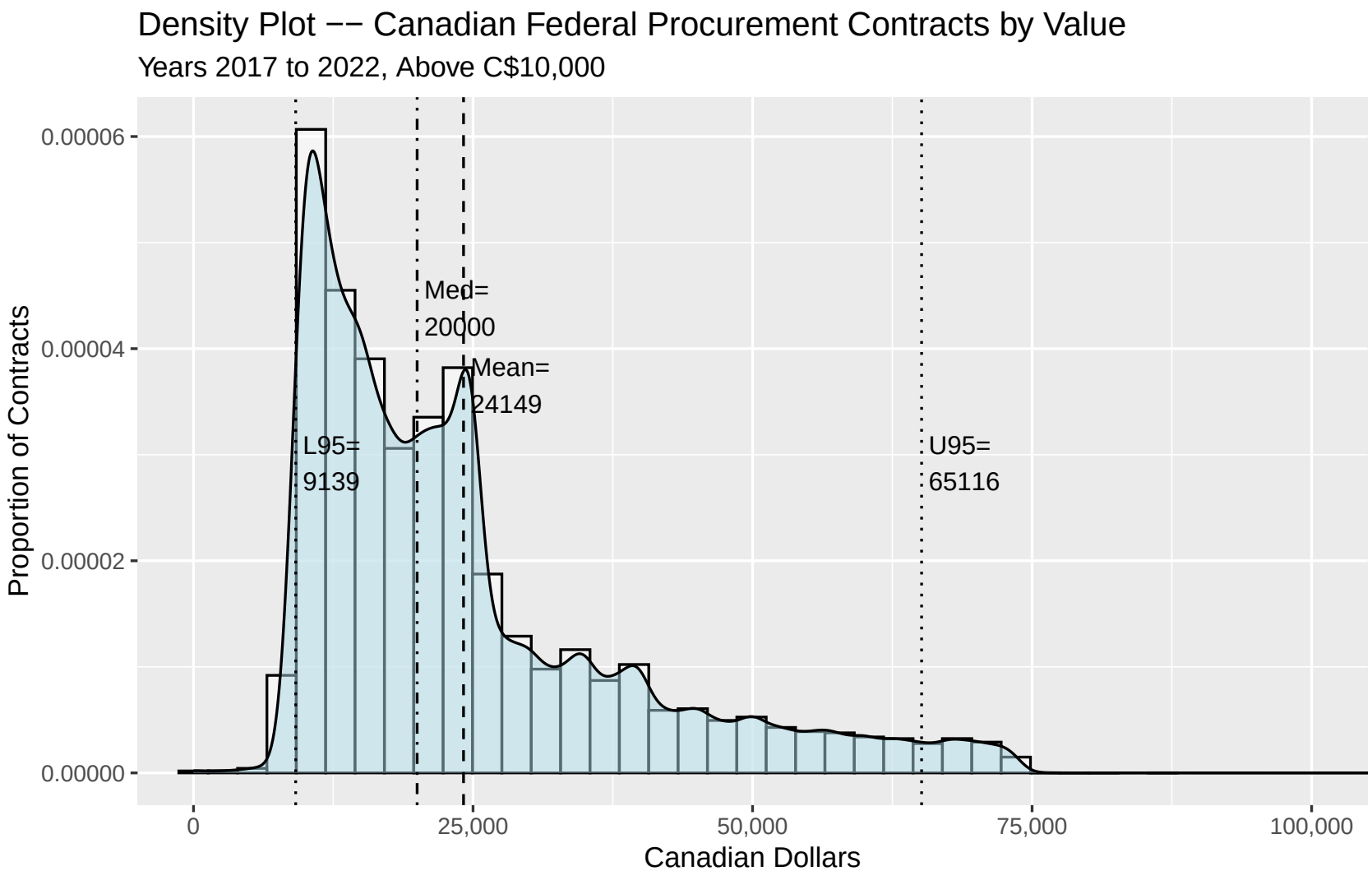
MEMORIAL UNIVERSITY

Density Chart — 1 Variable, 1 Series

```
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') +
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), se
  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

Histogram – 1 Variable, 1 Series

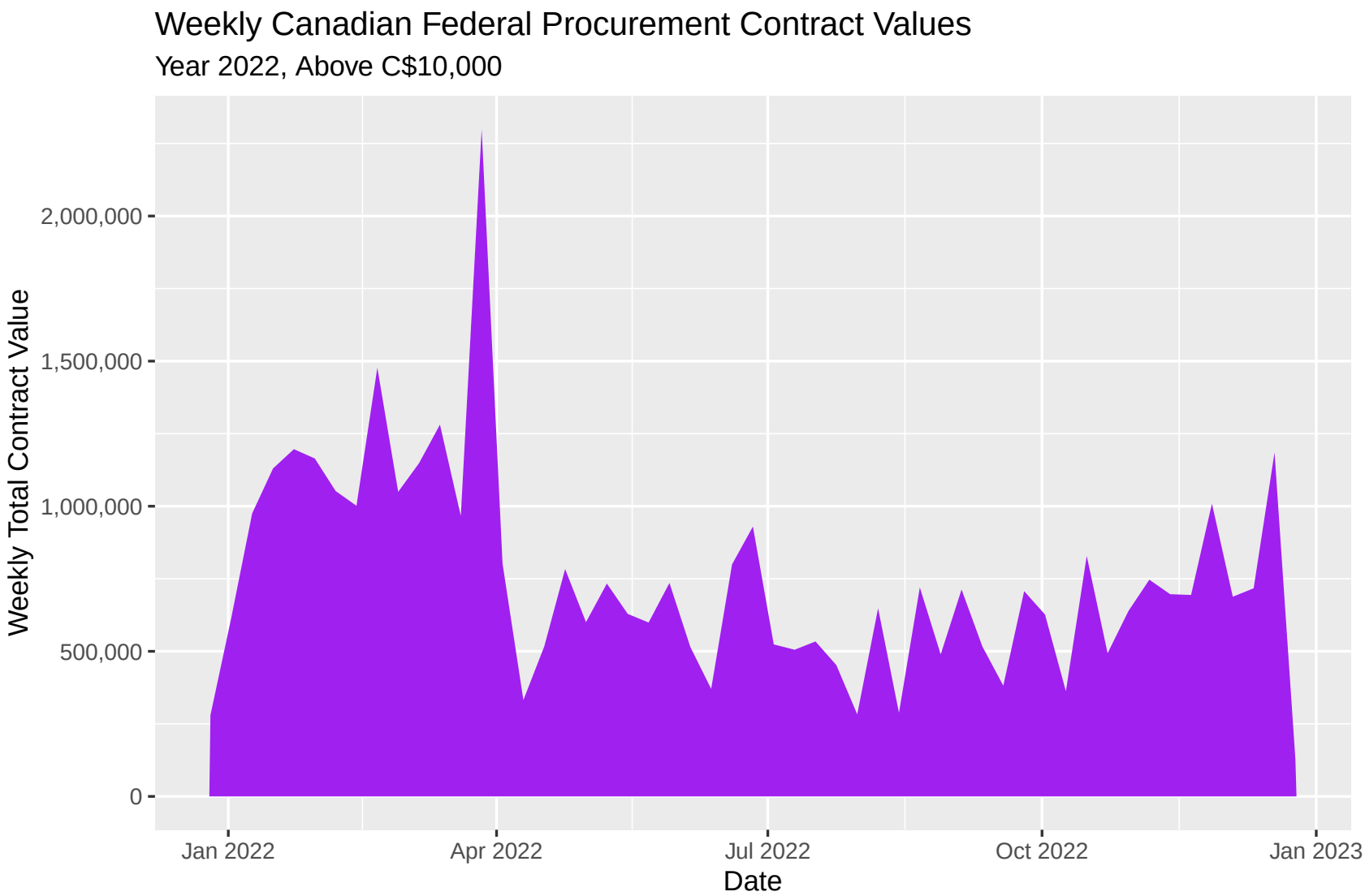


MEMORIAL UNIVERSITY

Histogram – 1 Variable, 1 Series

```
...

ggplot(aes(value)) +
 geom_histogram(
          aes(y=..density..),
          bins=50,
          fill='white',
          color='black',
          alpha=0.5) +

...
```

Area Chart – 1 Variable, 1 Series



MEMORIAL UNIVERSITY

Area Chart – 1 Variable, 1 Series

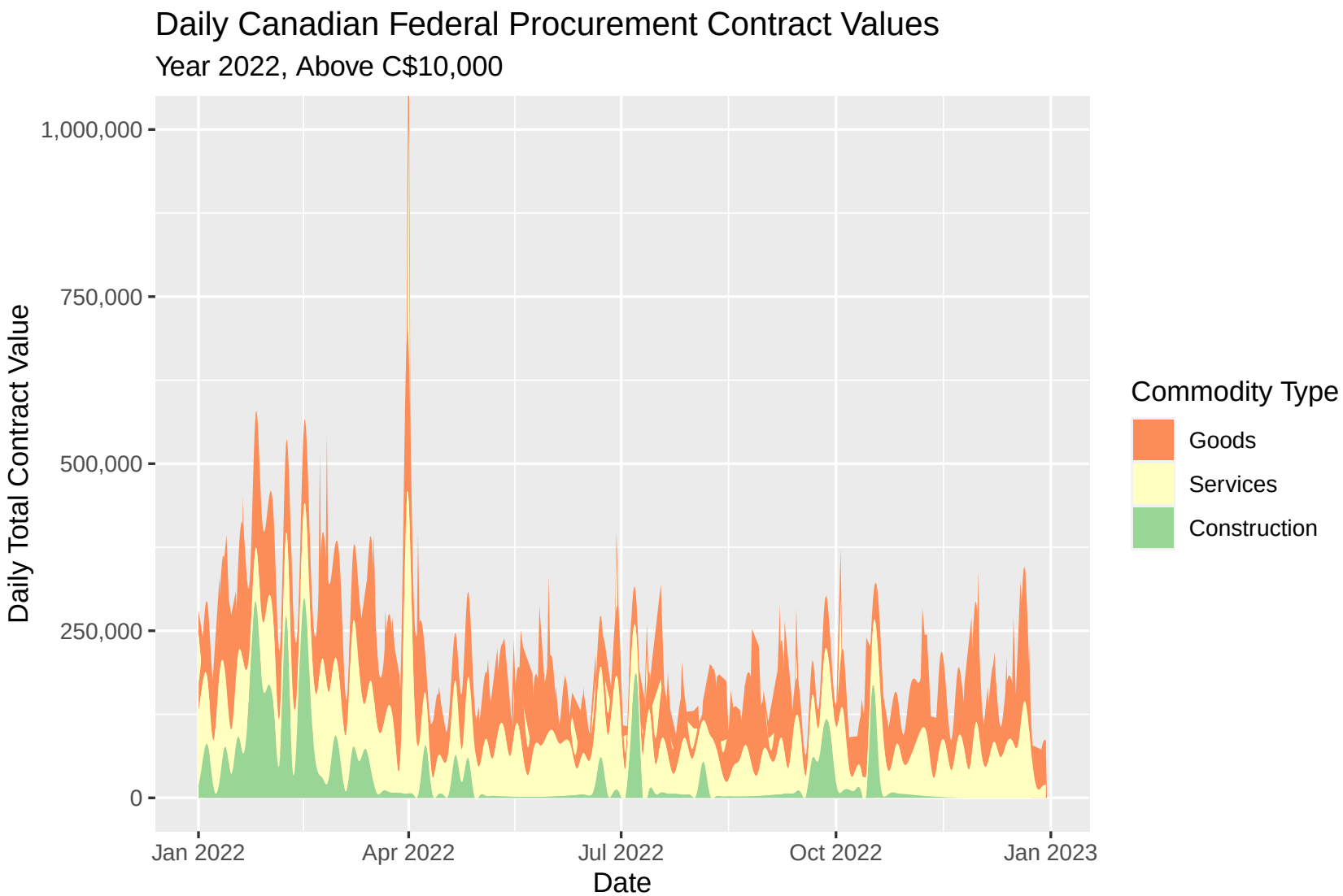
```
d.clean %>%
   filter(original_value < quantile(original_value, .90)) %>%
   filter(contract_value < quantile(contract_value, .90)) %>%
   filter(contract_date >= '2022-01-01') %>%
   filter(contract_date <= '2022-12-31') %>%
   mutate(commodity_type=case_when(
                commodity_type=='C' ~ 'Construction',
                commodity_type=='G' ~ 'Goods',
                commodity_type=='S' ~ 'Services')) %>%
   mutate(contract_quarter = floor_date(contract_date, 'week')) %>%
   group_by(contract_quarter) %>%
   summarize(daily_value = sum(contract_value)) %>%
   ungroup() %>%
```

MEMORIAL UNIVERSITY

Area Chart – 1 Variable, 1 Series

```
ggplot(aes(contract_quarter, daily_value)) +
 geom_area(fill='purple') +
 scale_y_continuous(labels=scales::comma) +
 labs( x = 'Date',
       y = 'Weekly Total Contract Value',
       title='Weekly Canadian Federal Procurement Contract Values',
       subtitle='Year 2022, Above C$10,000')
```

Area Chart – 1 Variable, 3 Series



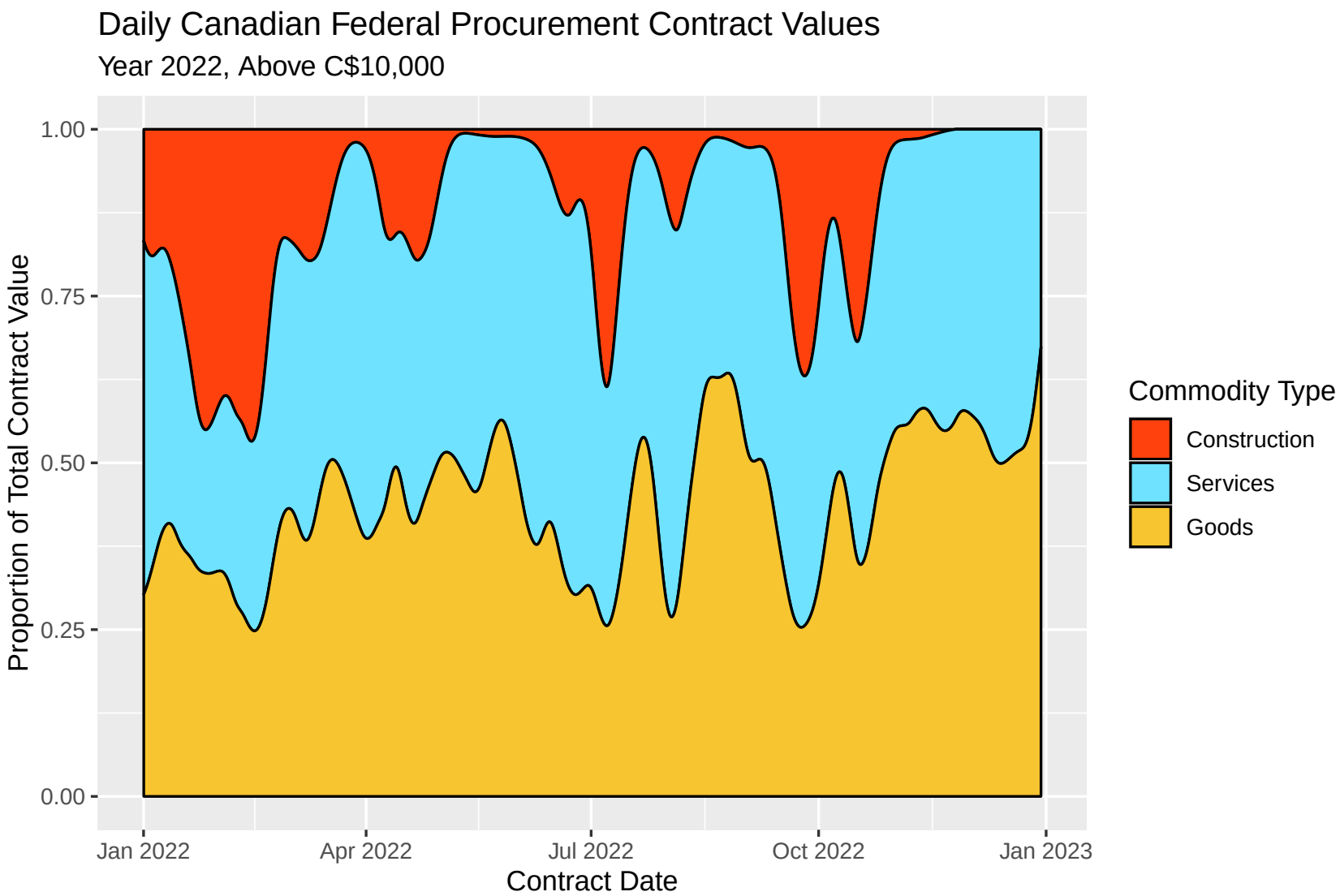
MEMORIAL UNIVERSITY

Area Chart – 1 Variable, 3 Series

```
...

ggplot(aes( contract_date,
            daily_value,
            fill=reorder(commodity_type, -daily_value))) +
 geom_area() +
 geom_stream(type='ridge', bw=0.2) +
 scale_fill_brewer(palette="Spectral") +

...
```

MEMORIAL UNIVERSITY

Area Chart (Stacked, Smoothed) – 1 Variable, 3 Series



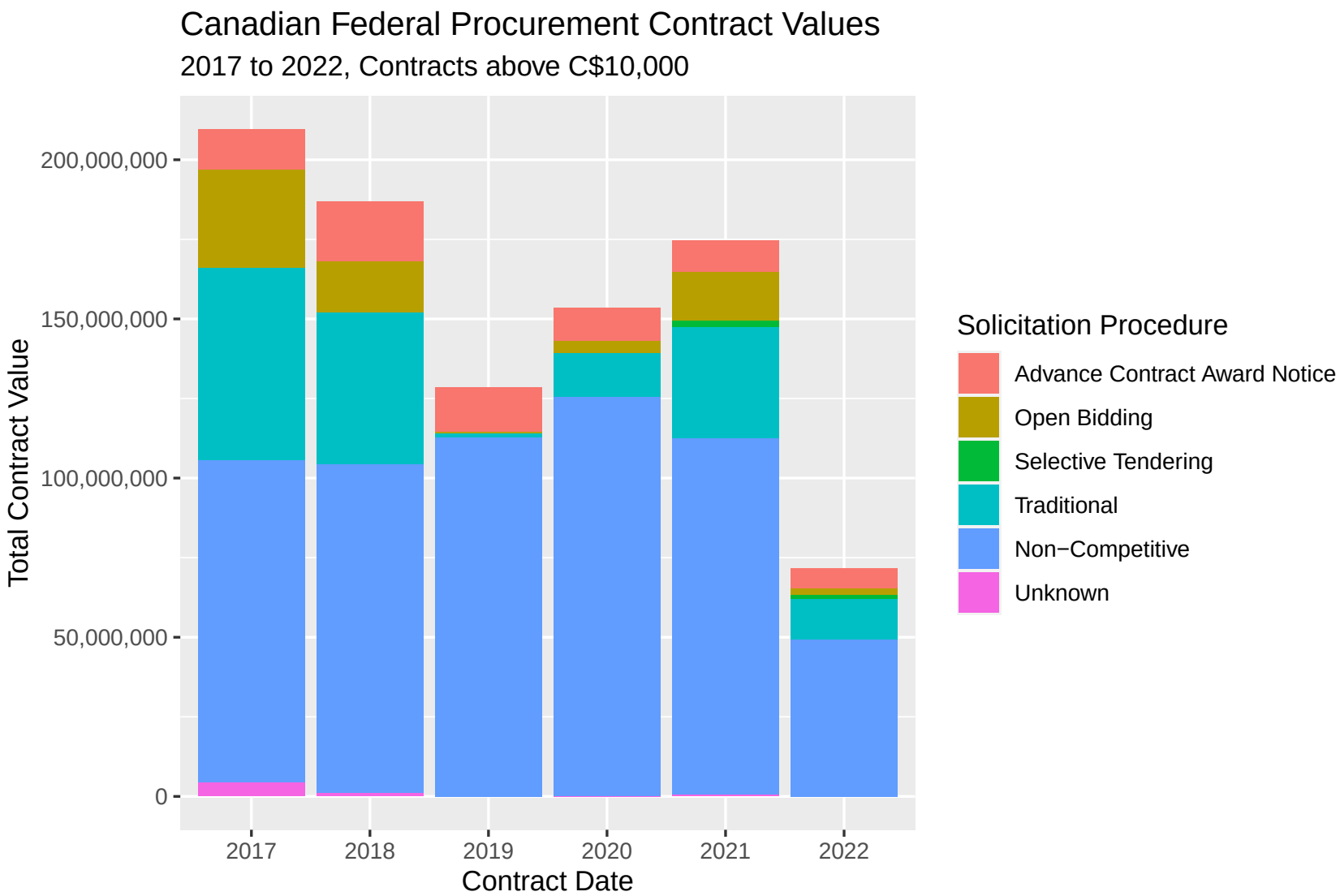
MEMORIAL UNIVERSITY

Area Chart (Stacked, Smoothed) – 1 Variable, 3 Series

```
...

ggplot(aes(
    x=contract_date,
    y=daily_value,
    fill=reorder(commodity_type, daily_value))) +
 geom_area(position="fill", alpha=0.5) +
 geom_stream(type = "proportional", bw=0.5, color='black') +

...
```

MEMORIAL UNIVERSITY

Column Chart – 2 Variables, 3 Series



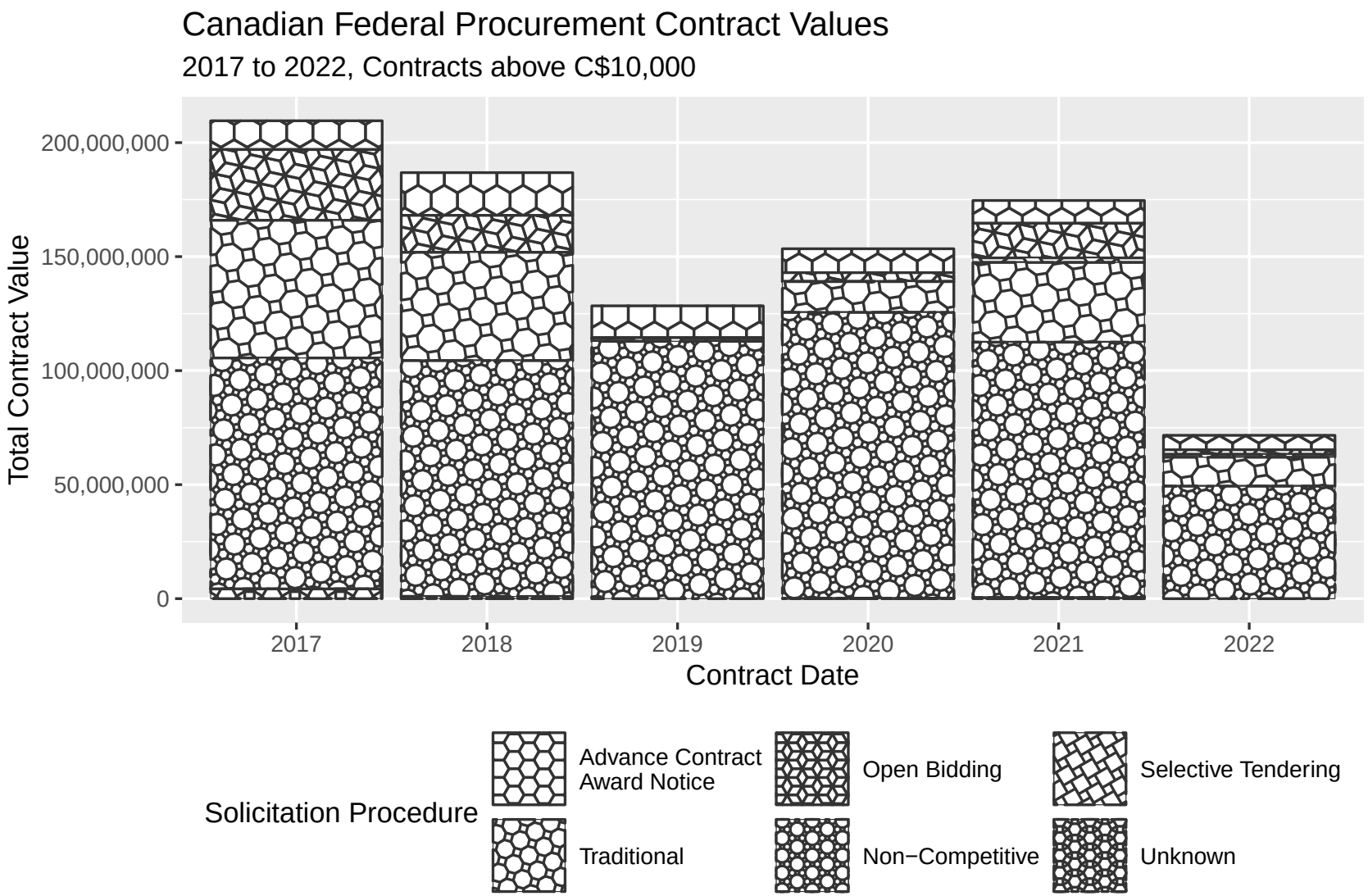
MEMORIAL UNIVERSITY

Column Chart – 2 Variables, 3 Series

```
d.clean %>%
   filter(original_value < quantile(original_value, .95)) %>%
   filter(contract_value < quantile(contract_value, .95)) %>%
   mutate(year=substr(contract_date, 1, 4)) %>%
   filter(contract_date >= '2017-01-01') %>%
   filter(contract_date <= '2022-12-31') %>%
   group_by(solicitation_procedure, year) %>%
   summarize(totalvalue = sum(contract_value)) %>%
   ungroup() %>%
   ggplot(aes(year, totalvalue, fill=solicitation_procedure)) +
      geom_col() +
      scale_fill_brewer(palette="Paired") +
      scale_fill_discrete(
         labels=c("Advance Contract Award Notice", "Open Bidding",
                  "Selective Tendering", "Traditional",
                  "Non-Competitive", "Unknown")) +
      scale_y_continuous(labels=scales::comma) +
      labs(x = 'Contract Date', y = 'Annual Total Contract Value', fil
```

MEMORIAL UNIVERSITY

Column Chart (Pattern) – 2 Variables, 6 Series



MEMORIAL UNIVERSITY

Column Chart (Pattern) – 2 Variables, 6 Series

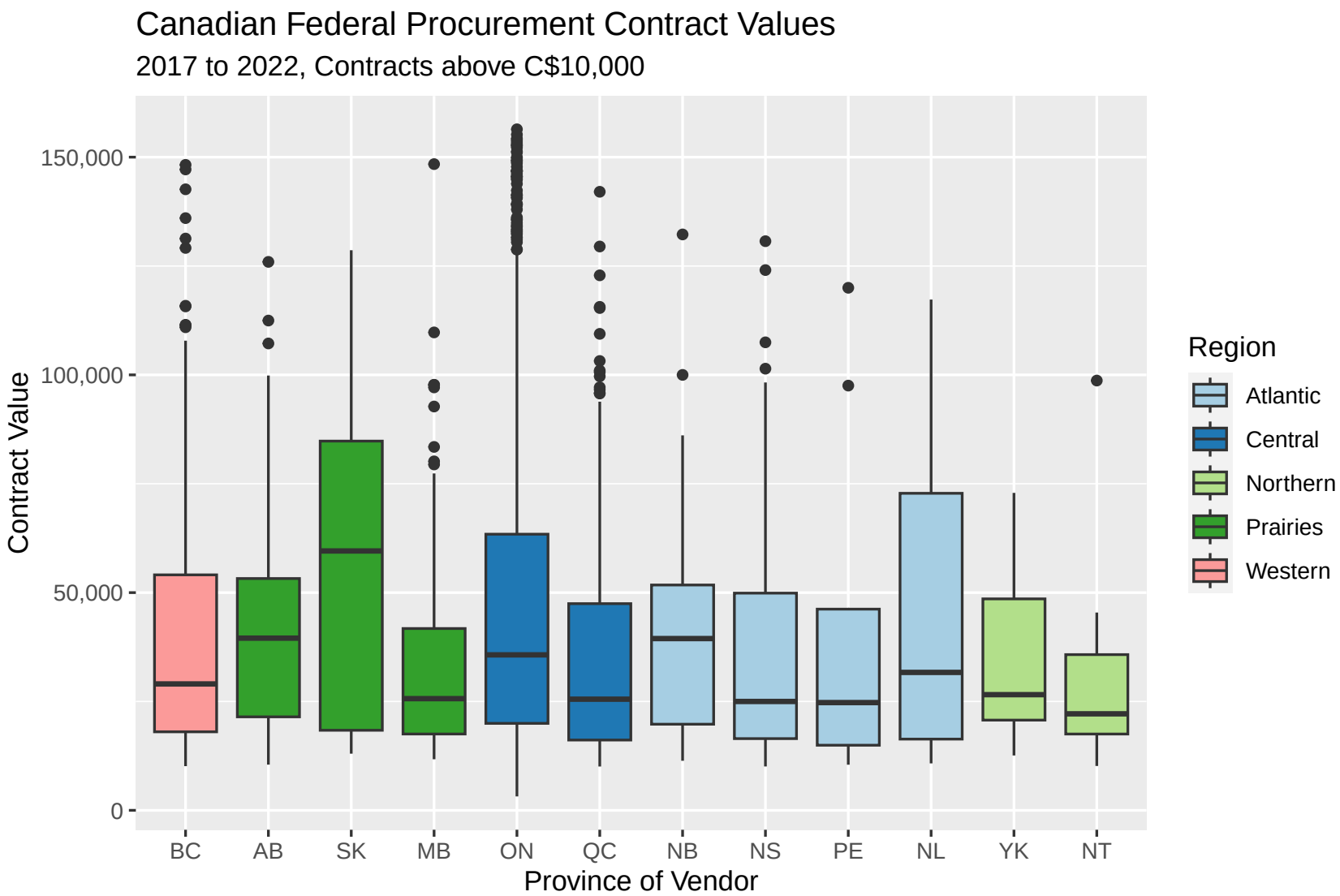
```
ggplot(aes(year, totalvalue, pattern_angle=solicitation_procedure)) +
  geom_col_pattern( aes(pattern_type=solicitation_procedure),
                    pattern_fill='white',
                    pattern='polygon_tiling',
                    pattern_scale=0.5,
                    pattern_key_scale_factor=0.4) +
  scale_pattern_type_manual(
      values = c('hexagonal', 'rhombille', 'pythagorean',
                 'truncated_square', 'rhombitrihexagonal',
                 'truncated_trihexagonal'),
      labels=c("Advance Contract\nAward Notice",
               "Open Bidding", "Selective Tendering",
               "Traditional", "Non-Competitive", "Unknown")) +
  scale_y_continuous(labels=scales::comma) +
  labs(x = 'Contract Date', y = 'Annual Total Contract Value',
       pattern_type='Solicitation Procedure',
       title='Canadian Federal Procurement Contract Values',
       subtitle='2017 to 2022, Contracts above C$10,000') +
```

MEMORIAL UNIVERSITY

Column Chart (Pattern) – 2 Variables, 6 Series

```
guides(pattern_angle=FALSE, pattern_type=guide_legend(nrow=1)) +
guides(pattern_type=guide_legend(
           nrow=2,byrow=TRUE,
           legend.position='bottom',
           keyheight=unit(1, 'cm'),
       keywidth=unit(1, 'cm'))) +
theme(legend.key.size=unit(1, 'cm'), legend.position='bottom')
```

Box Plot – 2 Variables, 5 Series



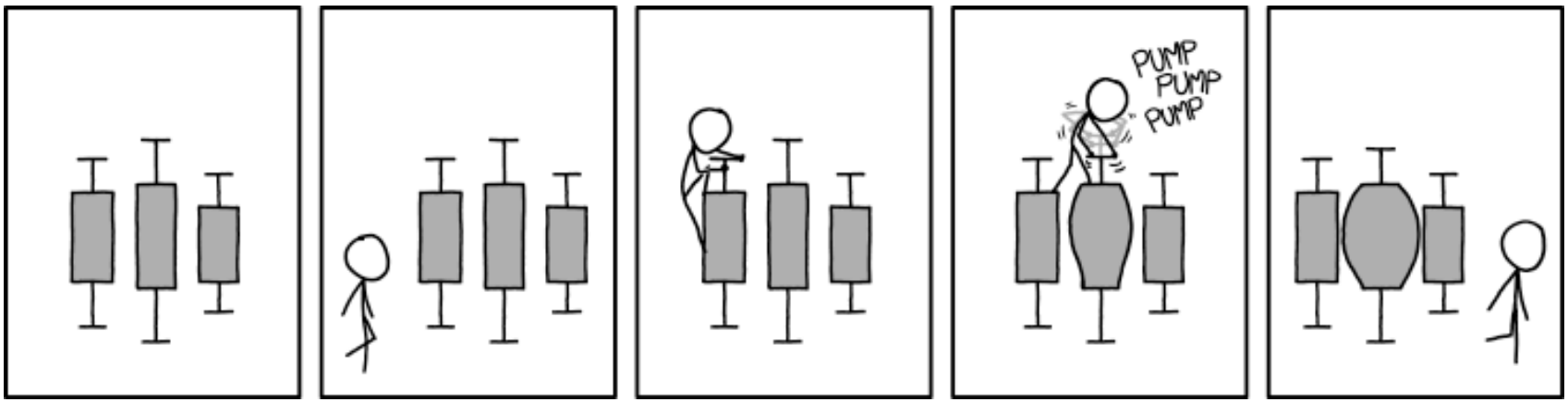
MEMORIAL UNIVERSITY

Box Plot – 2 Variables, 5 Series

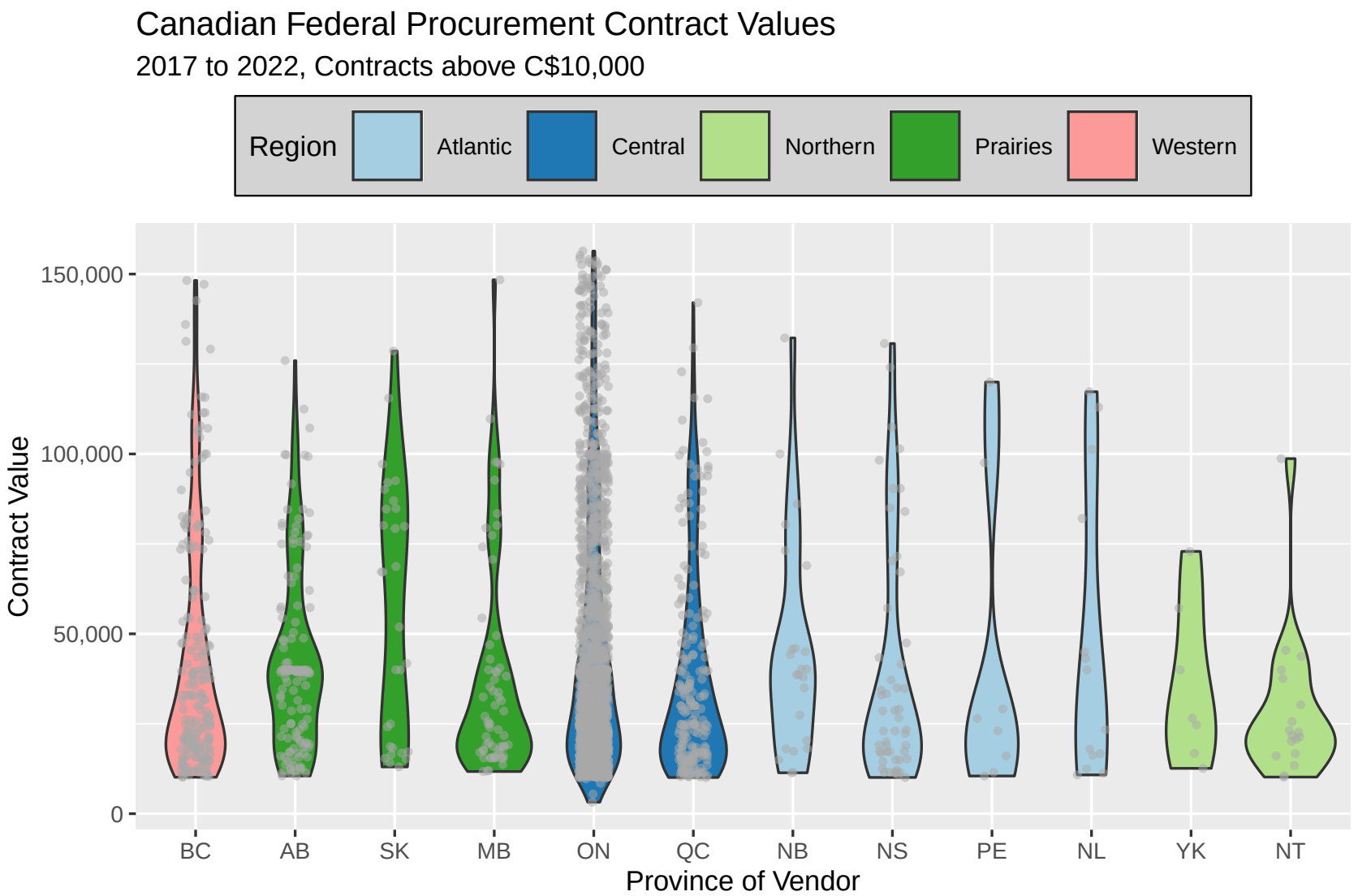
```
d.clean %>%
   mutate(region = case_when(
        vendor_province %in% c('NL', 'NS', 'PE', 'NB') ~ 'Atlantic',
        vendor_province %in% c('YK', 'NT') ~ 'Northern',
        vendor_province %in% c('AB', 'SK', 'MB') ~ 'Prairies',
        vendor_province %in% c('QC', 'ON') ~ 'Central',
        vendor_province %in% c('BC') ~ 'Western')) %>%
   mutate(vendor_province = factor(vendor_province,
        levels=c('BC', 'AB', 'SK', 'MB', 'ON', 'QC', 'NB',
                 'NS', 'PE', 'NL', 'YK', 'NT'), ordered=TRUE)) %>%
   ggplot(aes(vendor_province, contract_value, fill=region)) +
      geom_boxplot() +
      scale_fill_brewer(palette="Paired") +
      scale_y_continuous(labels=scales::comma) +
      labs(x = 'Province of Vendor',
           y = 'Contract Value',
           fill='Region',
           title='Canadian Federal Procurement Contract Values',
           subtitle='2017 to 2022, Contracts above C$10,000')
```

MEMORIAL UNIVERSITY

Boxplot (XKCD)



Violin Plot (with Jitter) – 2 Variables, 5 Series



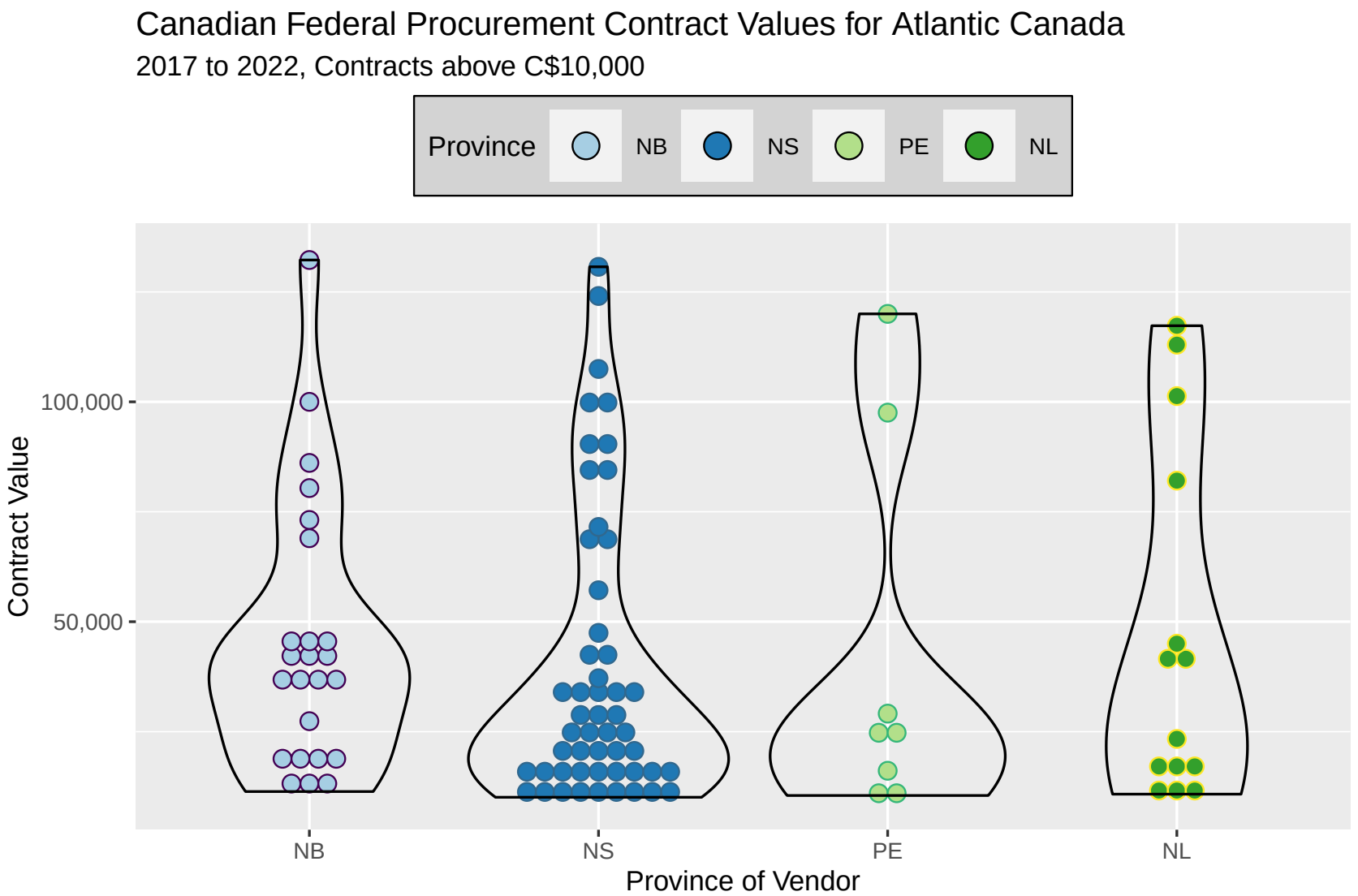
MEMORIAL UNIVERSITY

Violin Plot (with Jitter) – 2 Variables, 5 Series

```
...

ggplot(aes(vendor_province, contract_value, fill=region)) +
  geom_violin() +
  scale_fill_brewer(palette="Paired") +
  scale_y_continuous(labels=scales::comma) +
  labs(x = 'Province of Vendor',
       y = 'Contract Value',
       fill='Region',
       title='Canadian Federal Procurement Contract Values',
       subtitle='2017 to 2022, Contracts above C$10,000') +
  geom_jitter(width=0.15, color='darkgrey',
              size=1, fill=NA, alpha=0.5) +
  theme(legend.position='top',
        legend.background=element_blank(),
        legend.box.background=
            element_rect(color='black', fill='lightgrey'),
            legend.key.size=unit(1, 'cm'))
...
```

MEMORIAL UNIVERSITY

Dot Plot (with Violin) – 2 Variables

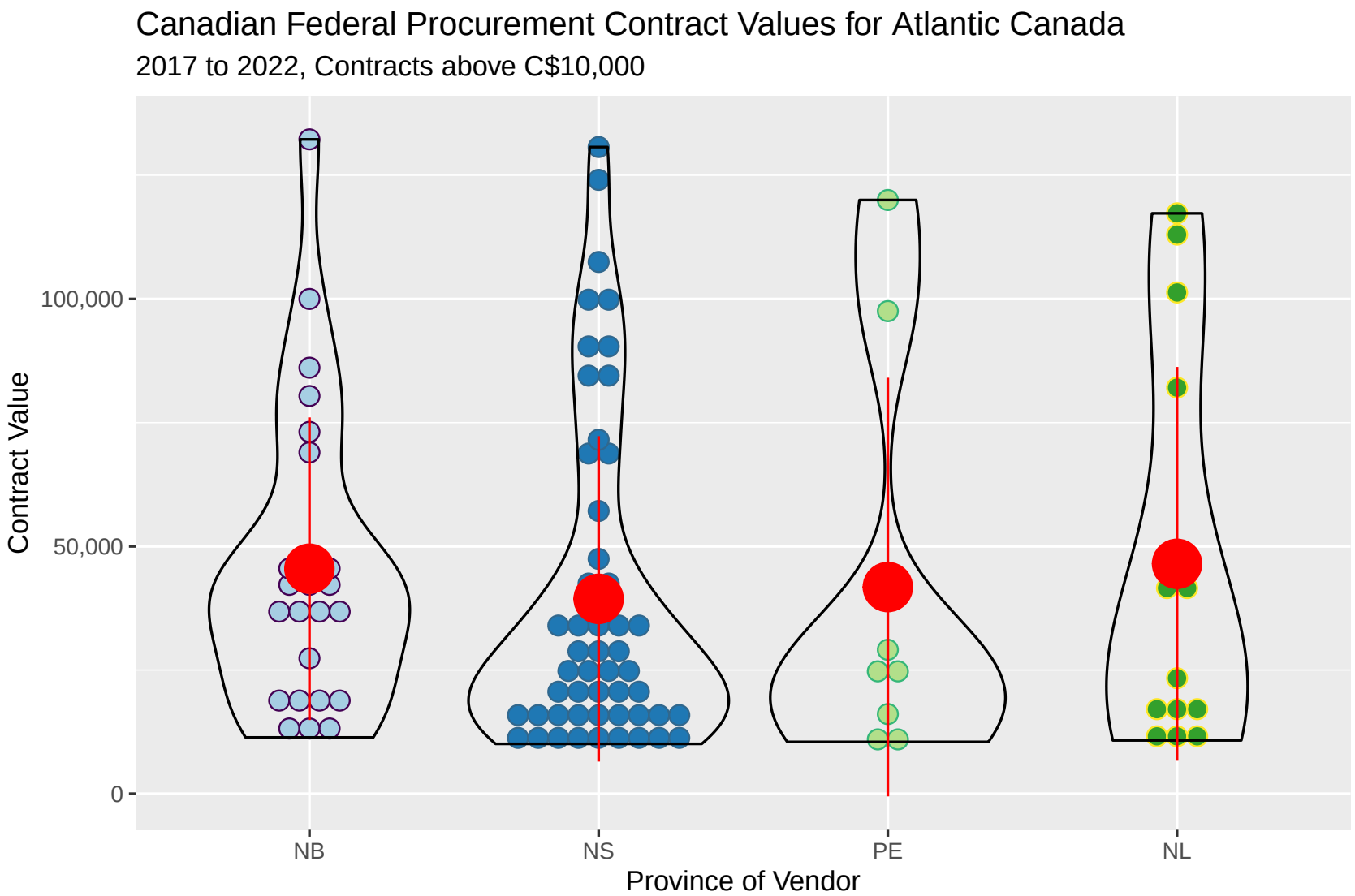


MEMORIAL UNIVERSITY

Box Plot – 2 Variables, 5 Series

```
...

ggplot(aes(x=vendor_province, y=contract_value,
           color=vendor_province, fill=vendor_province)) +
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1) +
  geom_violin(color='black', fill=NA) +
  scale_fill_brewer(palette="Paired")

...
```

Dot Plot (with Summaries) – 2 Variables

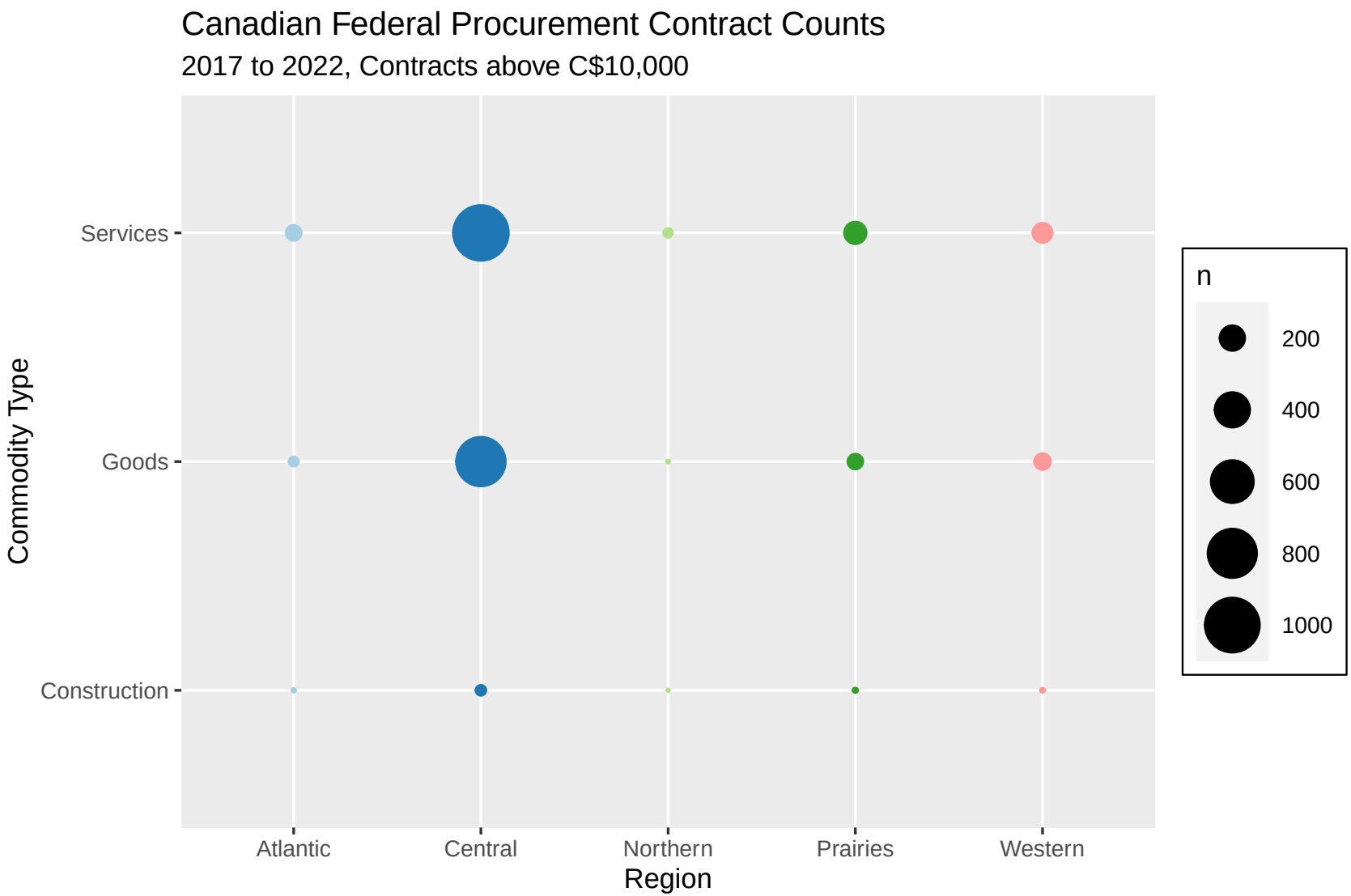


MEMORIAL UNIVERSITY

Dot Plot (with Summaries) – 2 Variables

```
...

ggplot(aes(x=vendor_province, y=contract_value,
           color=vendor_province, fill=vendor_province)) +
  geom_dotplot(binaxis='y', stackdir='center', dotsize=1) +
  geom_violin(color='black', fill=NA) +
  stat_summary(fun.data=mean_sdl,
               fun.args=list(mult=1),
               size=2, color='red',
               geom="pointrange")

...
```

Count Plot – 2 Discrete Variables



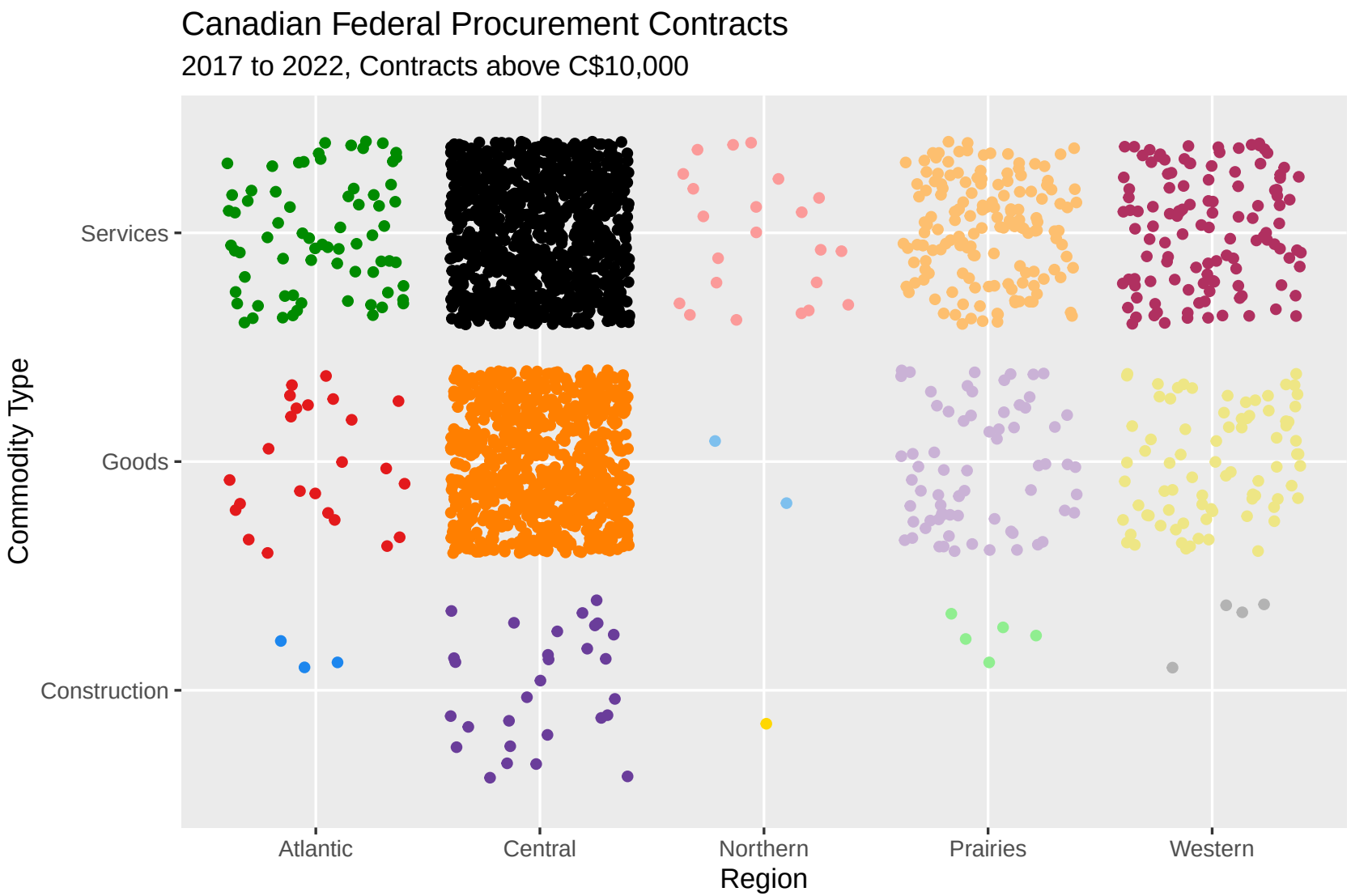
MEMORIAL UNIVERSITY

Count Plot – 2 Discrete Variables

```
...

ggplot(aes(region, commodity_type)) +
  geom_count(aes(color=region))+
  scale_size_area(max_size=10, n.breaks=6) +
  scale_color_brewer(palette="Paired") +
  scale_y_discrete(labels=c('Construction', 'Goods', 'Services')) +
  guides(color=FALSE) +
  labs(x = 'Region', y = 'Commodity Type'
       title='Canadian Federal Procurement Contract Counts',
       subtitle='2017 to 2022, Contracts above C$10,000') +
 theme(legend.background=element_blank(),
       legend.box.background=element_rect(color='black', fill=NA),
       legend.key.size=unit(1, 'cm'))

...
```

MEMORIAL UNIVERSITY

Jitter Plot – 2 Discrete Variables



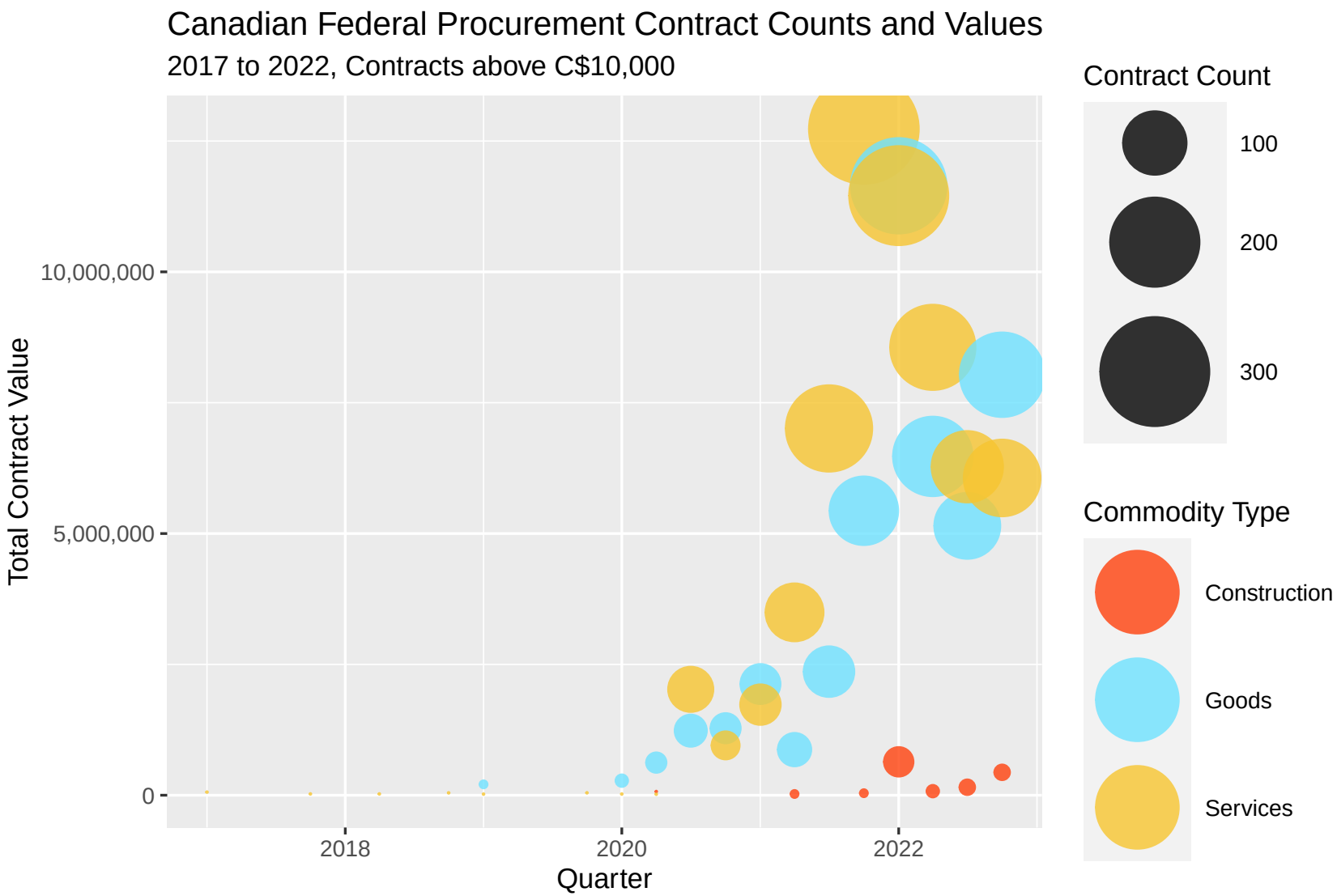
MEMORIAL UNIVERSITY

Jitter Plot – 2 Discrete Variables

```
...

ggplot(aes(region, commodity_type)) +
  geom_jitter(aes(color=paste(region, commodity_type))) +
  scale_color_manual(values=c25) +
  scale_y_discrete(labels=c('Construction', 'Goods', 'Services')) +
  guides(color=FALSE) +
  labs(x = 'Region', y = 'Commodity Type',
       title='Canadian Federal Procurement Contracts',
       subtitle='2017 to 2022, Contracts above C$10,000')

...
```

MEMORIAL UNIVERSITY

Point Plot – Multiple Variables



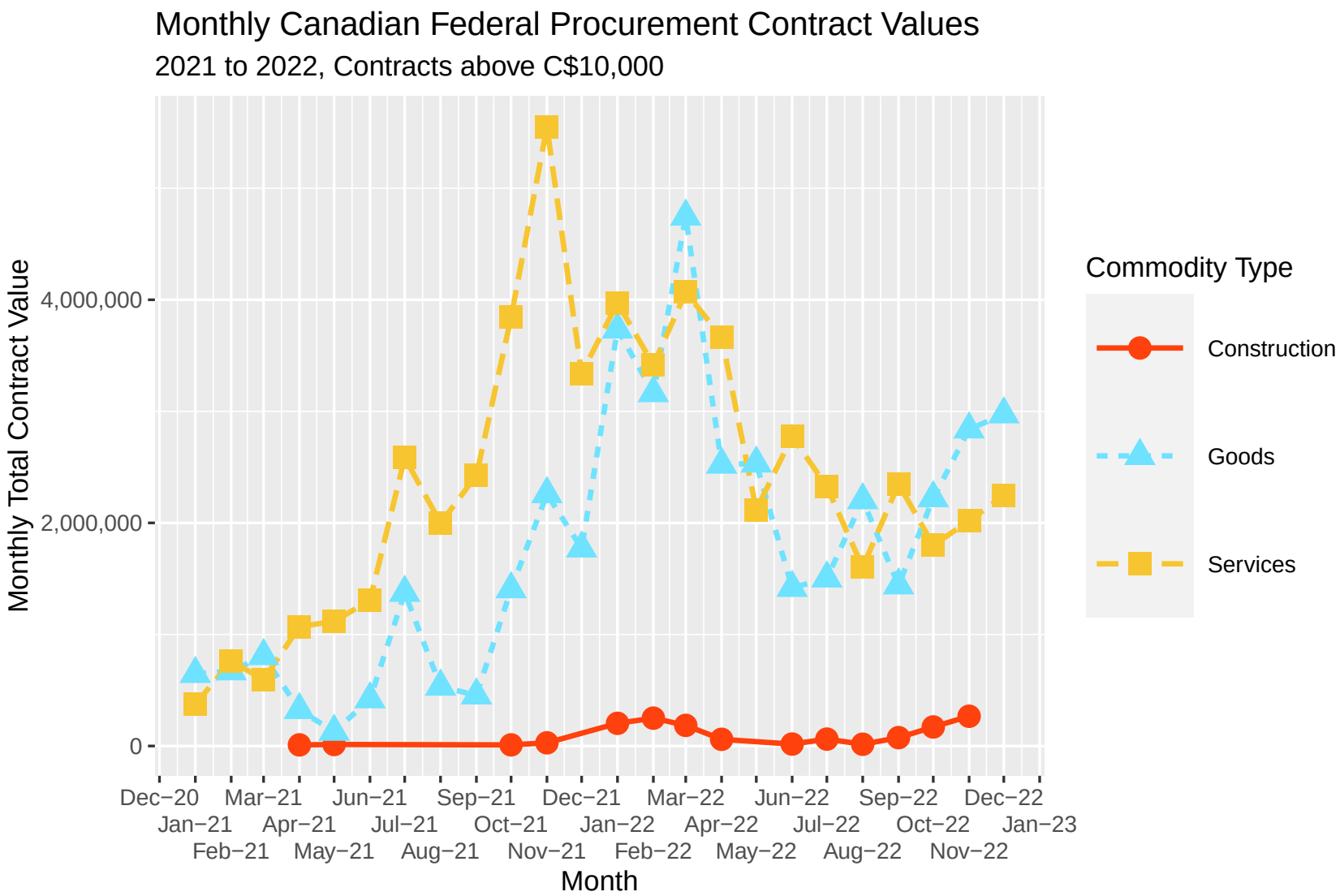
MEMORIAL UNIVERSITY

Point Plot – Multiple Variables

```
d.clean %>% ... %>%
   mutate(contract_quarter = floor_date(contract_date, 'quarter')) %>%
   group_by(contract_quarter, commodity_type) %>%
   summarize(totalcount = n(), totalvalue=sum(contract_value)) %>%
   ungroup() %>%
   ggplot(aes(contract_quarter, totalvalue,
              color=commodity_type, size=totalcount)) +
      geom_point(alpha=0.8) +
      scale_size_continuous(range=c(0, 20)) +
      scale_color_tron(labels=c('Construction', 'Goods', 'Services')) +
      scale_y_continuous(labels=scales::comma) +
          guides(color=guide_legend(override.aes = list(size=15))) +
      labs(x = 'Quarter',
           y = 'Quarterly Total Contract Value',
           color='Commodity Type',
           size='Contract Count',
           title='Quraterly Canadian Federal Procurement Contract Coun
           subtitle='2017 to 2022, Contracts above C$10,000')
```

MEMORIAL UNIVERSITY

Line Plot – 2 Variables, 3 Series



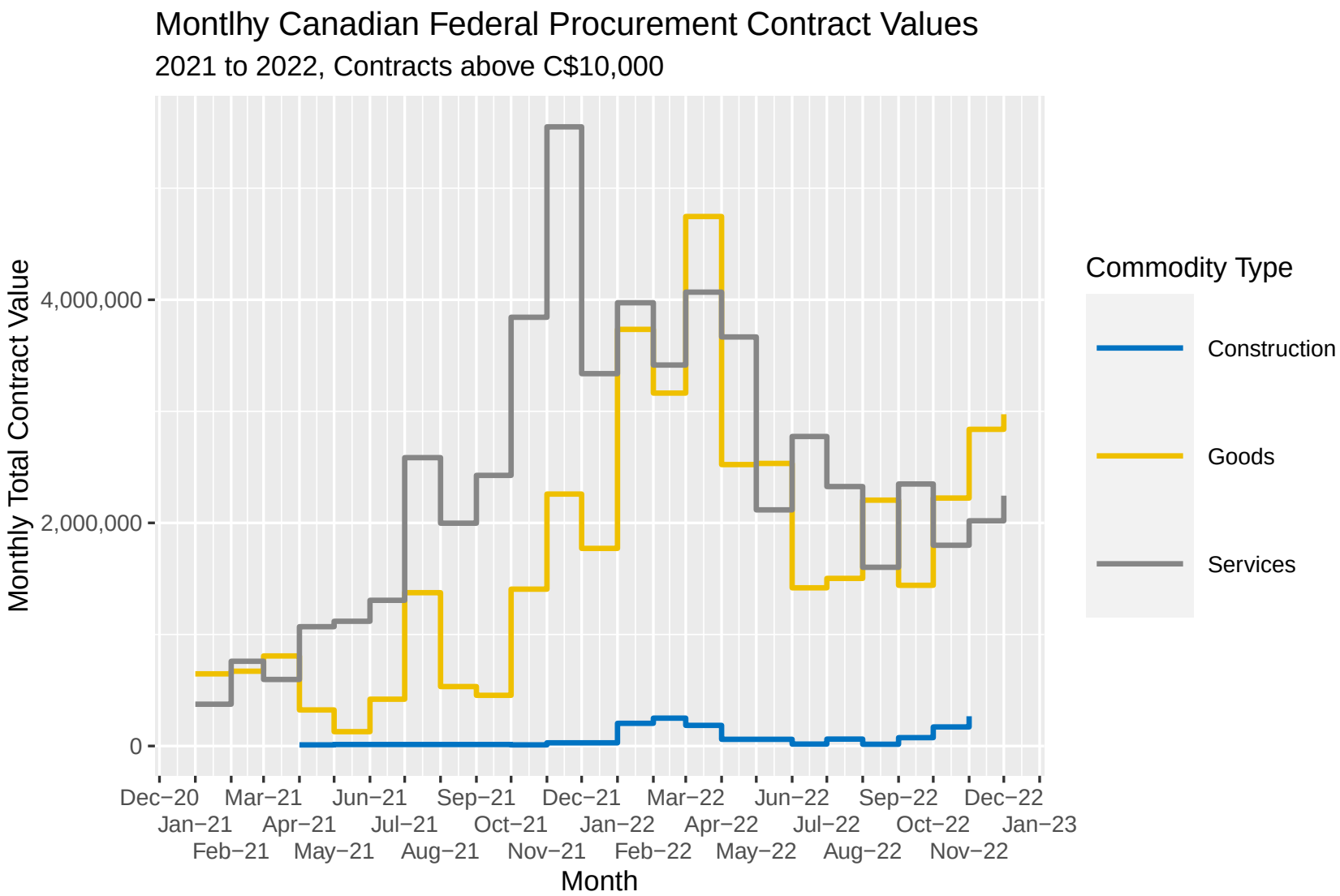
MEMORIAL UNIVERSITY

Line Plot – 2 Variables, 3 Series

```
...
ggplot(aes(contract_quarter, totalvalue,
           color=commodity_type,
           shape=commodity_type,
           linetype=commodity_type)) +
  geom_line(size=1) +
  geom_point(size=4) +
  scale_color_tron(labels=c('Construction', 'Goods', 'Services')) +
  scale_linetype(labels=c('Construction', 'Goods', 'Services')) +
  scale_shape(labels=c('Construction', 'Goods', 'Services')) +
  scale_y_continuous(labels=scales::comma) +
  scale_x_date(date_breaks = "months" ,
               date_labels = "%b-%y",
               guide = guide_axis(n.dodge=3)) +
  labs(x = 'Month',
       y = 'Monthly Total Contract Value',
       color='Commodity Type',
       shape='Commodity Type',
       linetype='Commodity Type',
       title='Monthly Canadian Federal Procurement Contract Values',
       subtitle='2021 to 2022, Contracts above C$10,000') +
  theme(legend.key.size=unit(1.5, 'cm'))

...
```

MEMORIAL UNIVERSITY

Step Plot – 2 Variables, 3 Series



MEMORIAL UNIVERSITY

Step Plot – 2 Variables, 3 Series

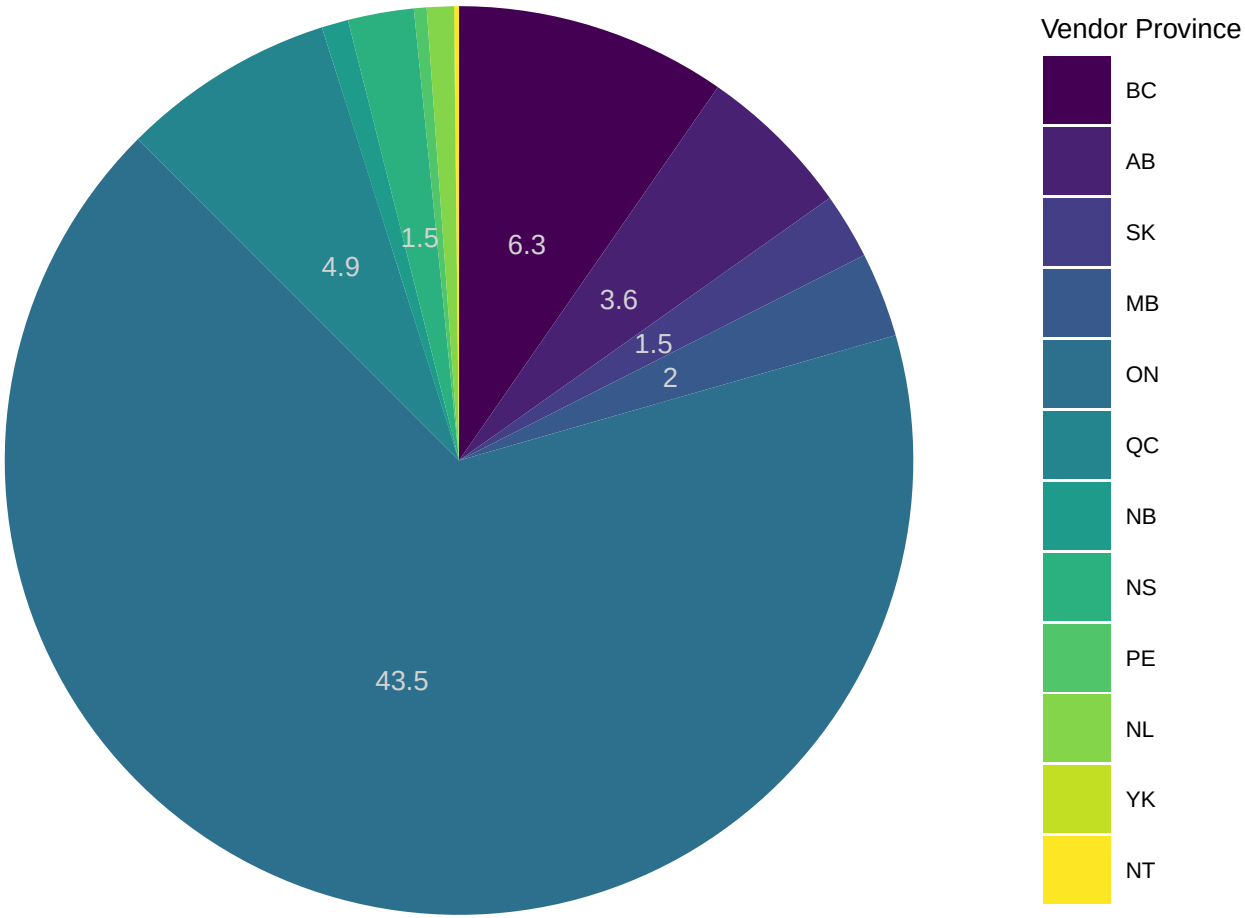
```
...

ggplot(aes(contract_quarter, totalvalue, color=commodity_type)) +
  geom_step(size=1) +
  scale_color_jco(labels=c('Construction', 'Goods', 'Services')) +
  scale_linetype(labels=c('Construction', 'Goods', 'Services')) +
  scale_y_continuous(labels=scales::comma) +
  scale_x_date(date_breaks = "months" ,
               date_labels = "%b-%y",
               guide = guide_axis(n.dodge=3)) +
  labs(x = 'Month',
       y = 'Monthly Total Contract Value',
       color='Commodity Type',
       title='Montlhy Canadian Federal Procurement Contract Values',
       subtitle='2021 to 2022, Contracts above C$10,000') +
  theme(legend.key.size=unit(1.5, 'cm'))

...
```

MEMORIAL UNIVERSITY

Pie Chart – 1 Variable

Canadian Federal Procurement Contract Values

2022, Millions of C$, Contracts above C$10,000



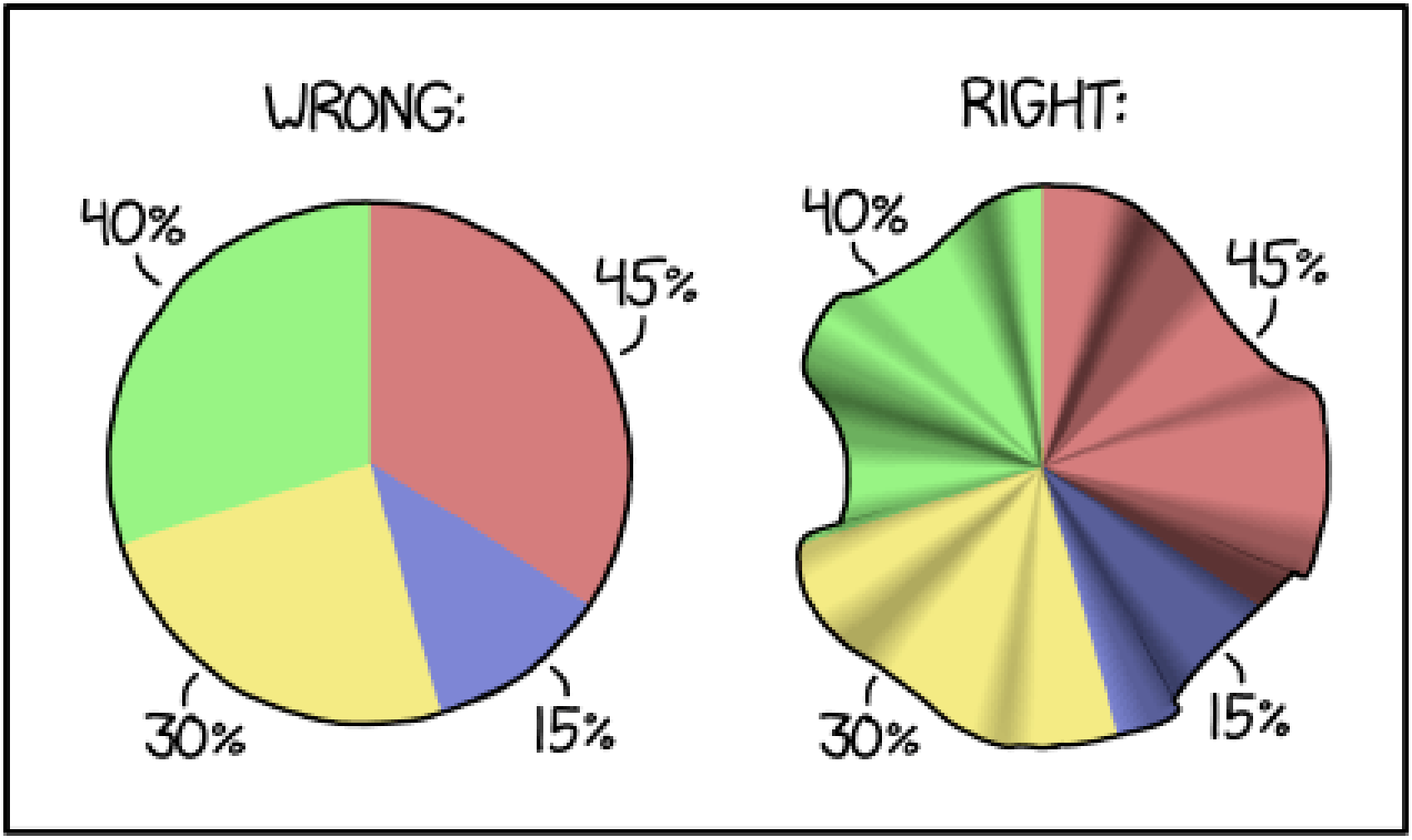
MEMORIAL UNIVERSITY

Pie Chart – 1 Variable

```
d.clean %>% .... %>%
   group_by(vendor_province) %>%
   summarize(totalvalue=sum(contract_value)) %>%
   ungroup() %>%
   ggplot(aes(x='', y=totalvalue, fill=vendor_province)) +
     geom_bar(stat='identity', width=1) +
     coord_polar('y', direction=-1, start=0) +
     geom_text(aes(label=ifelse(totalvalue > 1000000,
                                round(totalvalue/1000000, digits=1),
                                '')),
                   color='lightgrey',
                   position = position_stack(vjust=0.5)) +
      scale_y_continuous(labels=NULL) +
      scale_color_brewer(palette="Paired") +
      labs(x = '', y = '',
           fill='Vendor Province',
           title='Canadian Federal Procurement Contract Values',
           subtitle='2022, Millions of C$, Contracts above C$10,000')
      theme(legend.key.size=unit(1, 'cm')) +
      theme_void()
```

MEMORIAL UNIVERSITY

Pie Charts Done Well (XKCD)

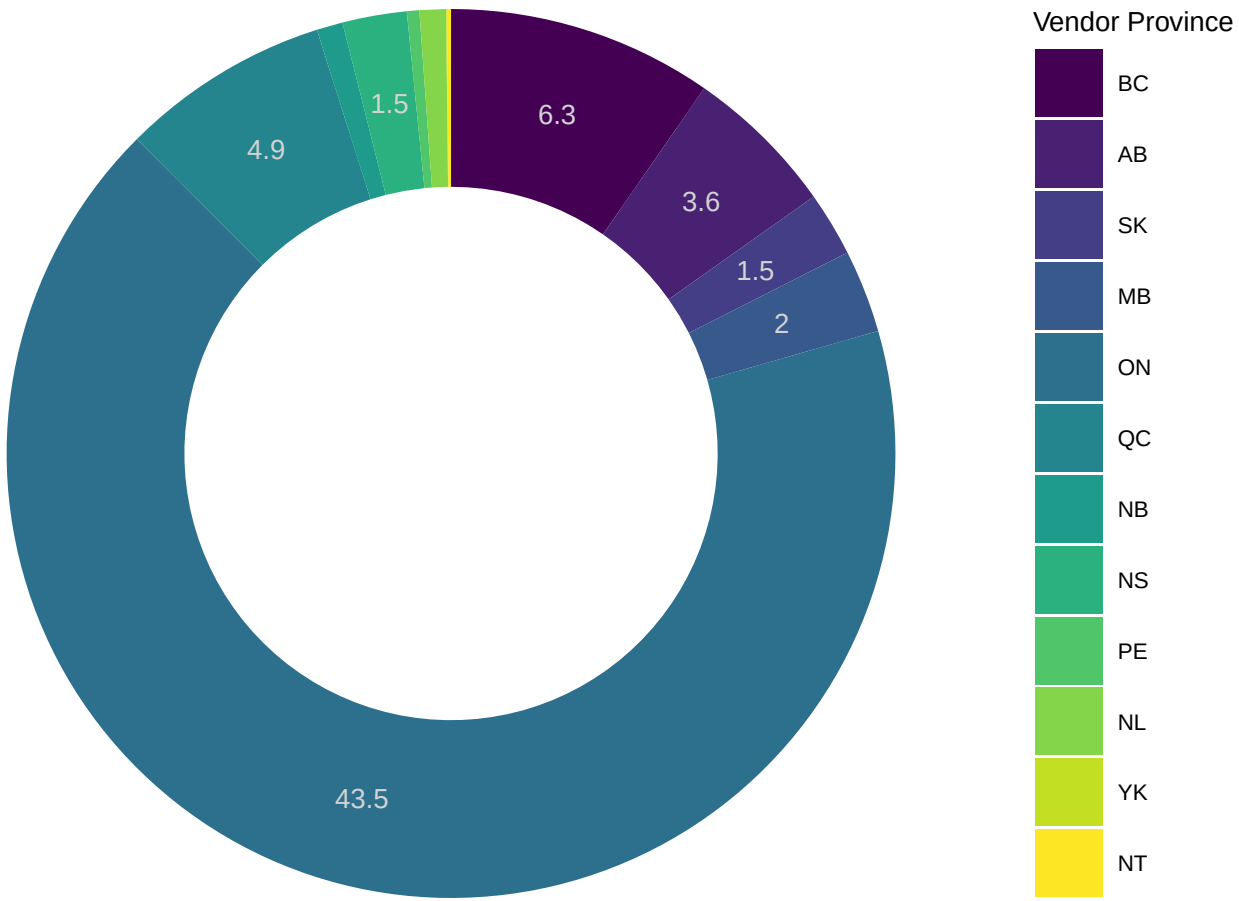


HOW TO MAKE A PIE CHART IF YOUR PERCENTAGES DON'T ADD UP TO 100

Donut Chart – 1 Variable

Canadian Federal Procurement Contract Values

2022, Millions of C$, Contracts above C$10,000



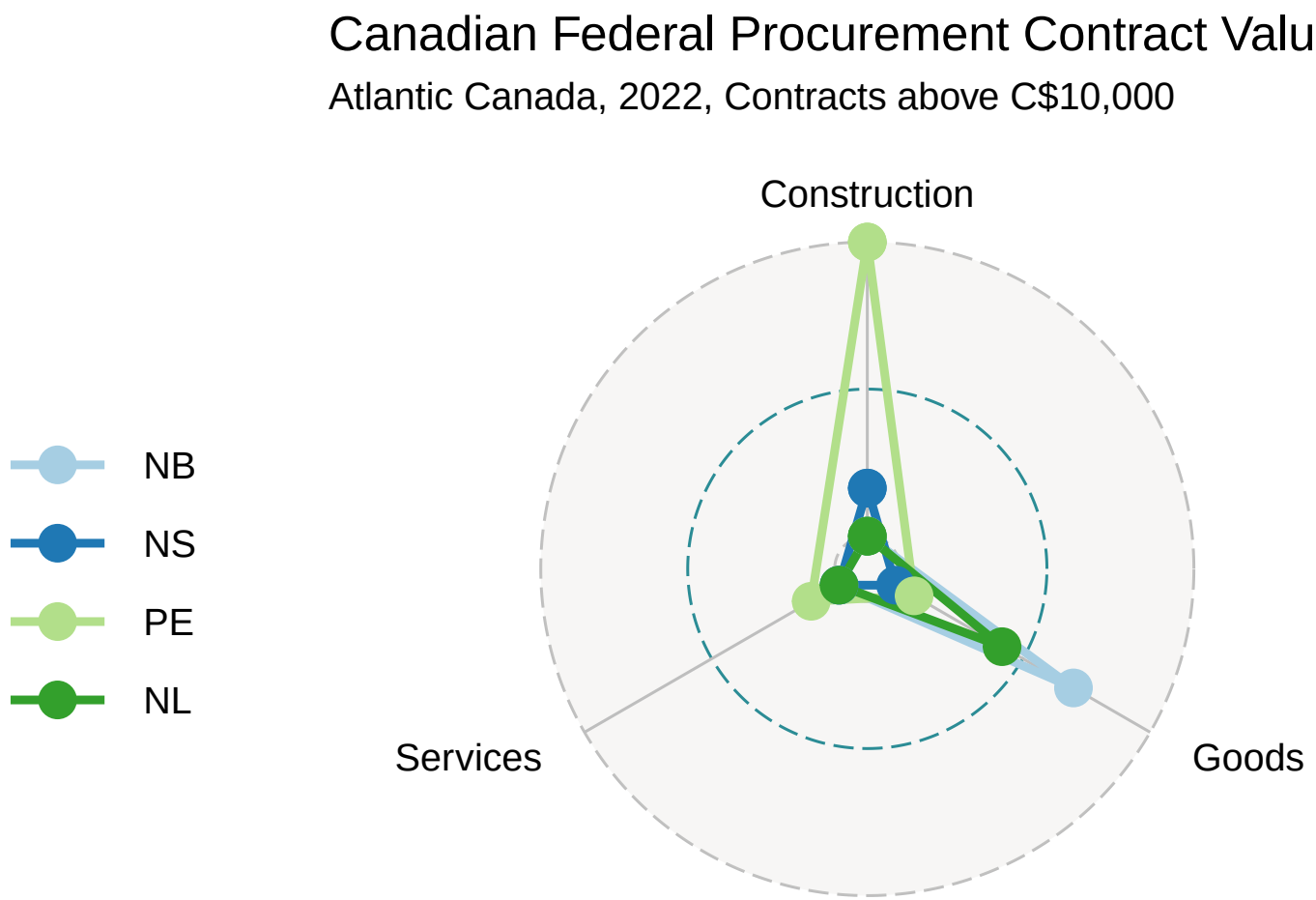
MEMORIAL UNIVERSITY

Donut Chart – 1 Variable

```
holesize <- 2

...
ggplot(aes(x=holesize, y=totalvalue, fill=vendor_province)) +
  geom_col() +
  coord_polar('y', direction=-1, start=0) +
  xlim(c(0.2, holesize+0.5)) +
  geom_text(aes(label=ifelse(totalvalue > 1000000,
                             round(totalvalue/1000000, digits=1),
                             '')),
                color='lightgrey',
                position = position_stack(vjust=0.5)) +

...
```

MEMORIAL UNIVERSITY

Radar Plot – 3 Variables



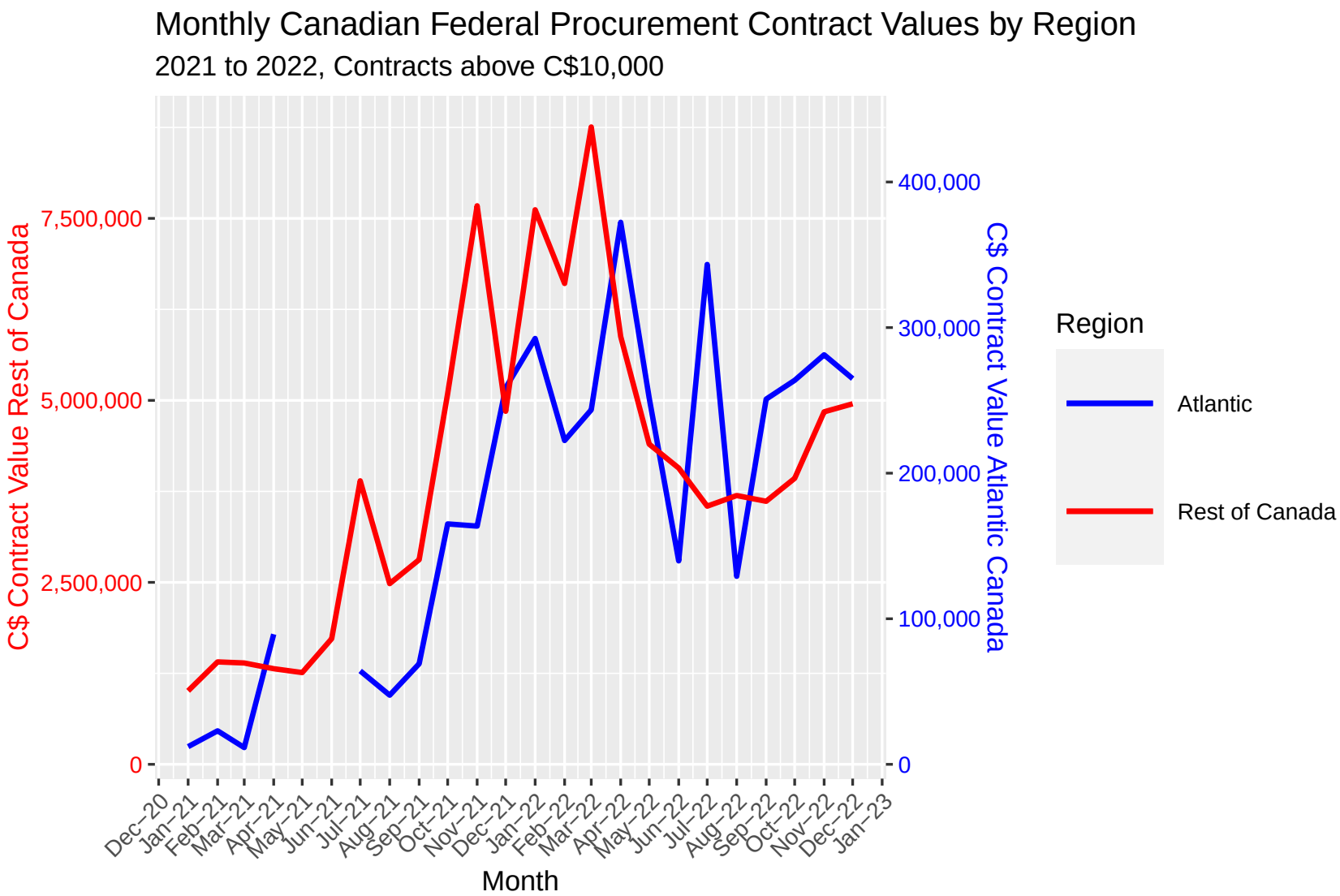
MEMORIAL UNIVERSITY

Radar Plot – 3 Variables

```
d.clean %>%  ... %>%
   filter(vendor_province %in% c('NL', 'PE', 'NB', 'NS')) %>%
   group_by(vendor_province, commodity_type) %>%
   summarize(totalvalue=sum(contract_value)/1000000) %>%
   ungroup() %>%
   pivot_wider(names_from='commodity_type', values_from=totalvalue) %>
   replace_na(list(C=0, G=0, S=0)) %>%
   select(vendor_province, 'C', 'G', 'S') %>%
   mutate_at(vars(-vendor_province), rescale) %>%
   relocate(vendor_province, .before=1) %>%
   ggradar(axis.labels=c('Construction', 'Goods', 'Services'),
           values.radar='') +
      scale_color_brewer(palette="Paired") +
      labs(x = '', y = '',  fill='Vendor Province',
           title='Canadian Federal Procurement Contract Values',
           subtitle='Atlantic Canada, 2022, Contracts above C$10,000')
      theme(plot.title=element_text(size=18),
            plot.subtitle=element_text(size=14))
```

MEMORIAL UNIVERSITY

Multiple Axes – 2 Variables, 3 Series



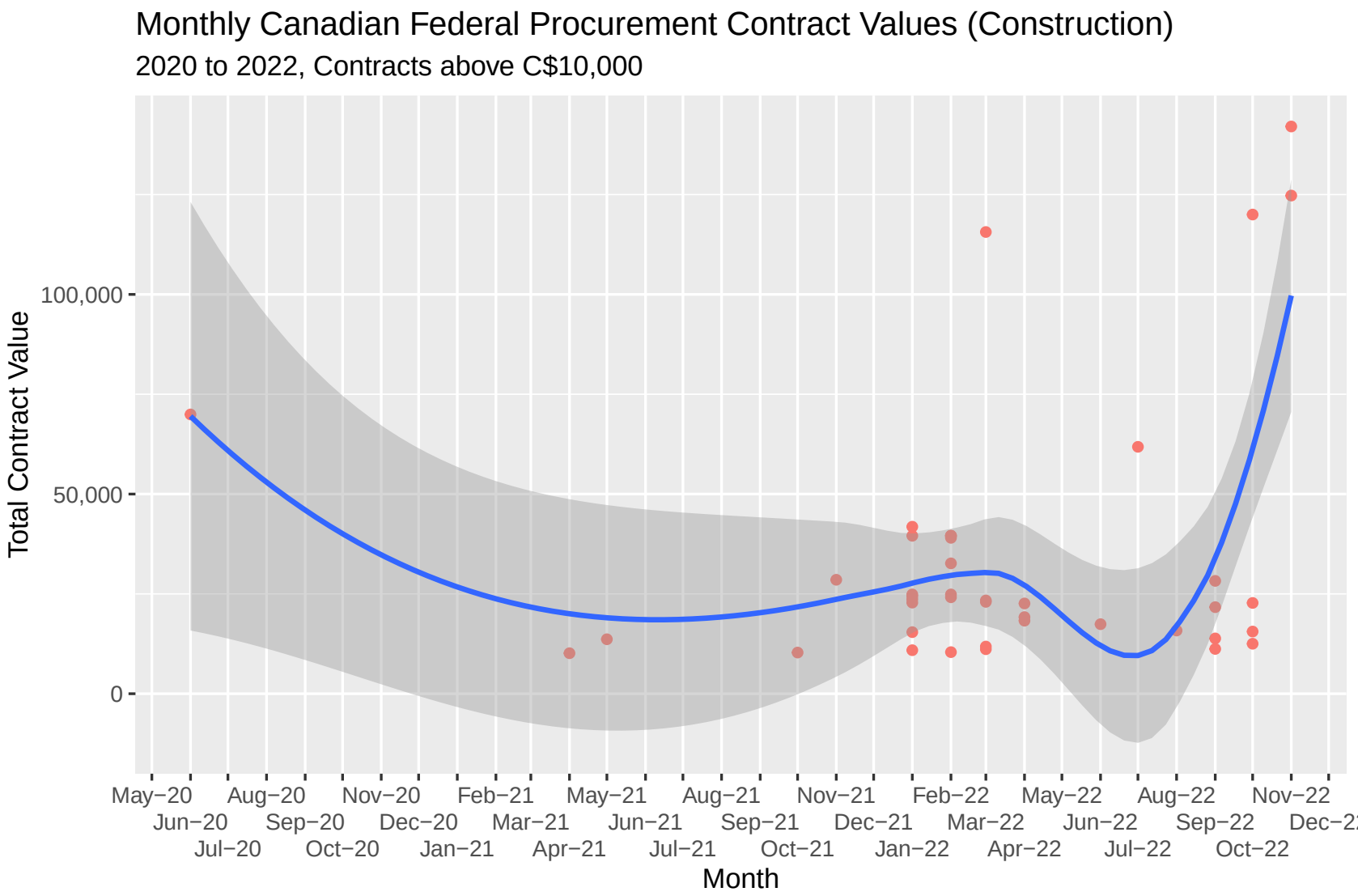
MEMORIAL UNIVERSITY

Multiple Axes – 2 Variables, 3 Series

```
ggplot(aes(x=contract_quarter)) +
  scale_color_manual(
      name='Region',
      values=c('Rest of Canada' = 'red', 'Atlantic' = 'blue')) +
  geom_line(aes(y=Atlantic2, color='Atlantic'), size=1) +
  geom_line(aes(y=Rest, color='Rest of Canada'), size=1) +
  scale_y_continuous(
      labels=scales::comma,
      name="C$ Contract Value Rest of Canada",
      sec.axis=sec_axis(~./20,
                        labels=scales::comma,
                        name="C$ Contract Value Atlantic Canada")) +
  scale_x_date(date_breaks = "months" , date_labels = "%b-%y") +
  labs(x = 'Month', y = 'Total Contract Value', color='Region',
       title='Monthly Canadian Federal Procurement Contract Values by
       subtitle='2021 to 2022, Contracts above C$10,000') +
  theme(legend.key.size=unit(1.5, 'cm'),
        axis.text.x = element_text(angle=45, hjust=1),
        axis.text.y.left = element_text(color='red'),
        axis.text.y.right = element_text(color='blue'),
        axis.title.y.left = element_text(color='red'),
        axis.title.y.right=element_text(color='blue'))
```

MEMORIAL UNIVERSITY

Smoothing (Local Regression) – 2 Variables



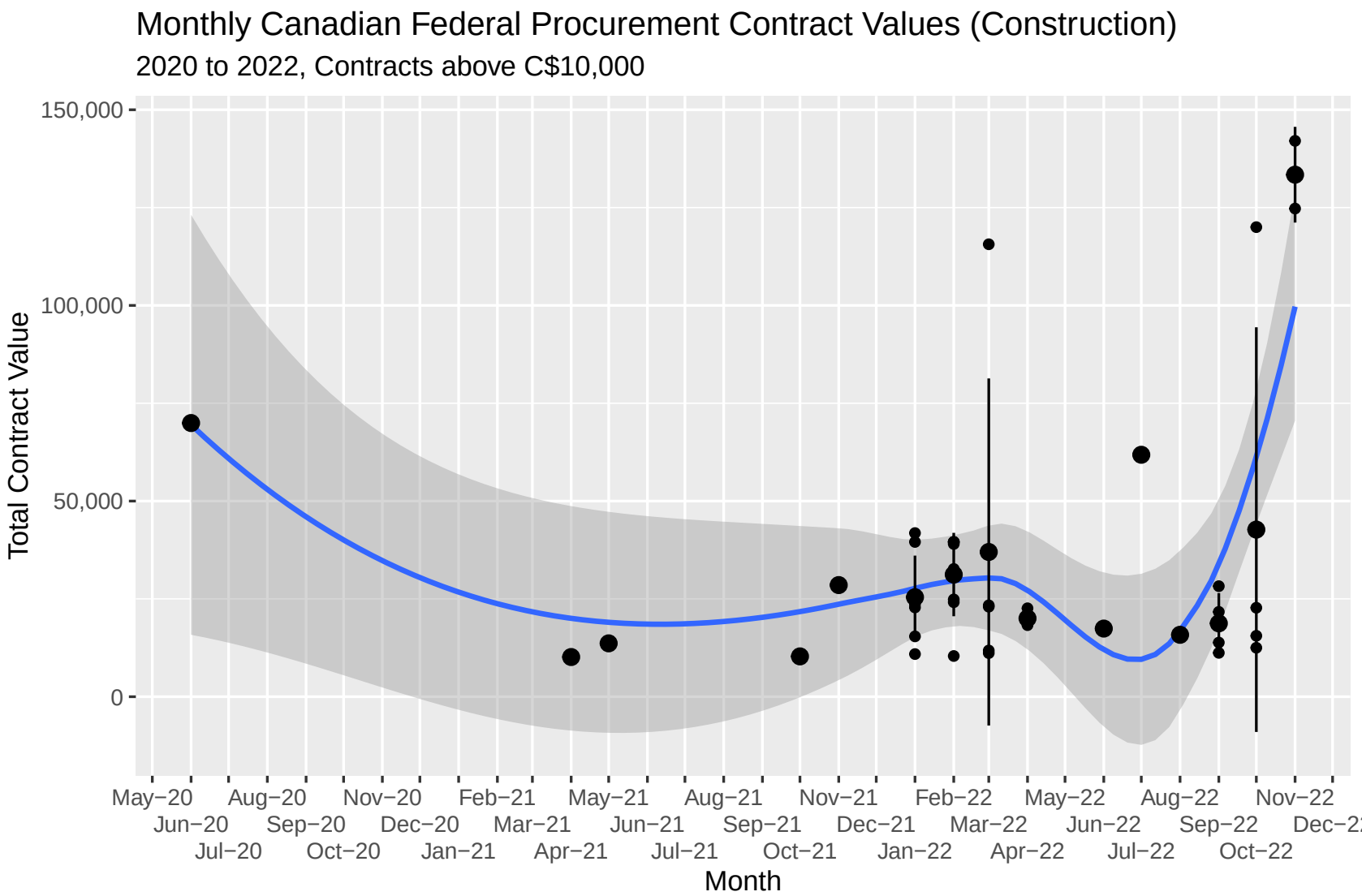
MEMORIAL UNIVERSITY

Smoothing (Local Regression) – 2 Variables

```
d.clean %>% ... %>%
   mutate(contract_week = floor_date(contract_date, 'month')) %>%
   filter(commodity_type=='C') %>%
   ggplot(aes(contract_week, contract_value)) +
     geom_point(aes(color=commodity_type)) +
     geom_smooth() +
     guides(color=FALSE) +
       scale_y_continuous(labels=scales::comma) +
       scale_x_date(date_breaks = "months" ,
                    minor_breaks=NULL,
                    date_labels = "%b-%y",
                    guide = guide_axis(n.dodge=3)) +
       labs(x = 'Month', y = 'Total Contract Value',
           title='Monthly Canadian Federal Procurement Contract Values
           subtitle='2020 to 2022, Contracts above C$10,000')
```

MEMORIAL UNIVERSITY

Smoothing (with Ranges) – 2 Variables

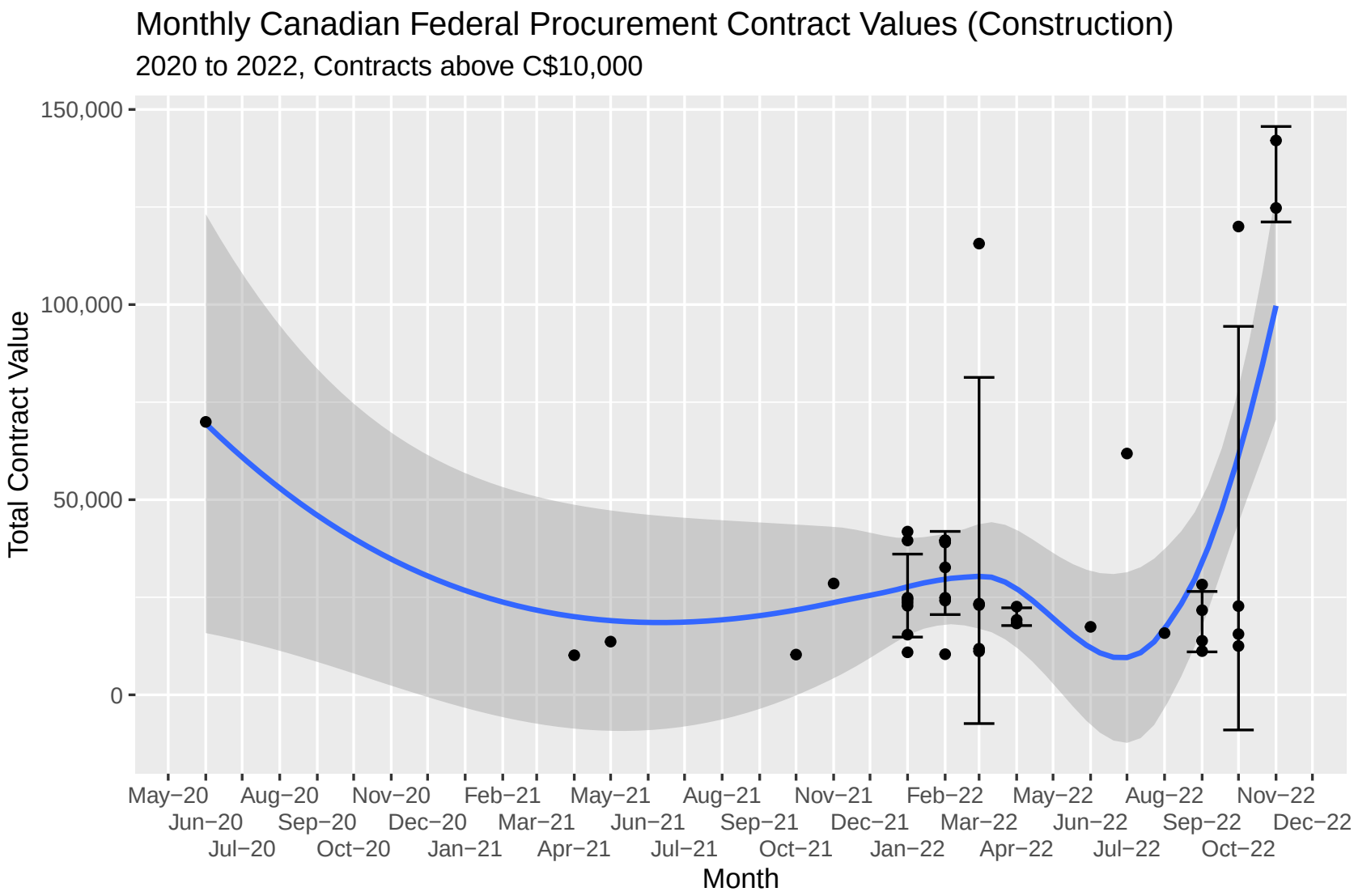


MEMORIAL UNIVERSITY

Smoothing (with Ranges) – 2 Variables

```
...

ggplot(aes(contract_week, contract_value)) +
  geom_smooth() +
  geom_point() +
  stat_summary(fun.data=mean_sdl,
               fun.args=list(mult=1),
               geom="pointrange") +

...
```

Smoothing (with Error Bars) – 2 Variables

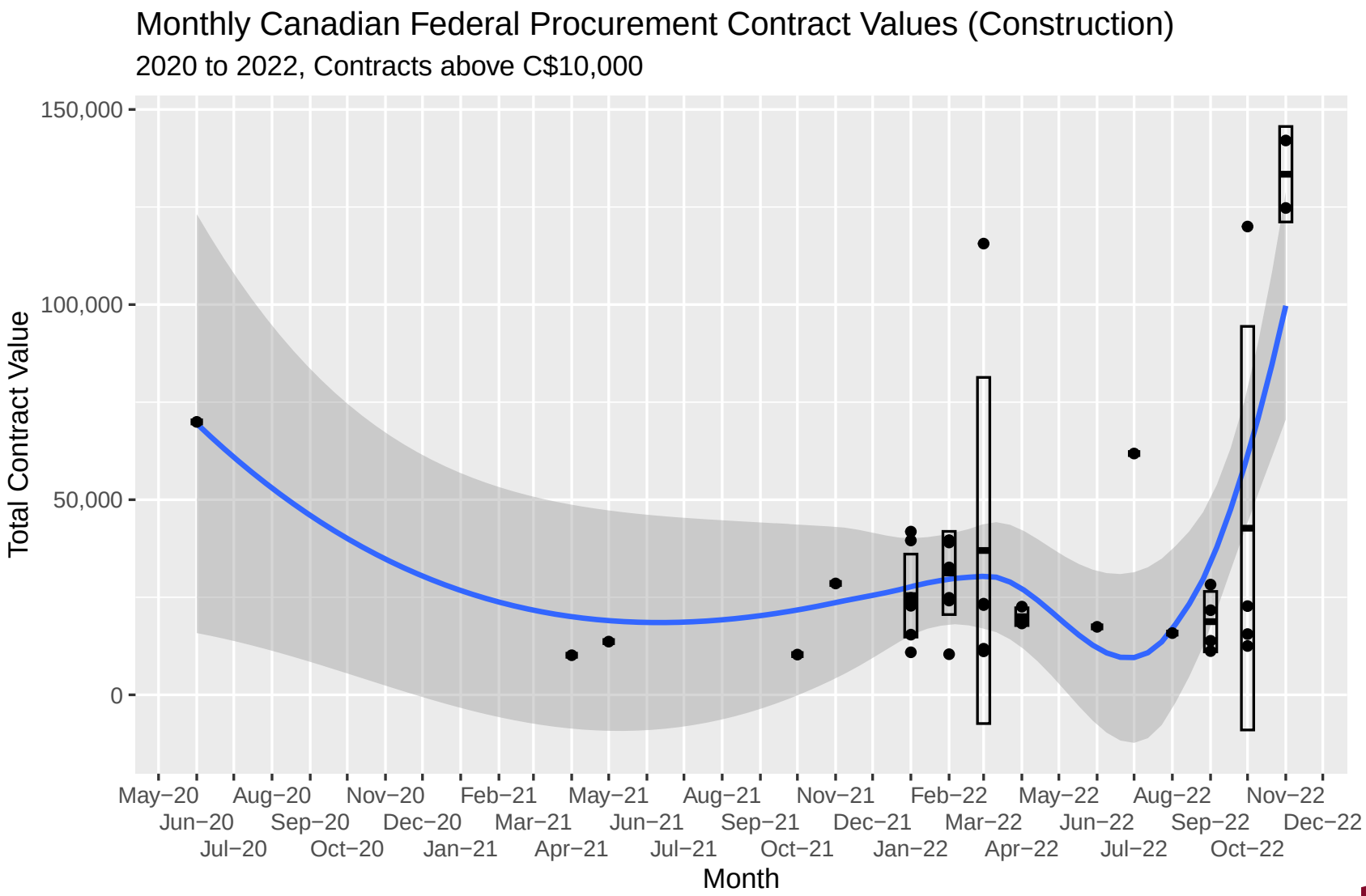


MEMORIAL UNIVERSITY

Smoothing (with Error Bars) – 2 Variables

```
...

ggplot(aes(contract_week, contract_value)) +
  geom_smooth() +
  geom_point() +
  stat_summary(fun.data=mean_sdl,
               fun.args=list(mult=1),
               geom="errorbar") +

...
```

Smoothing (with Cross Bars) – 2 Variables



MEMORIAL UNIVERSITY

Smoothing (with Cross Bars) – 2 Variables

```
...

ggplot(aes(contract_week, contract_value)) +
  geom_smooth() +
  geom_point() +
  stat_summary(fun.data=mean_sdl,
               fun.args=list(mult=1),
               geom="crossbar",
               width=10) +

...
```

Example Dataset 2 (”Fuel”)

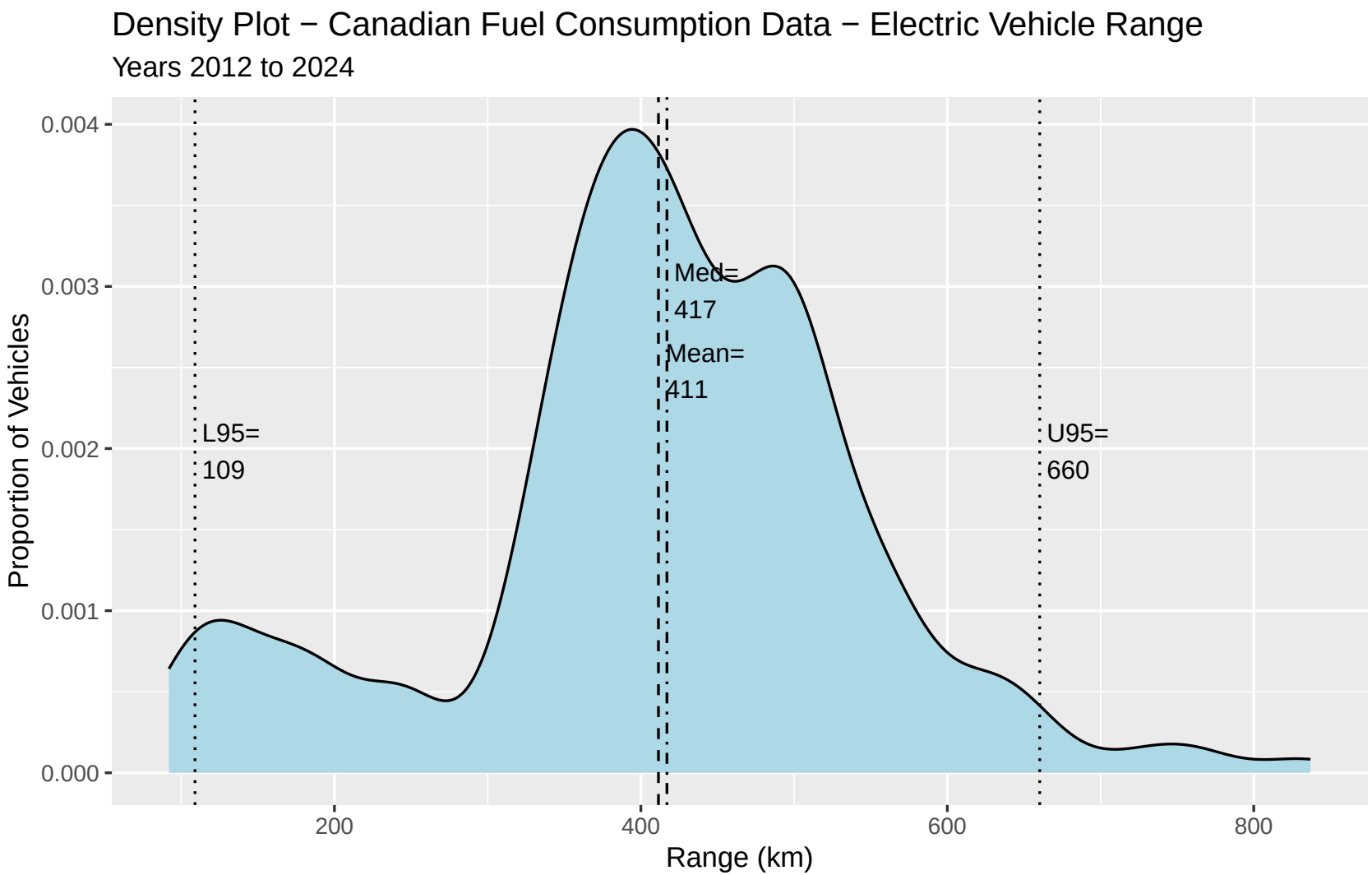
- Government of Canada, Open Government Portal
- Fuel Consumption Ratings – Battery-electric vehicles – 2012–2023
- Last updated Oct 10, 2023
- https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64

Column	Data Type
Make	Discrete
Model	Discrete
Year	Numeric
Category	Discrete[2]
City	Numeric[3]
Hwy	Numeric
Comb	Numeric
Range	Numeric[4]

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

MEMORIAL UNIVERSITY

Density Plot



Lower and Upper 95 percentile, median and mean

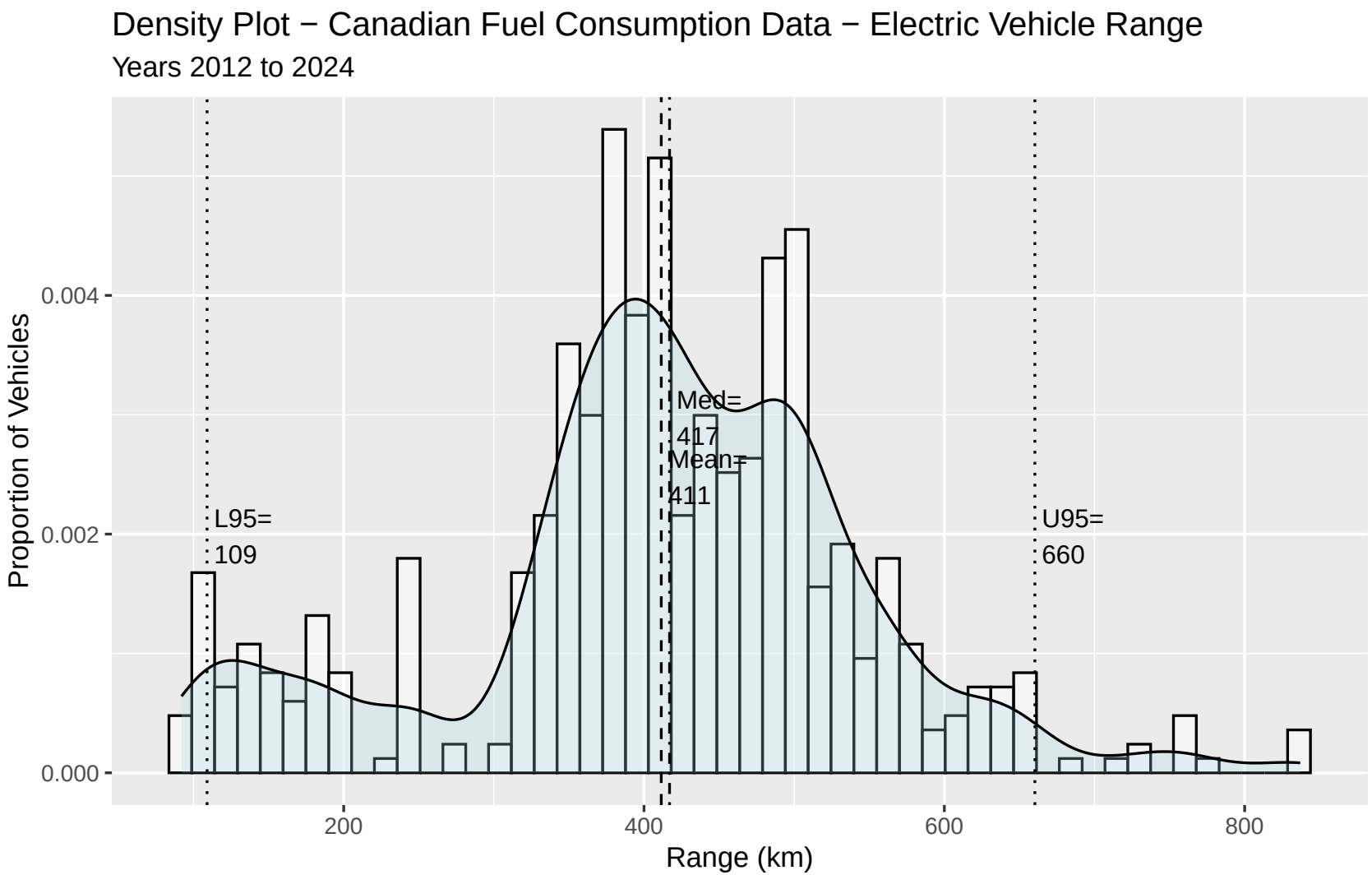
MEMORIAL UNIVERSITY

Density Plot

```
mean_v <- e.clean %>%
   summarize(mean_v = mean(Range), median_v = median(Range),
             lower95=quantile(Range, .025), upper95=quantile(Range, .9
             maxdensity = max(density(Range)$y))

e.clean %>%
   ggplot(aes(Range)) +
     geom_density(kernel='gaussian', fill='lightblue') +
     scale_x_continuous(labels=scales::comma) +
     scale_y_continuous(labels=scales::comma) +
     labs(x = 'Range (km)', y = 'Proportion of Vehicles',
          title='Density Plot - Canadian Fuel Consumption Data - Elect
          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') +
     ...
     annotate('text', label = paste(' L95=\n ', round(mean_v$lower95),
              x = mean_v$lower95,
              y = mean_v$maxdensity/2,
              size=3.5, hjust=0) +
     ...
```

MEMORIAL UNIVERSITY

Histogram



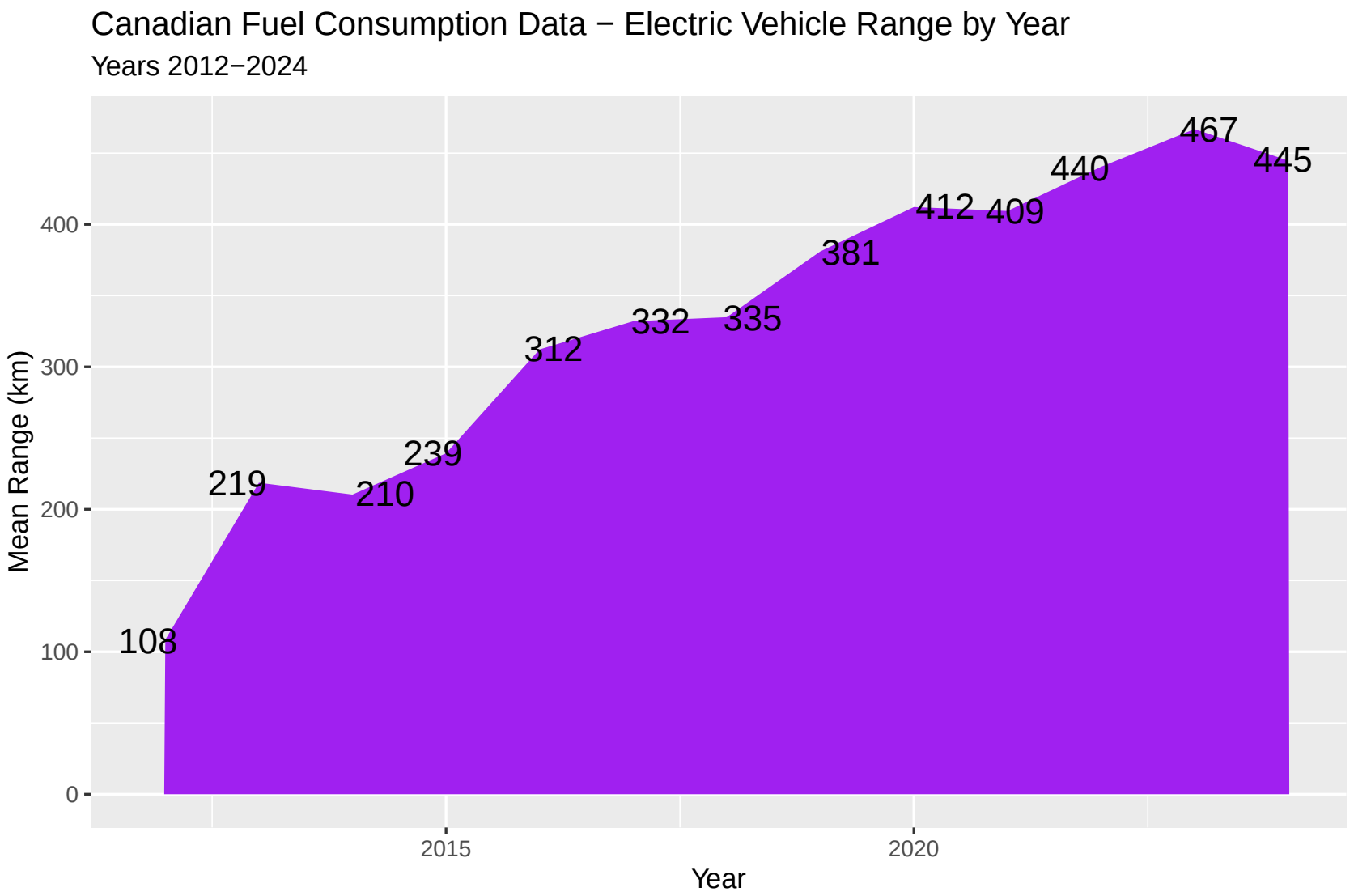
Lower and Upper 95 percentile, median and mean

MEMORIAL UNIVERSITY

Histogram

```
...

ggplot(aes(Range)) +
  geom_histogram(aes(y=..density..),
                 bins=50,
                 fill='white',
                 color='black',
                 alpha=0.5) +
  geom_density(kernel='gaussian', alpha=0.25, fill='lightblue') +
  scale_x_continuous(labels=scales::comma) +
  scale_y_continuous(labels=scales::comma) +

...
```

Area Plot

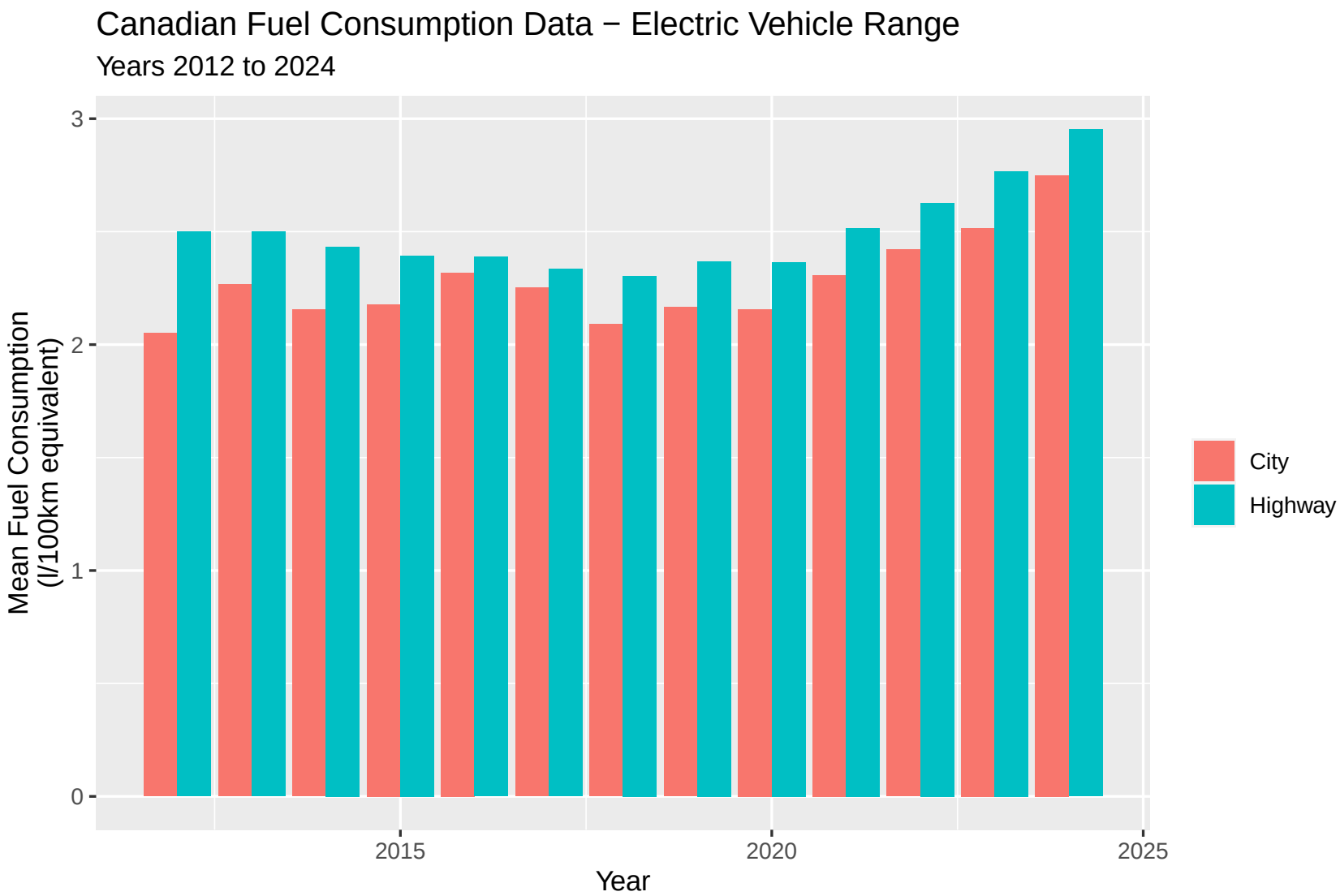


MEMORIAL UNIVERSITY

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') +
     scale_y_continuous(labels=scales::comma) +
     labs(x = 'Year', y = 'Mean Range (km)',
          title='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2012-2024')
```

Column Chart



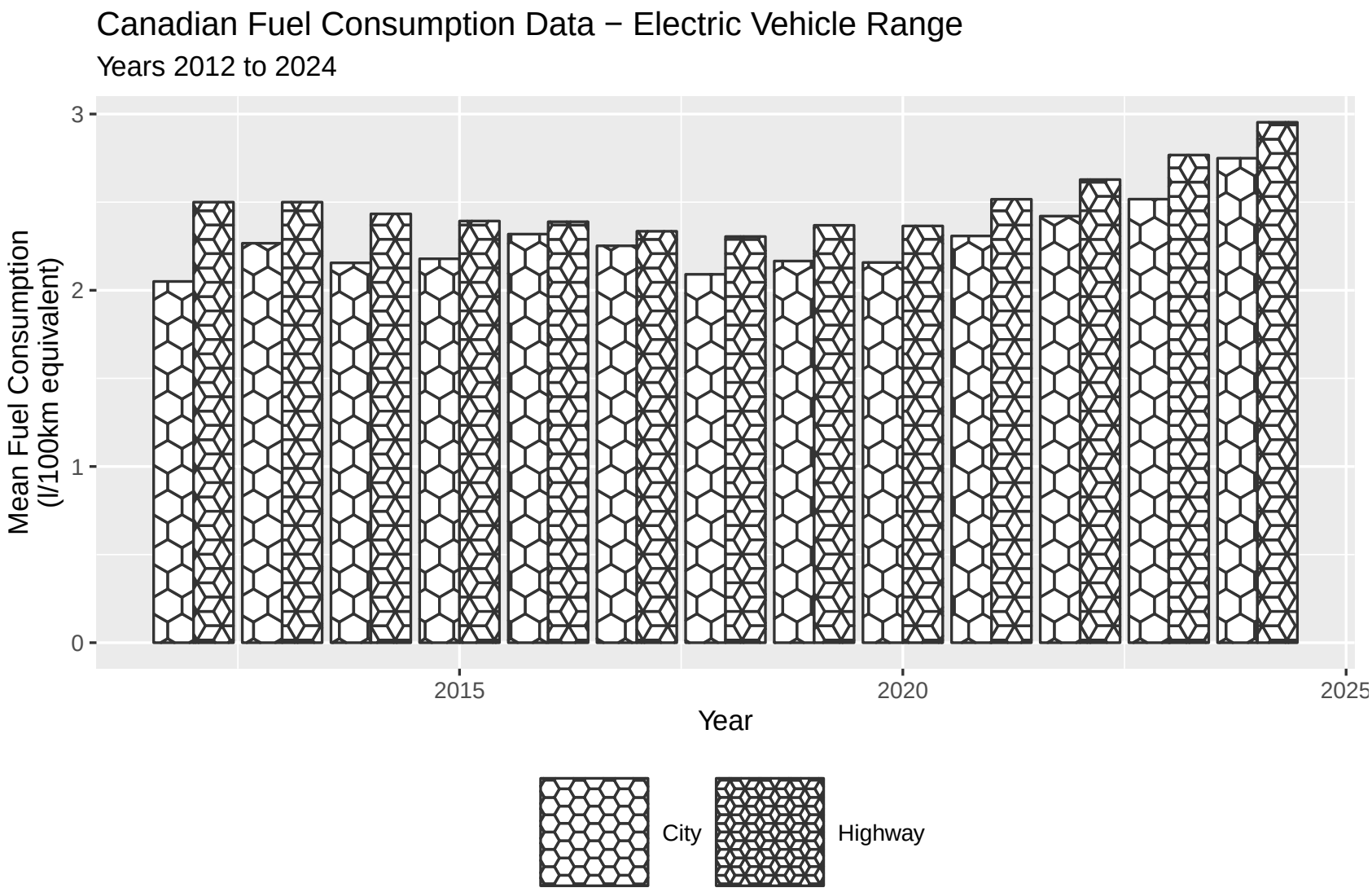
MEMORIAL UNIVERSITY

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")) +
      scale_y_continuous(labels=scales::comma) +
      labs(x = 'Year',
           y='Mean Fuel Consumption\n(l/100km equivalent)',
           fill='',
           title='Canadian Fuel Consumption Data - Electric Vehicle Ra
           subtitle='Years 2012 to 2024', )
```

MEMORIAL UNIVERSITY

Column Chart (with Patterns)

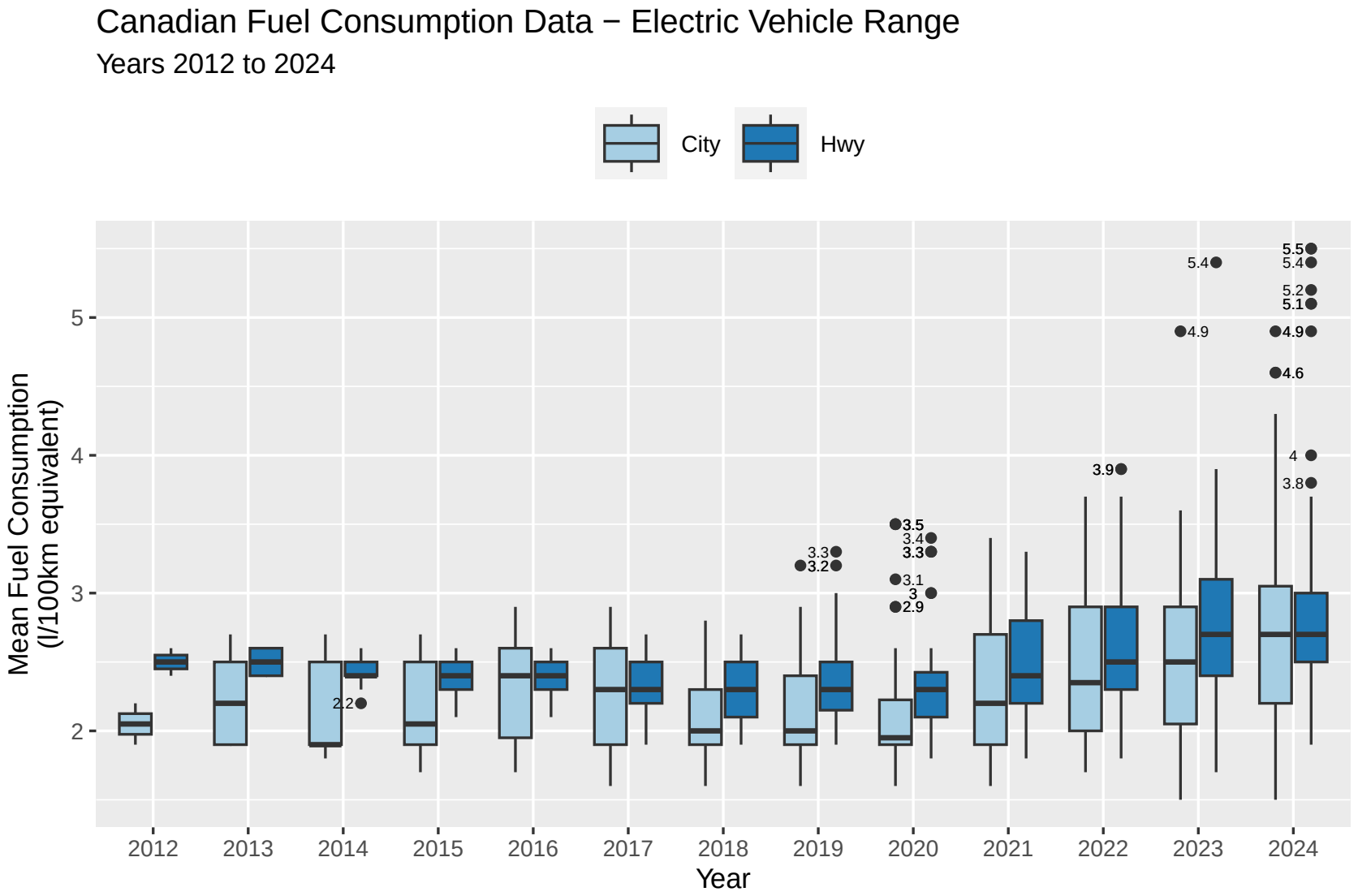


MEMORIAL UNIVERSITY

Column Chart (with Patterns)

```
e.clean %>%
   group_by(Year) %>%
   summarize(meanCity = mean(City), meanHwy = mean(Hwy)) %>%
   ungroup() %>%
   pivot_longer(cols=c('meanCity', 'meanHwy'), names_to='metric', valu
   ggplot(aes(Year, consumption)) +
      geom_col_pattern( aes(pattern_type=metric, pattern_angle=metric)
         position='dodge',
         pattern_fill='white',
         pattern='polygon_tiling',
         pattern_scale=0.5,
         pattern_key_scale_factor=0.4) +
      scale_pattern_type_manual(
         values = c('hexagonal', 'rhombille', 'pythagorean',
                    'truncated_square', 'rhombitrihexagonal',
                    'truncated_trihexagonal'),
         labels=c("City", "Highway")) +
      scale_y_continuous(labels=scales::comma) +
      labs(x = 'Year', y='Mean Fuel Consumption\n(l/100km equivalent)'
           pattern_type='',
           title='Canadian Fuel Consumption Data - Electric Vehicle Ra
           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')
```

Box Plot



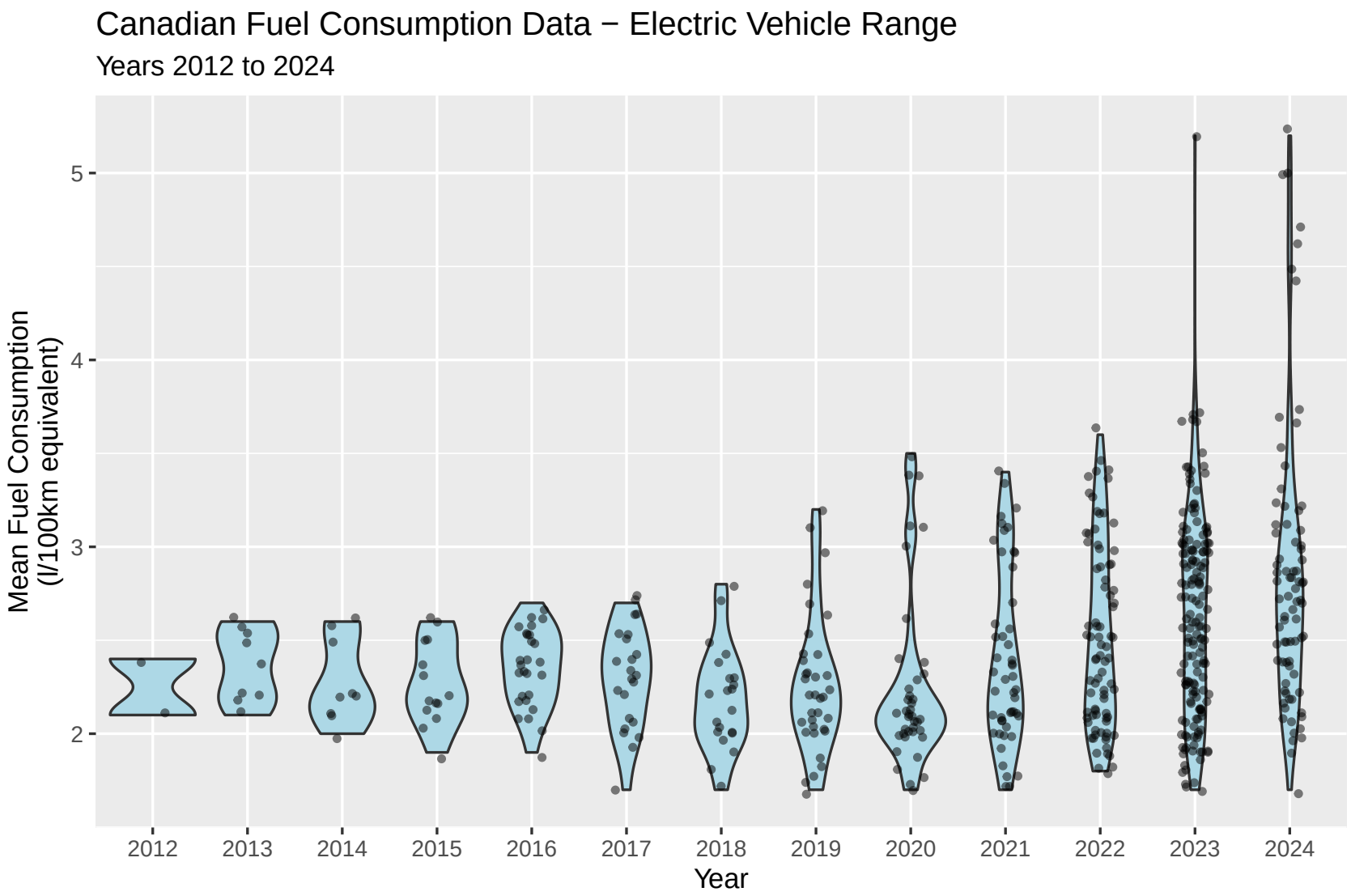
MEMORIAL UNIVERSITY

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() +
          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='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2012 to 2024', ) +
     scale_y_continuous(labels=scales::comma) +
     theme(legend.key.size=unit(1, 'cm'),
           legend.position='top')
```

MEMORIAL UNIVERSITY

Violin Plot



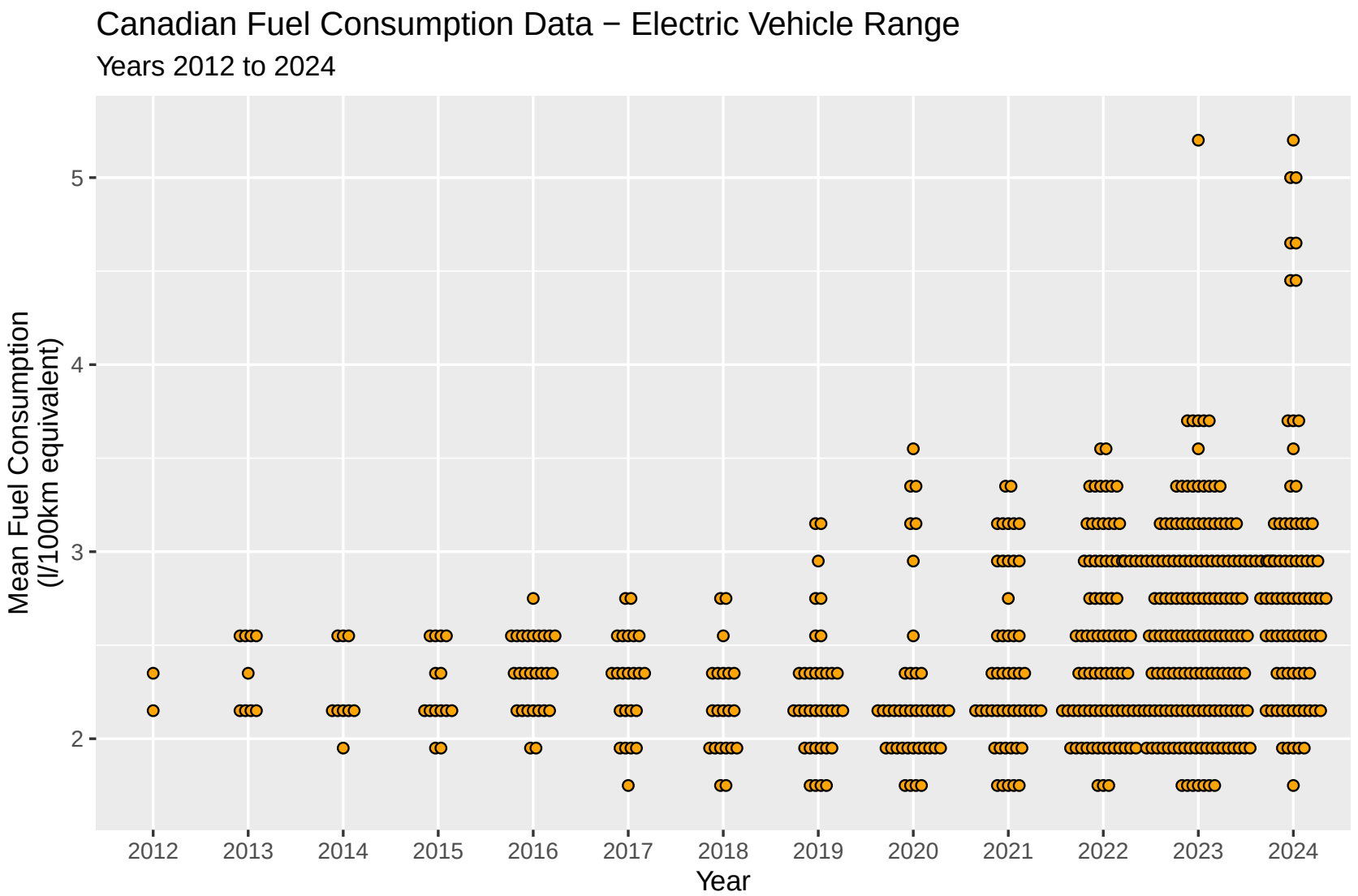
MEMORIAL UNIVERSITY

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='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2012 to 2024', ) +
     scale_y_continuous(labels=scales::comma) +
     theme(legend.position='top',
           legend.background=element_blank(),
           legend.box.background=element_rect(color='black',
                                              fill='lightgrey'),
           legend.key.size=unit(1, 'cm'))
```

MEMORIAL UNIVERSITY

Dot Plot



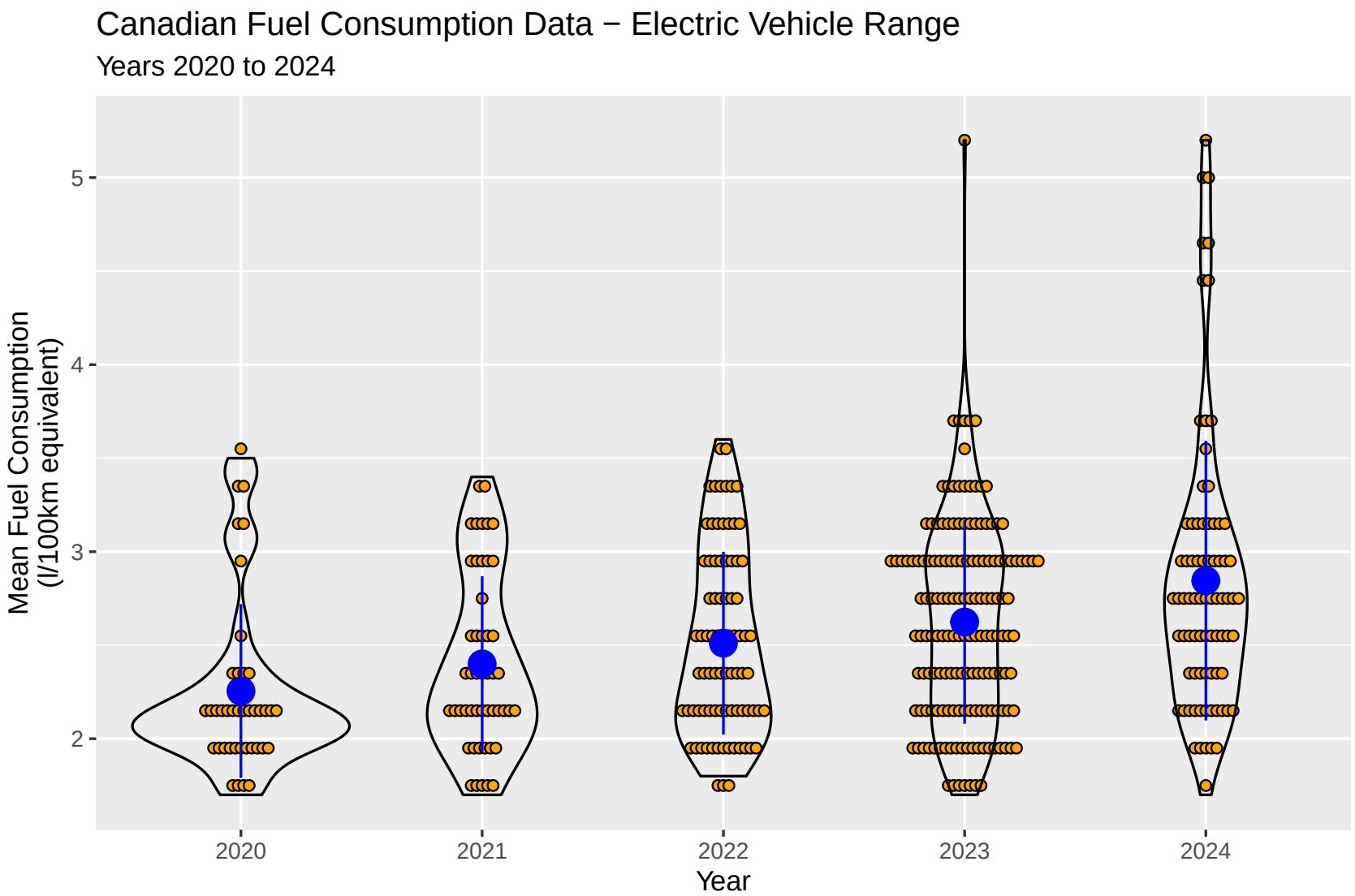
MEMORIAL UNIVERSITY

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 equivalent)',
          fill='',
          title='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2012 to 2024') +
     scale_y_continuous(labels=scales::comma) +
```

MEMORIAL UNIVERSITY

Dot Plot (with Violin and Range Summary)



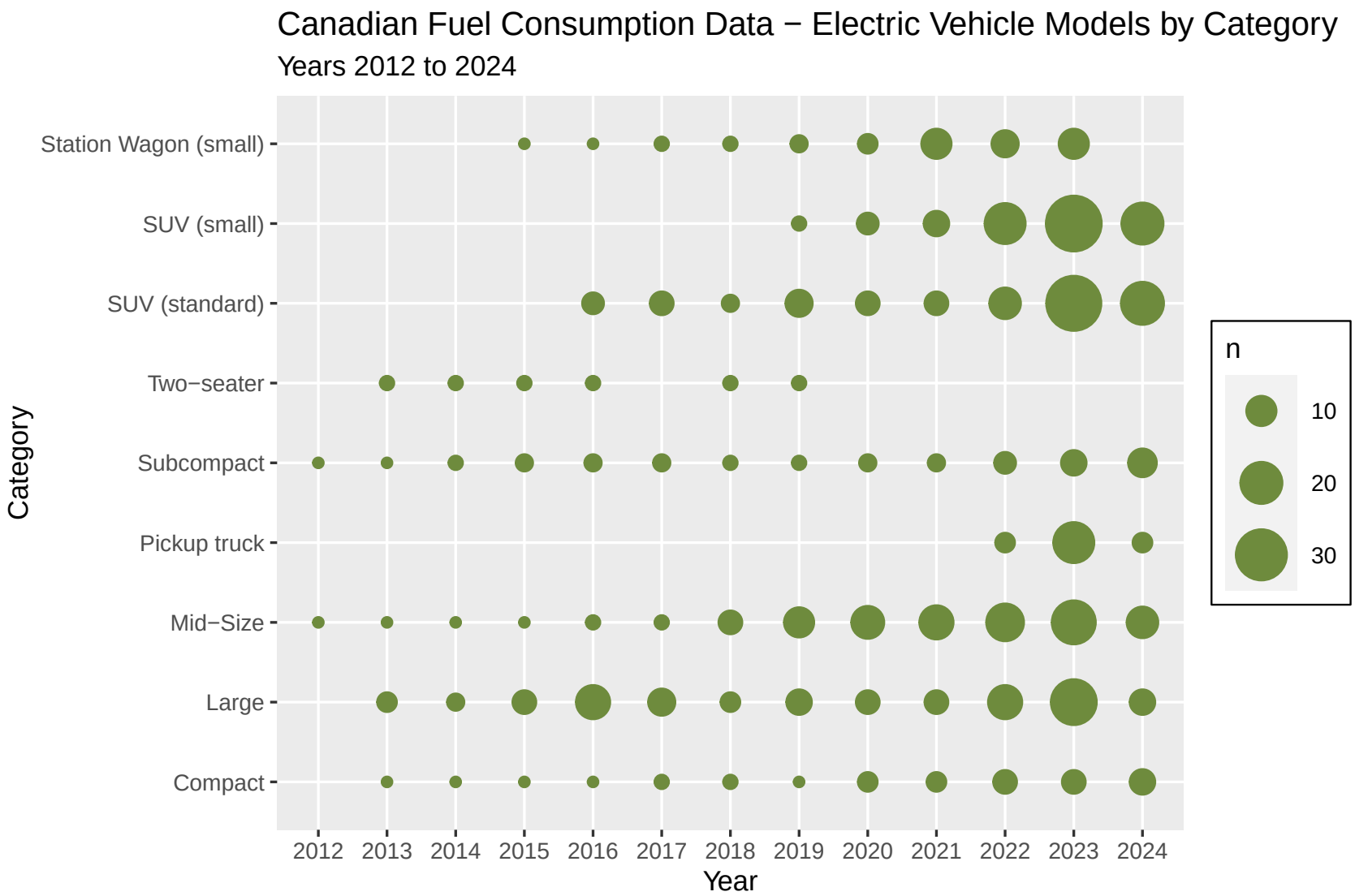
MEMORIAL UNIVERSITY

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 equivalent)',
          fill='',
          title='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2020 to 2024') +
     scale_y_continuous(labels=scales::comma) +
     theme(legend.position='none')
```

MEMORIAL UNIVERSITY

Count Plot



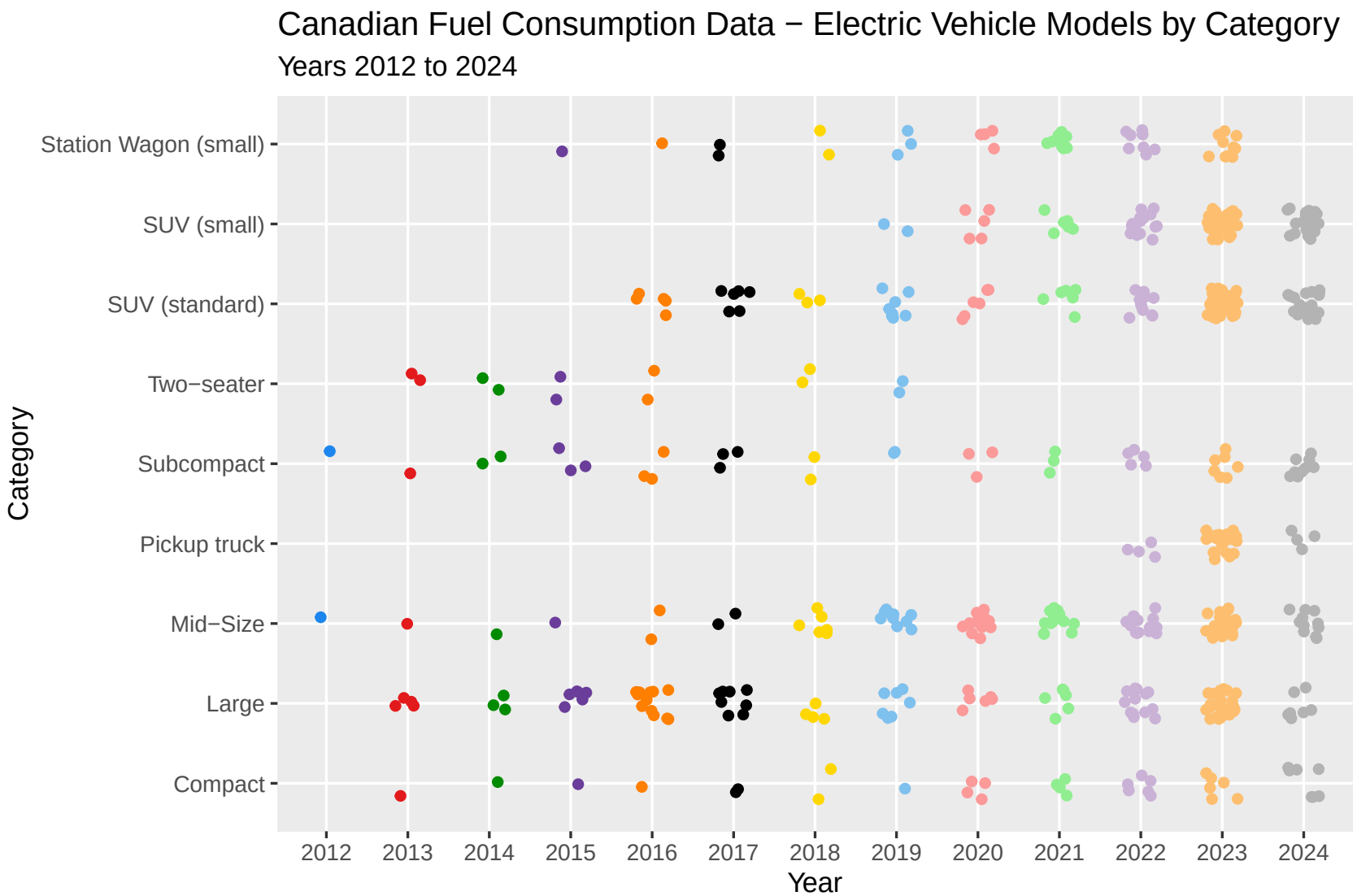
MEMORIAL UNIVERSITY

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='Canadian Fuel Consumption Data - Electric Vehicle Mo
           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

Jitter Plot



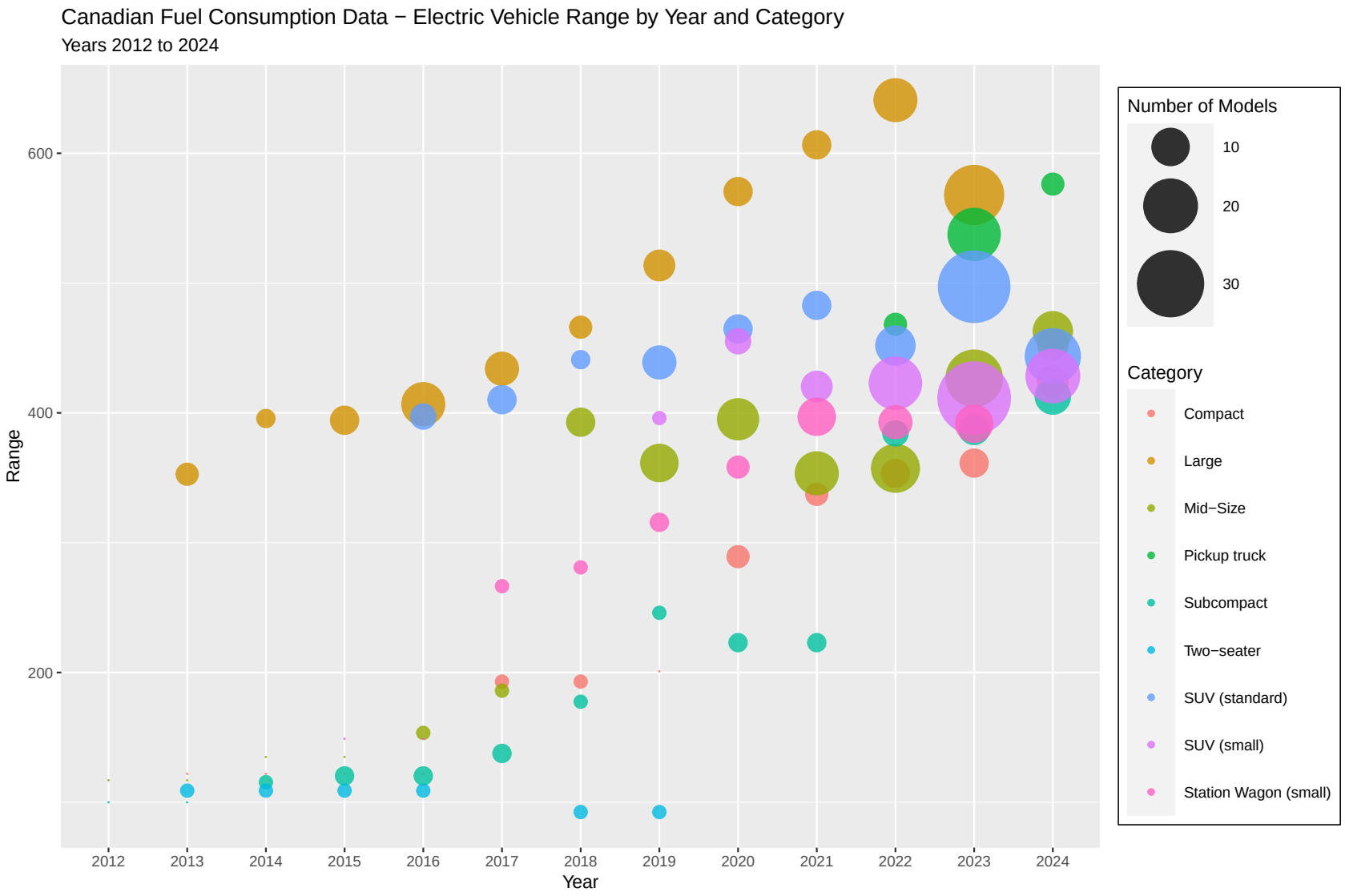
MEMORIAL UNIVERSITY

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 (standard)',
                       'SUV (small)', 'Station Wagon (small)')) +
          guides(color=FALSE) +
      labs(x = 'Year',
           y='Category',
           fill='Make',
           title='Canadian Fuel Consumption Data - Electric Vehicle Mo
           subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

Points Plot



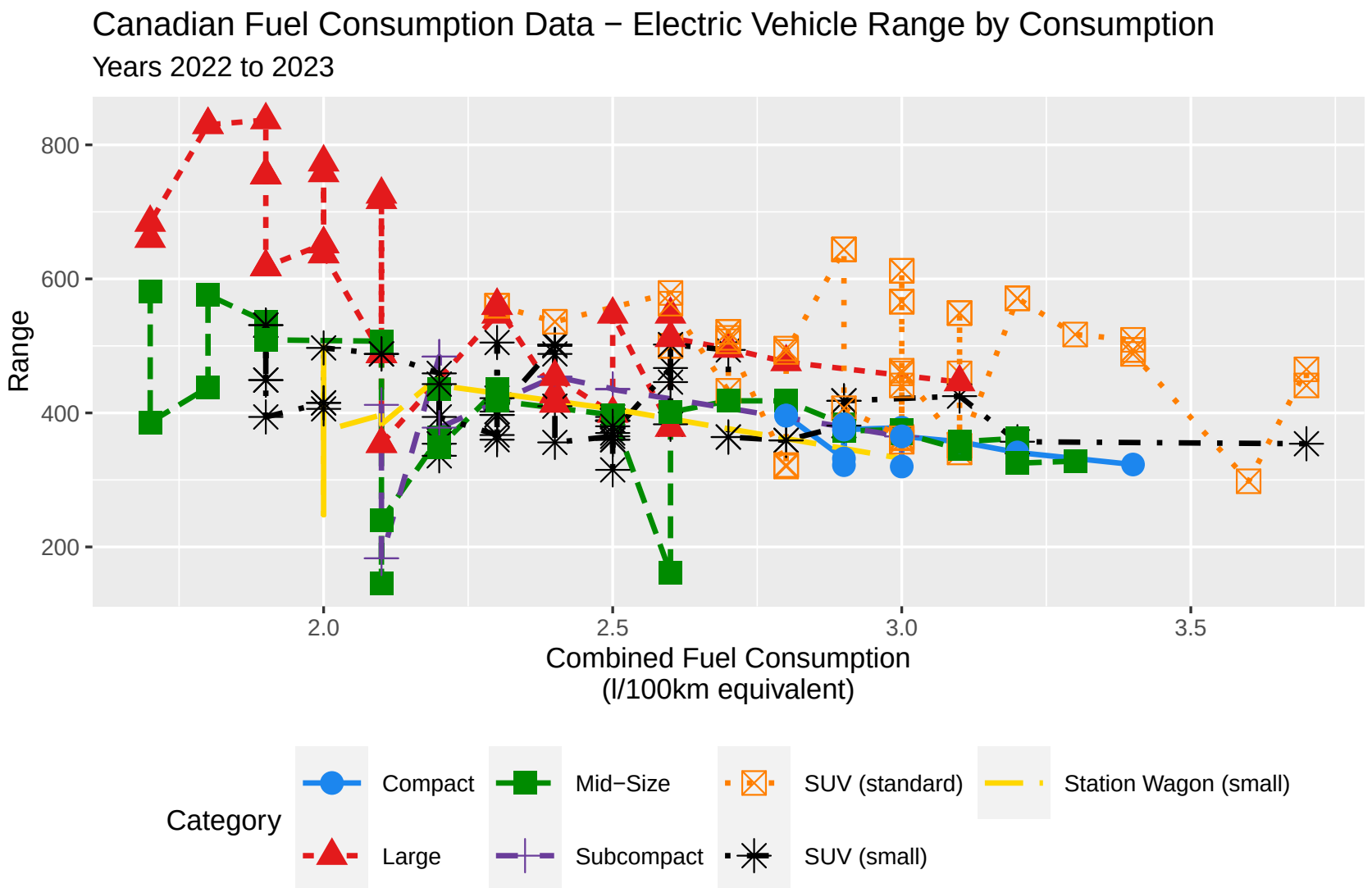
MEMORIAL UNIVERSITY

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 (standard)',
                       'SUV (small)', 'Station Wagon (small)')) +
    labs(x = 'Year', y='Range',
    fill='Make', size='Number of Models',
          title='Canadian Fuel Consumption Data - Electric Vehicle Ran
          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'))
```

MEMORIAL UNIVERSITY

Lines and Points Plot



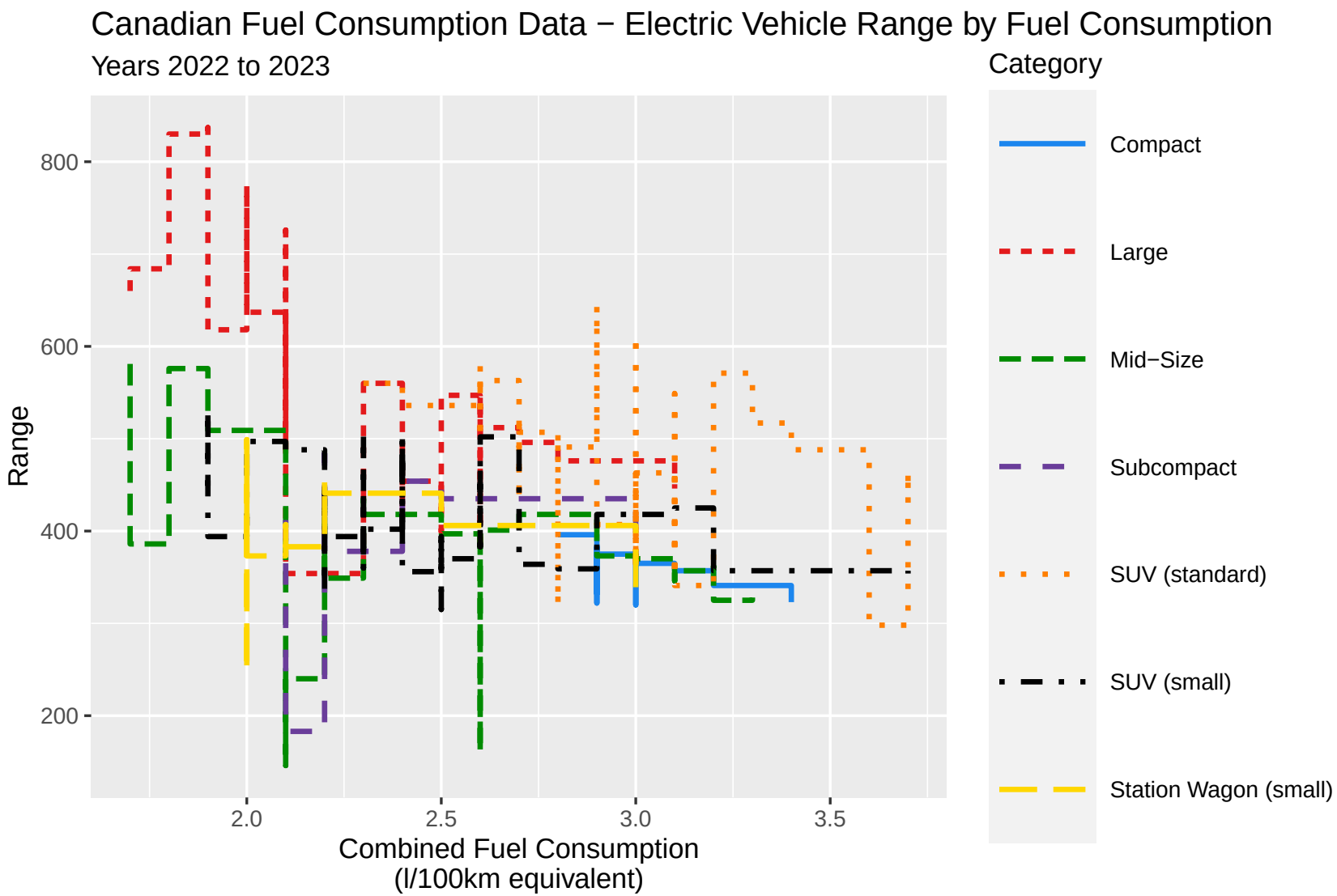
MEMORIAL UNIVERSITY

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

MEMORIAL UNIVERSITY

Stepped Lines Plot



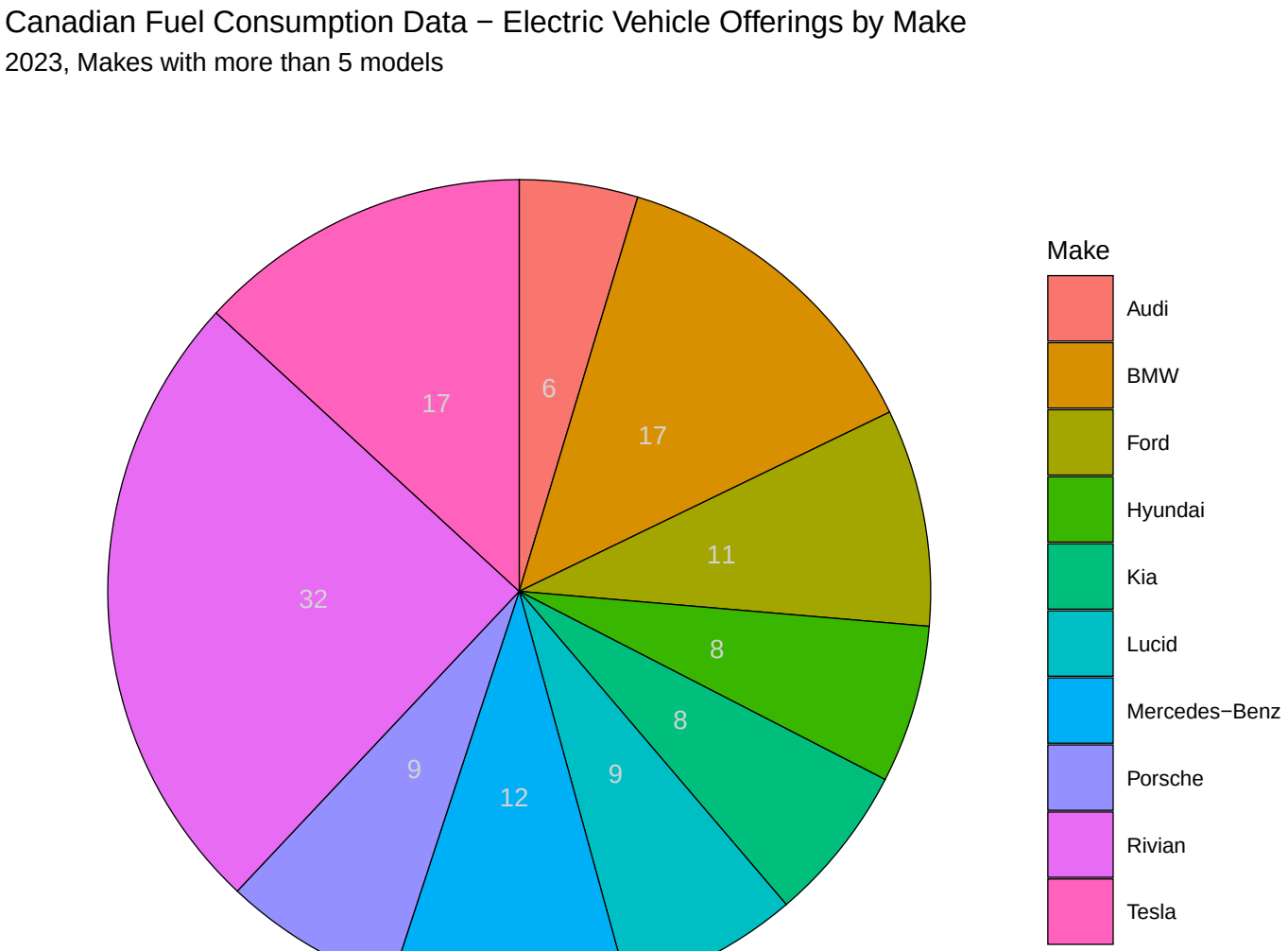
MEMORIAL UNIVERSITY

Stepped Lines Plot

```
e.clean %>%
   filter(Year >= 2022 & Year <= 2023) %>%
   filter(Comb <= 4) %>%
   filter(Category != 'PL') %>%
   filter(Category != 'T') %>%
   ggplot(aes(Comb, Range, color=Category, linetype=Category)) +
     geom_step(size=1) +
     scale_color_manual(values=c25,
          labels=c('Compact', 'Large', 'Mid-Size', 'Subcompact',
                   'SUV (standard)', 'SUV (small)',
                   'Station Wagon (small)')) +
     scale_linetype(
          labels=c('Compact', 'Large', 'Mid-Size', 'Subcompact',
                   'SUV (standard)', 'SUV (small)',
                   'Station Wagon (small)')) +
     scale_y_continuous(labels=scales::comma) +
     labs(x = 'Combined Fuel Consumption\n(l/100km equivalent)',
          y='Range',
          title='Canadian Fuel Consumption Data - Electric Vehicle Ran
          subtitle='Years 2022 to 2023', ) +
     theme(legend.key.size=unit(1.5, 'cm'))
```

MEMORIAL UNIVERSITY

Pie Chart



Pie Chart

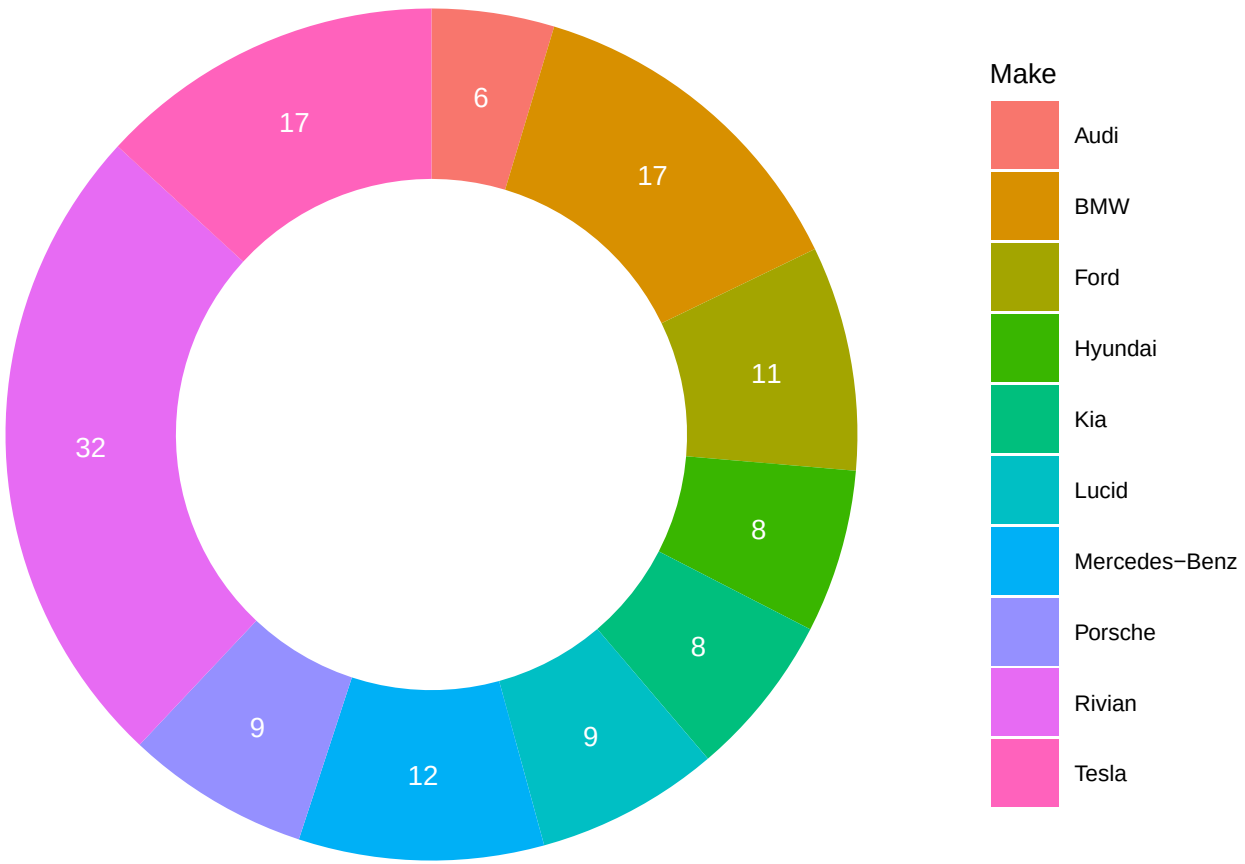
```
e.clean %>%
   filter(Year==2023) %>%
   group_by(Make) %>%
   summarize(totalcount = n()) %>%
   filter(totalcount >= 5) %>%
   ungroup() %>%
   ggplot(aes(x='', y=totalcount, fill=Make)) +
     geom_bar(stat='identity', color='black', size=0.25, width=1) +
     coord_polar('y', direction=-1, start=0) +
     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='Canadian Fuel Consumption Data - Electric Vehicle Of
           subtitle='2023, Makes with more than 5 models') +
      theme_void() +
      theme(legend.key.size=unit(1, 'cm'))
```

MEMORIAL UNIVERSITY

Donut Chart

Canadian Fuel Consumption Data − Electric Vehicle Offerings by Make

2023, Makes with more than 5 models

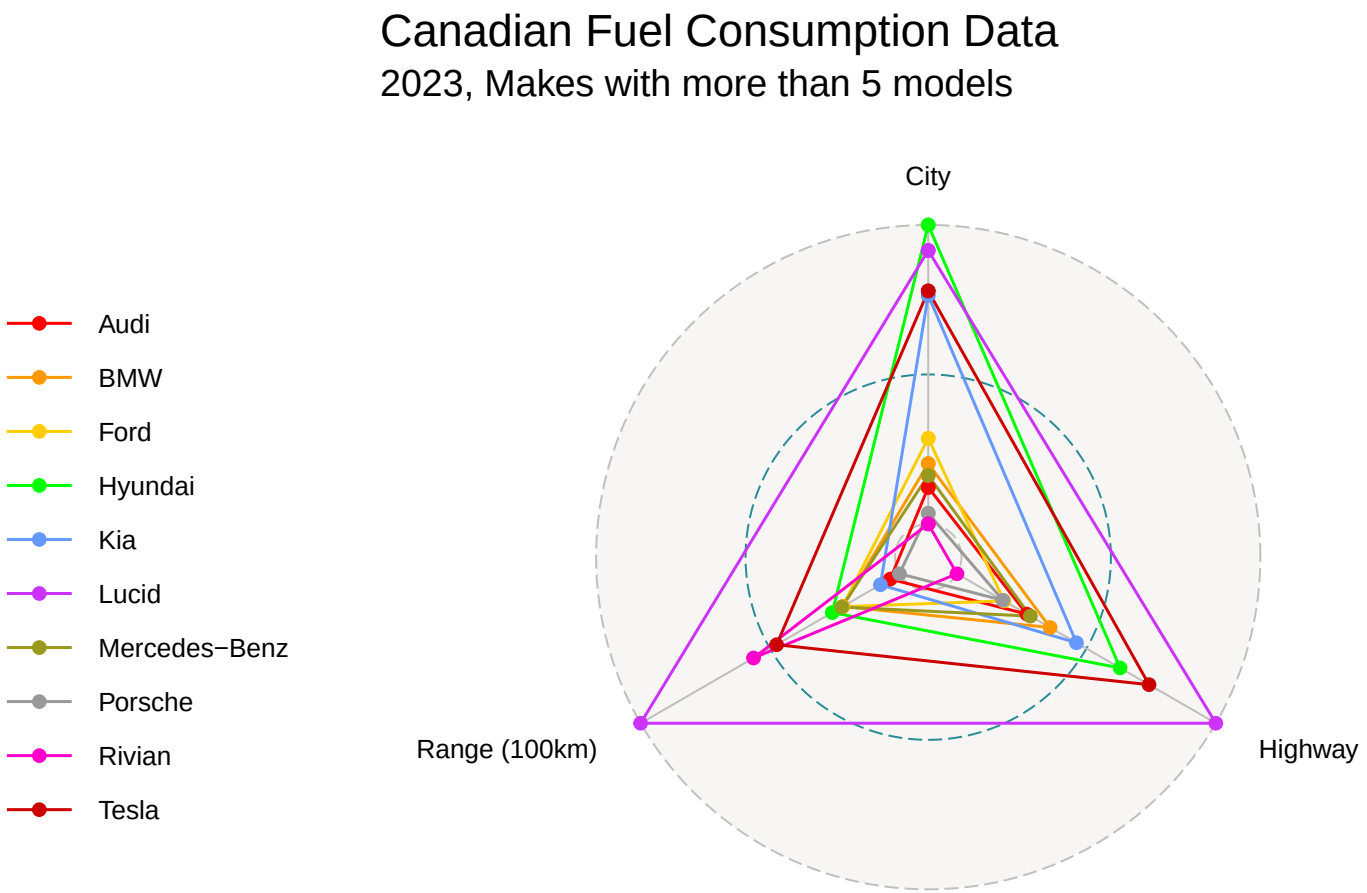


MEMORIAL UNIVERSITY

Donut Chart

```
holesize <- 2

....

   ggplot(aes(x=holesize, y=totalcount, fill=Make)) +
     geom_col() +
     coord_polar('y', direction=-1, start=0) +
     xlim(c(0.2, holesize+0.5)) +
     geom_text(aes(label=ifelse(totalcount > 5, totalcount, '')),
               color='white',
               position = position_stack(vjust=0.5)) +
...
```

Radar Plot

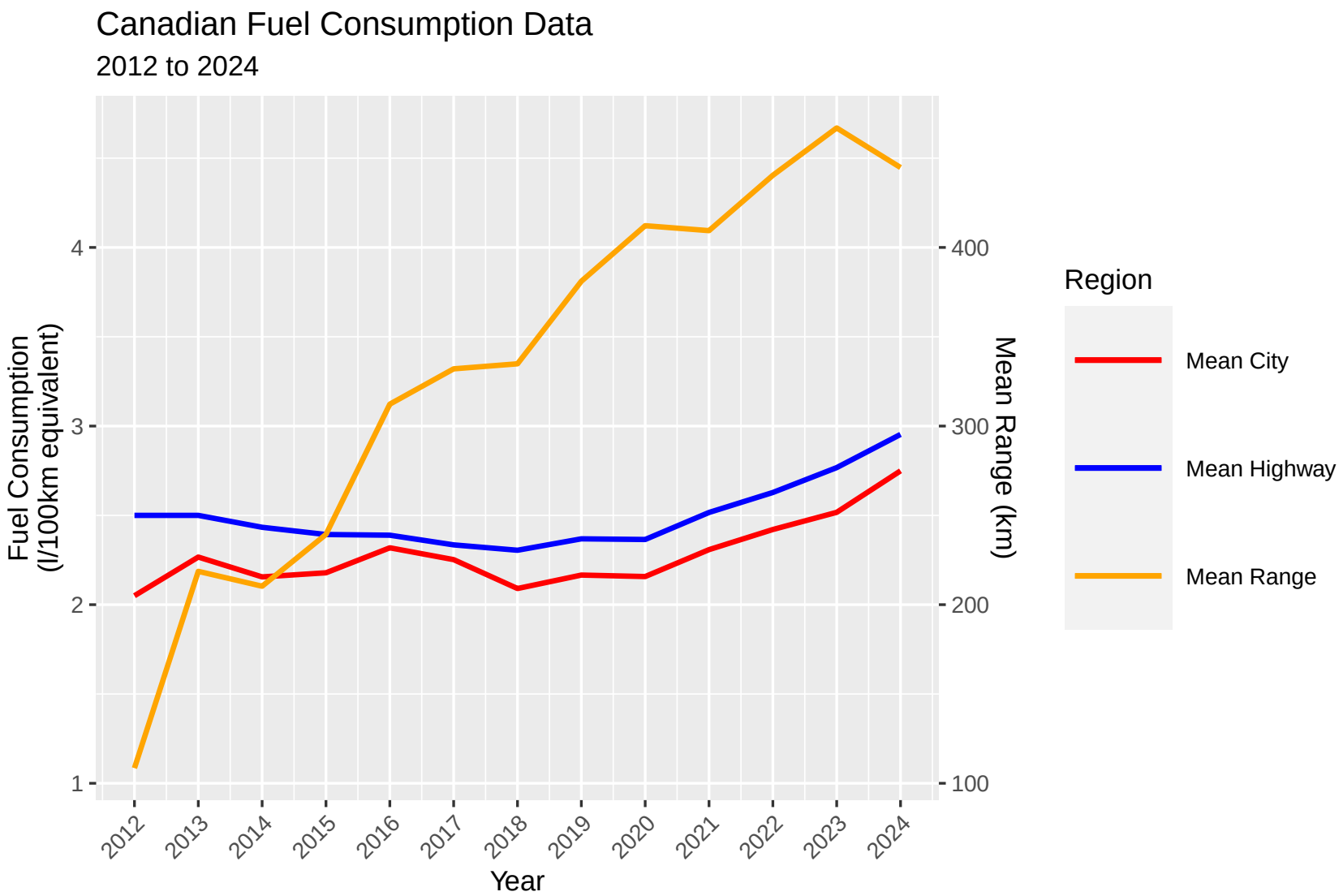


Radar Plot

```
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')
```

MEMORIAL UNIVERSITY

Lines with Multiple Axes



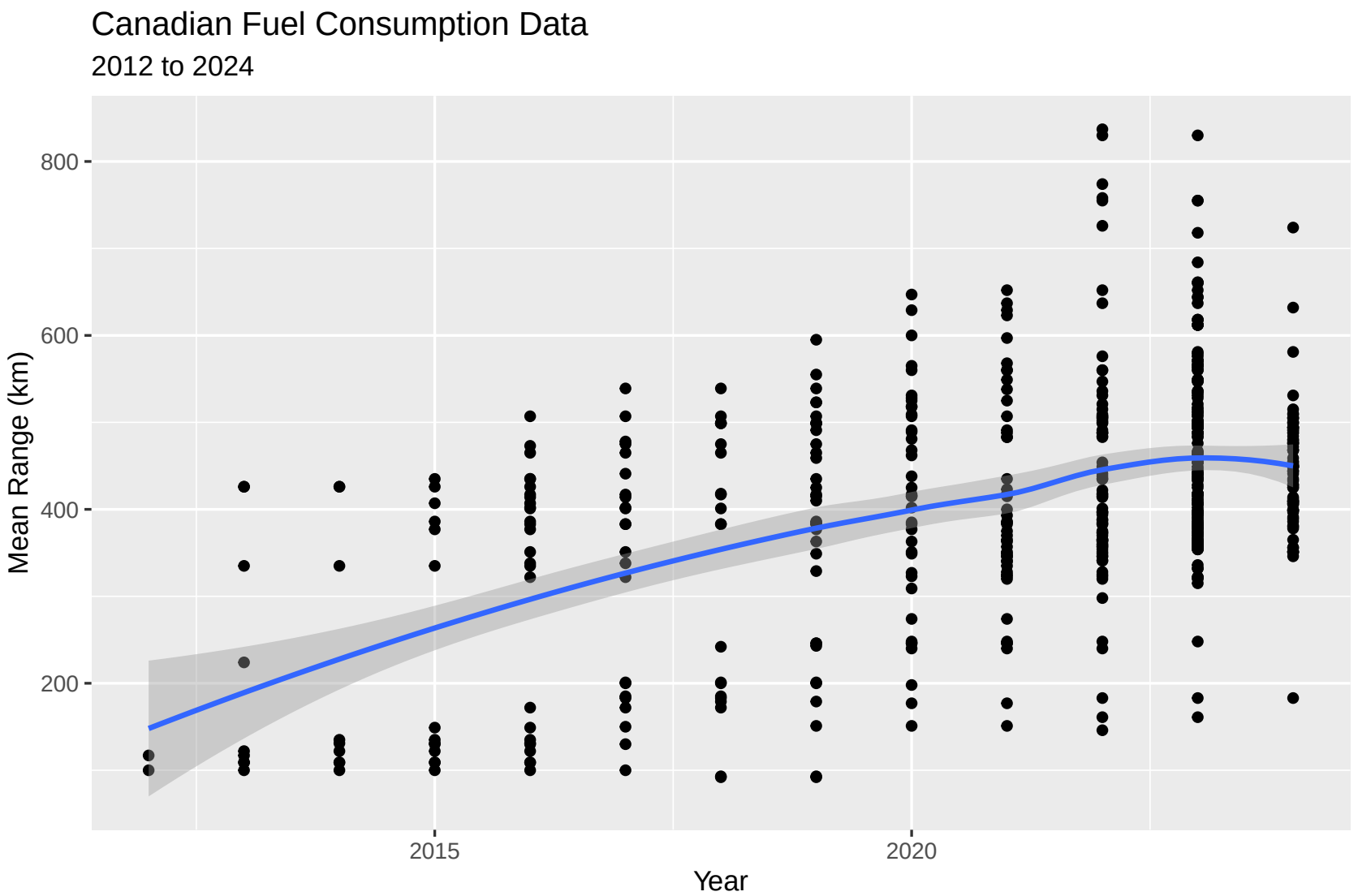
MEMORIAL UNIVERSITY

Lines with Multiple Axes

```
e.clean %>%
   group_by(Year) %>%
   summarize(meanCity = mean(City),
             meanHwy = mean(Hwy),
             meanRange = mean(Range)) %>%
   ungroup() %>%
   mutate(meanRange2 = meanRange/100) %>%
   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'), size=1) +
      geom_line(aes(y=meanHwy, color='Mean Highway'), size=1) +
      geom_line(aes(y=meanRange2, color='Mean Range'), size=1) +
      scale_y_continuous(labels=scales::comma,
          name="Fuel Consumption\n(l/100km equivalent)",
          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='', y = 'Mean Fuel Consumption\n(l/100km
           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

Local Regression Smoothing Plot

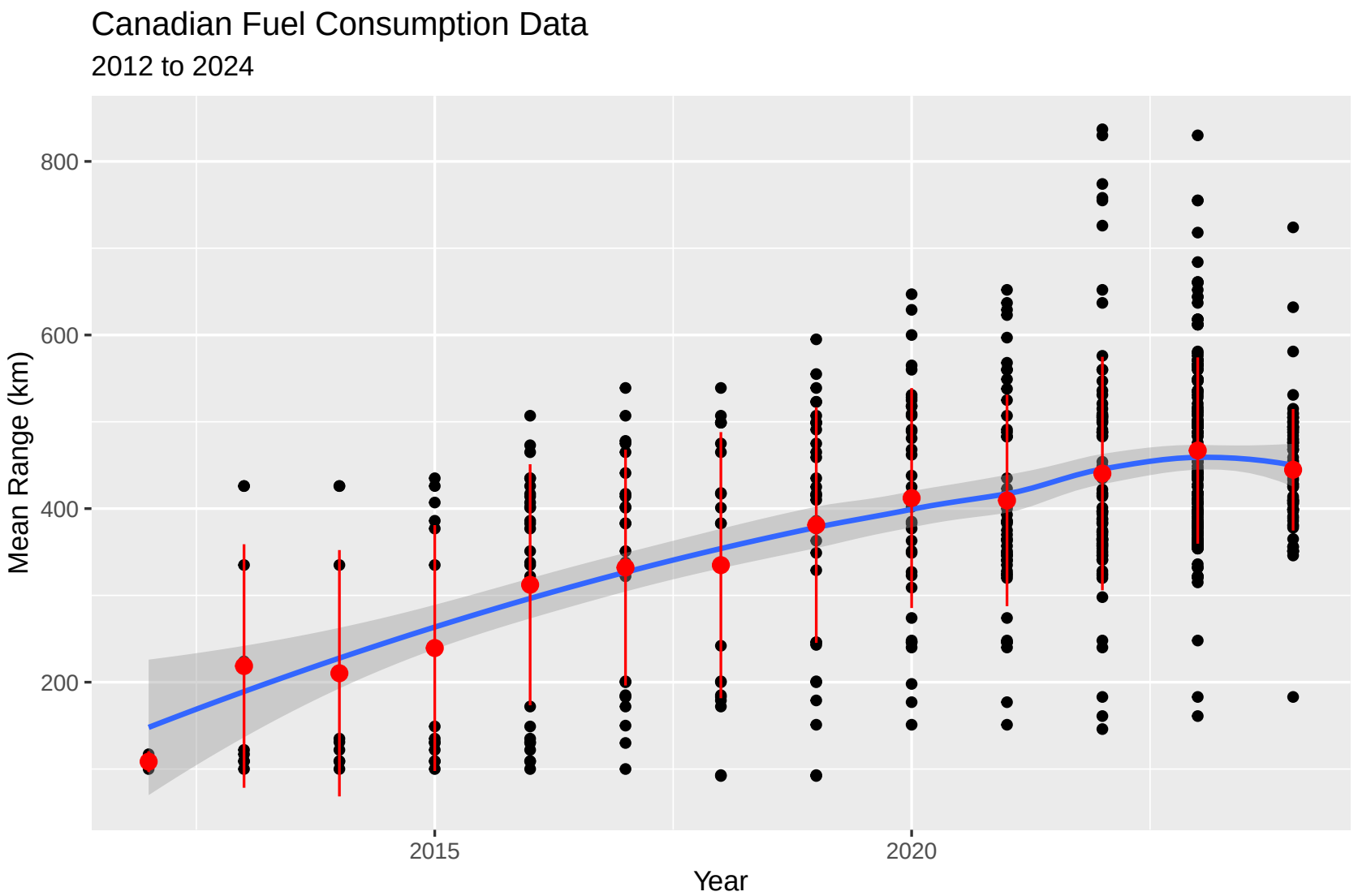


MEMORIAL UNIVERSITY

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')
```

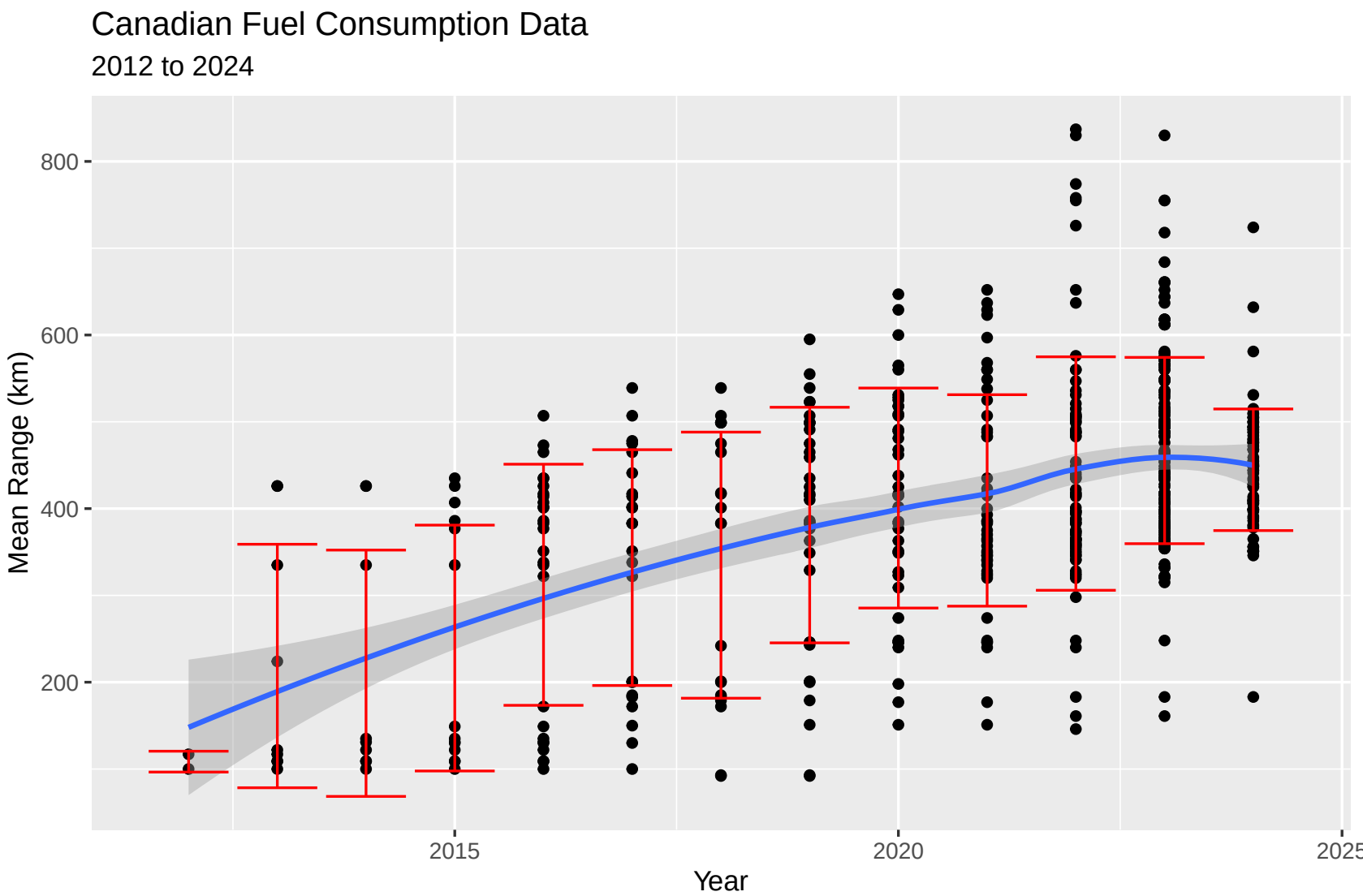
Local Regression Smoothing Plot (with range bar)



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')
```

Local Regression Smoothing Plot (with error bars)

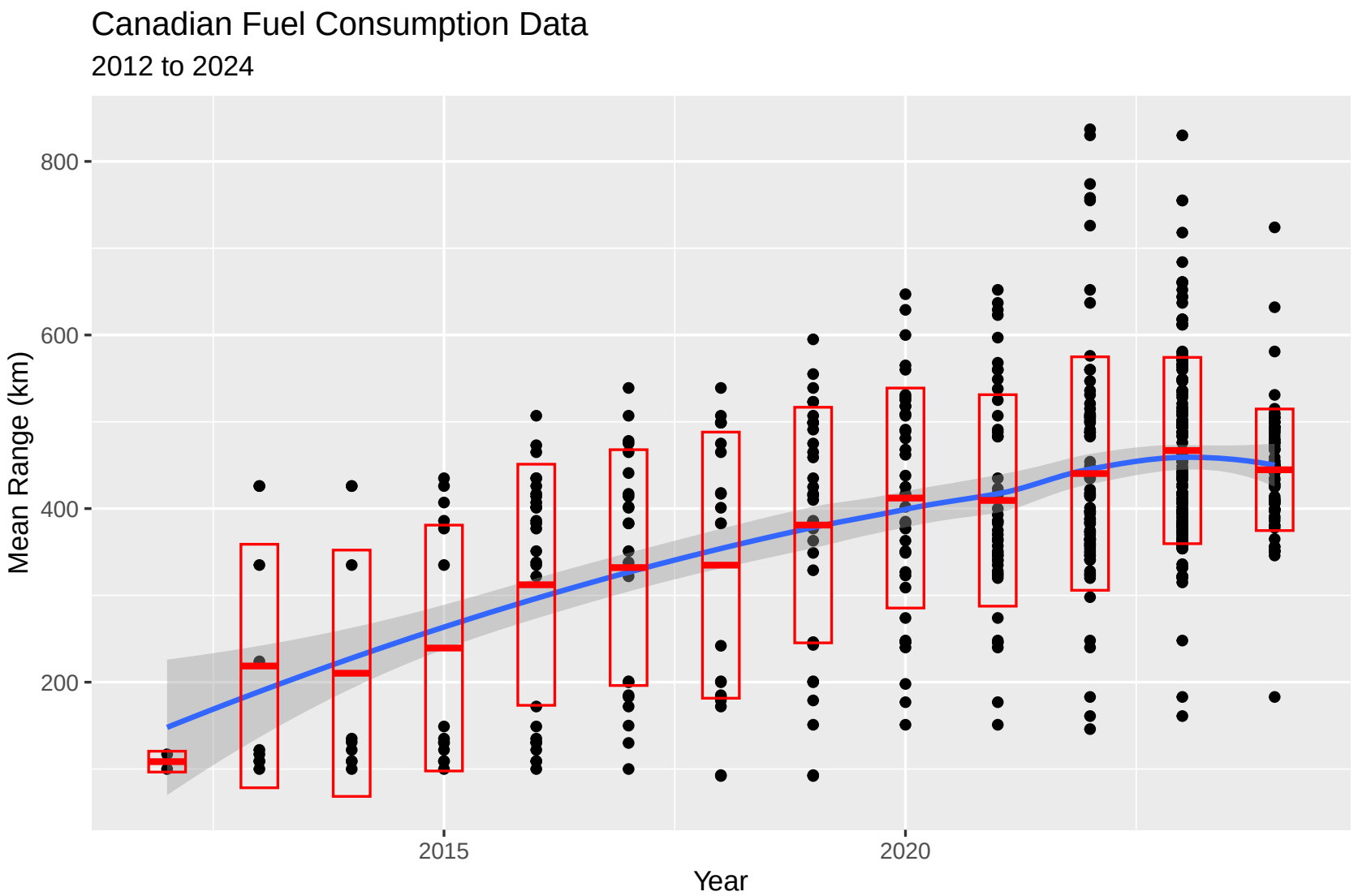


MEMORIAL UNIVERSITY

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')
```

Local Regression Smoothing Plot (with cross bars)



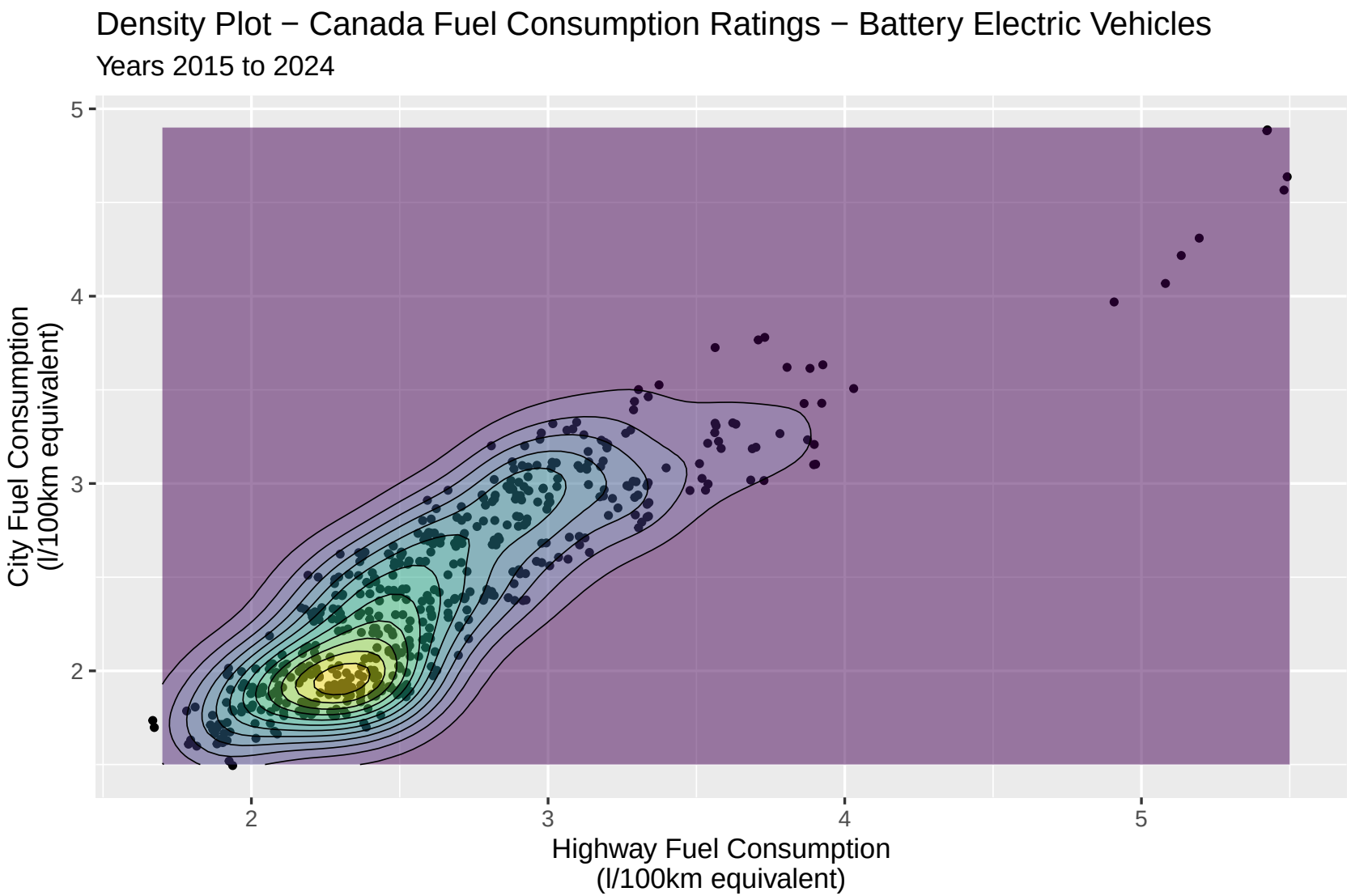
MEMORIAL UNIVERSITY

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

2D Density Plot

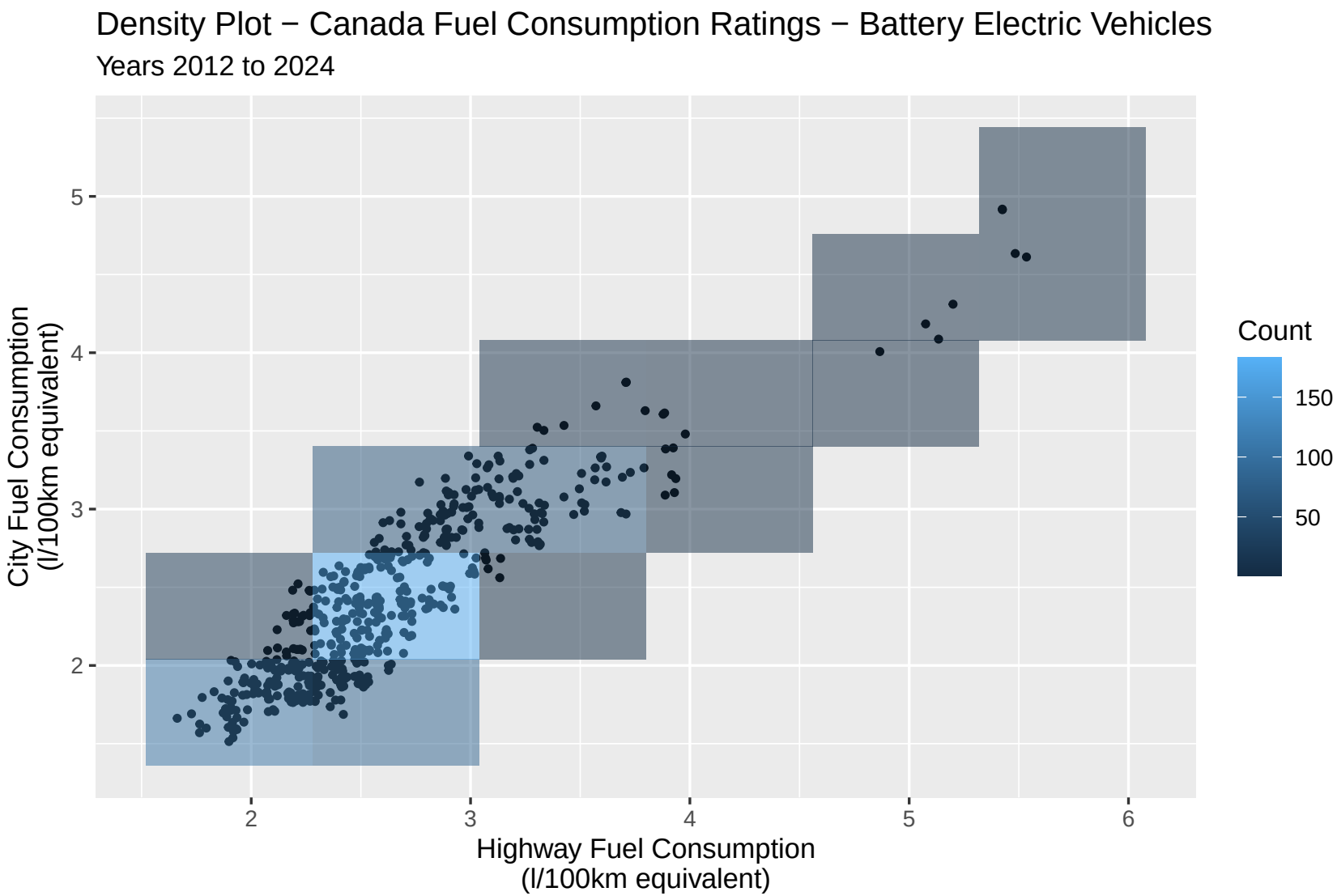


MEMORIAL UNIVERSITY

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 Fuel Consumption\n(l/100km equivalent)',
          y = 'City Fuel Consumption\n(l/100km equivalent)',
          title='Density Plot - Canada Fuel Consumption Ratings - Batt
          subtitle='Years 2015 to 2024') +
     theme(legend.position='none')
```

2D Bin Plot



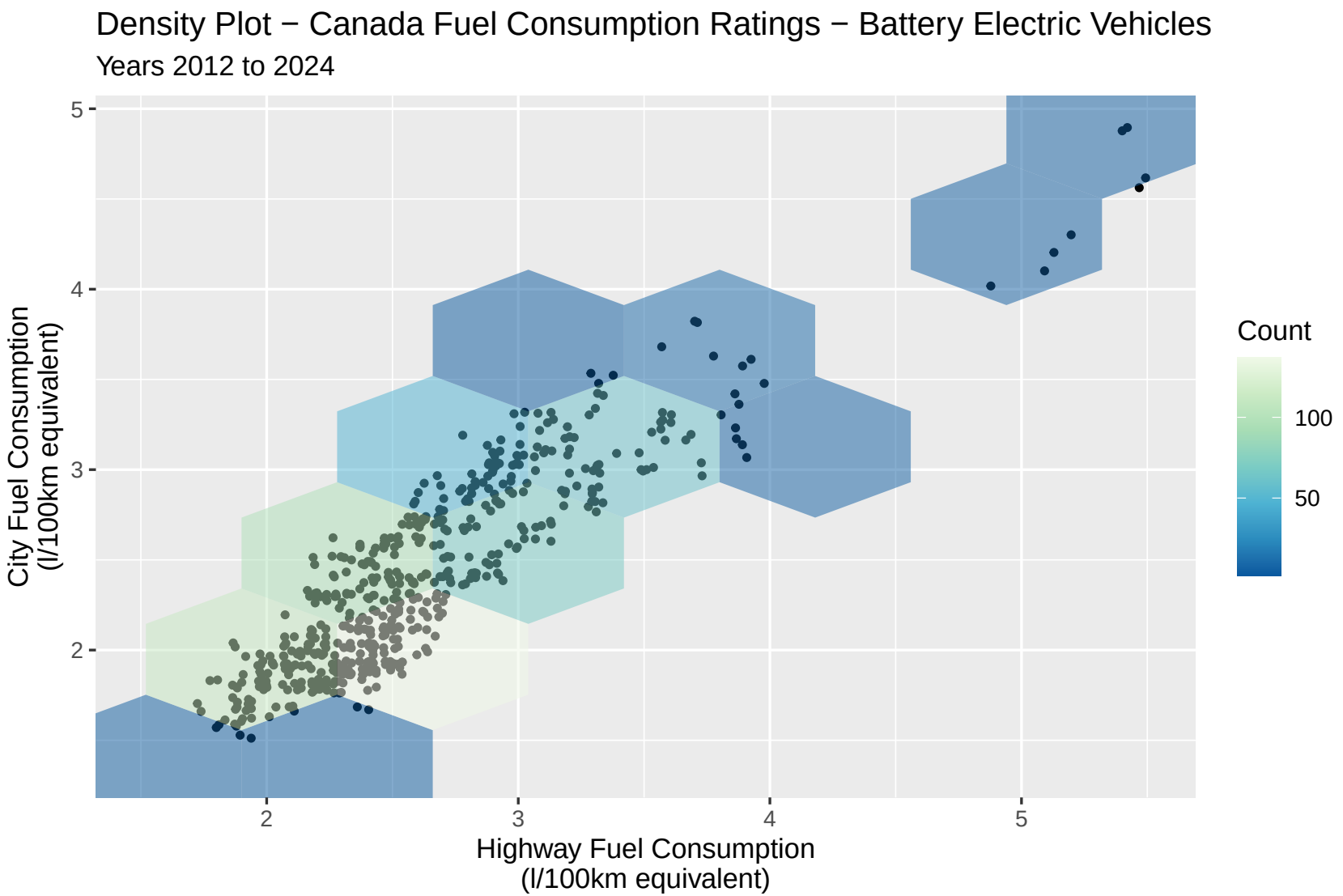
MEMORIAL UNIVERSITY

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 Fuel Consumption\n(l/100km equivalent)',
          y = 'City Fuel Consumption\n(l/100km equivalent)',
          fill='Count',
          title='Density Plot - Canada Fuel Consumption Ratings - Batt
          subtitle='Years 2012 to 2024')
```

MEMORIAL UNIVERSITY

2D Hex Plot

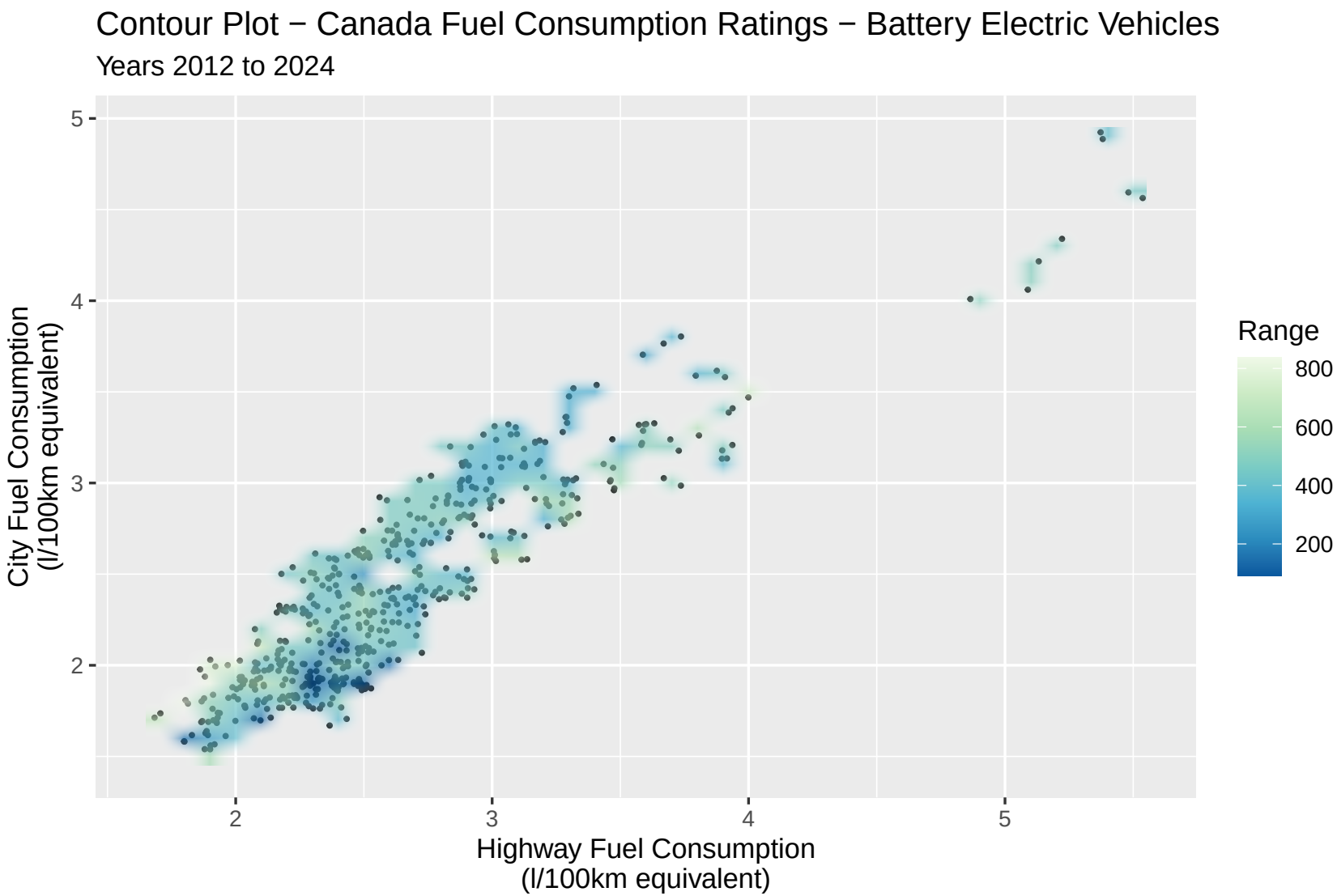


MEMORIAL UNIVERSITY

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 Fuel Consumption\n(l/100km equivalent)',
          y = 'City Fuel Consumption\n(l/100km equivalent)',
          fill='Count',
          title='Density Plot - Canada Fuel Consumption Ratings - Batt
          subtitle='Years 2012 to 2024')
```

3D Raster Plot

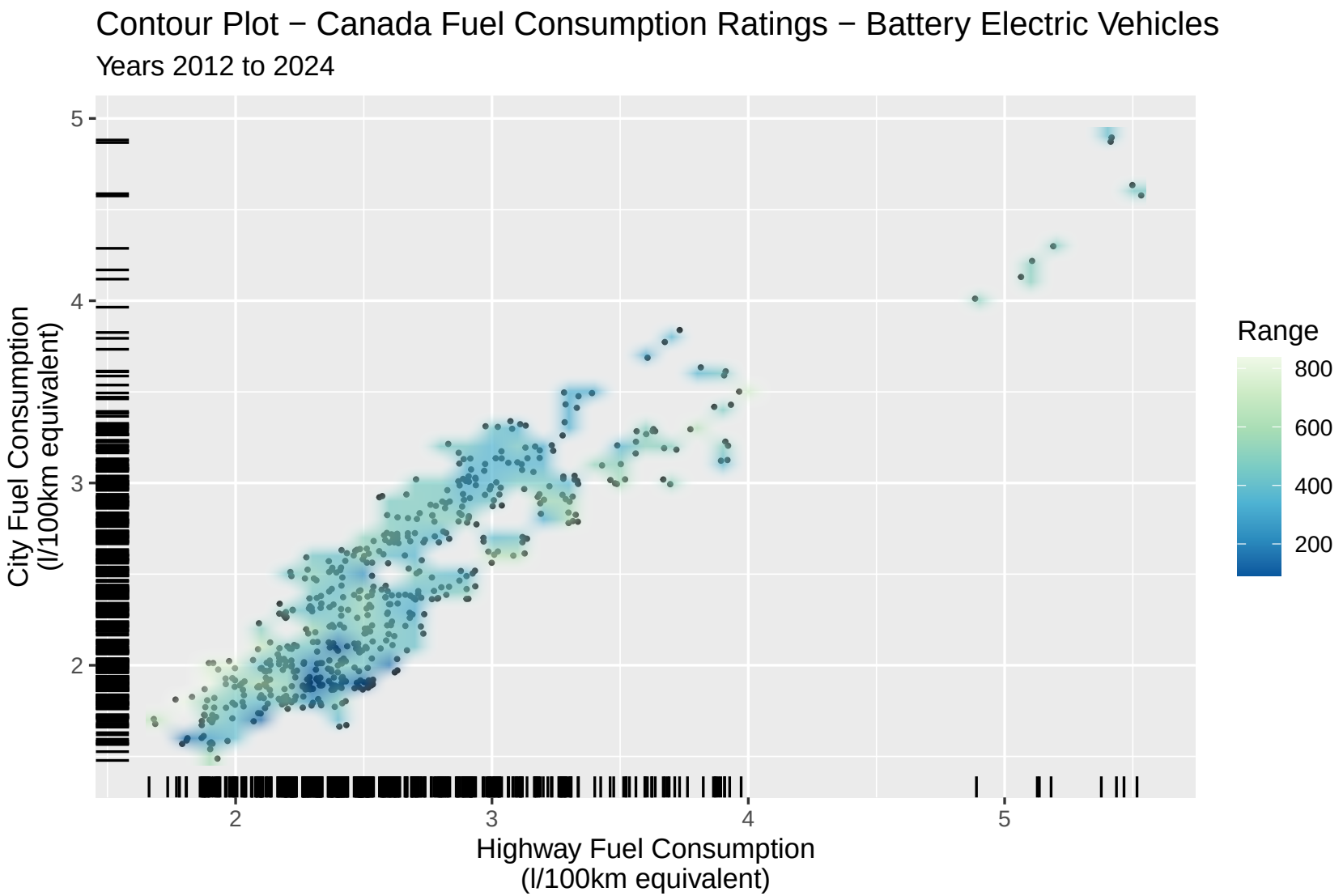


MEMORIAL UNIVERSITY

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 Fuel Consumption\n(l/100km equivalent)',
          y = 'City Fuel Consumption\n(l/100km equivalent)',
          fill='Range',
          title='Contour Plot - Canada Fuel Consumption Ratings - Batt
          subtitle='Years 2012 to 2024')
```

3D Raster Plot (with rug)



MEMORIAL UNIVERSITY

3D Raster Plot (with rug)

```
e.clean %>%
   ggplot(aes(x=Hwy, y=City)) +
     geom_point(color="black", size=0.5, position='jitter') +
     geom_rug(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 Fuel Consumption\n(l/100km equivalent)',
          y = 'City Fuel Consumption\n(l/100km equivalent)',
          fill='Range',
          title='Contour Plot - Canada Fuel Consumption Ratings - Batt
          subtitle='Years 2012 to 2024')
```

Geographic Data (Maps)

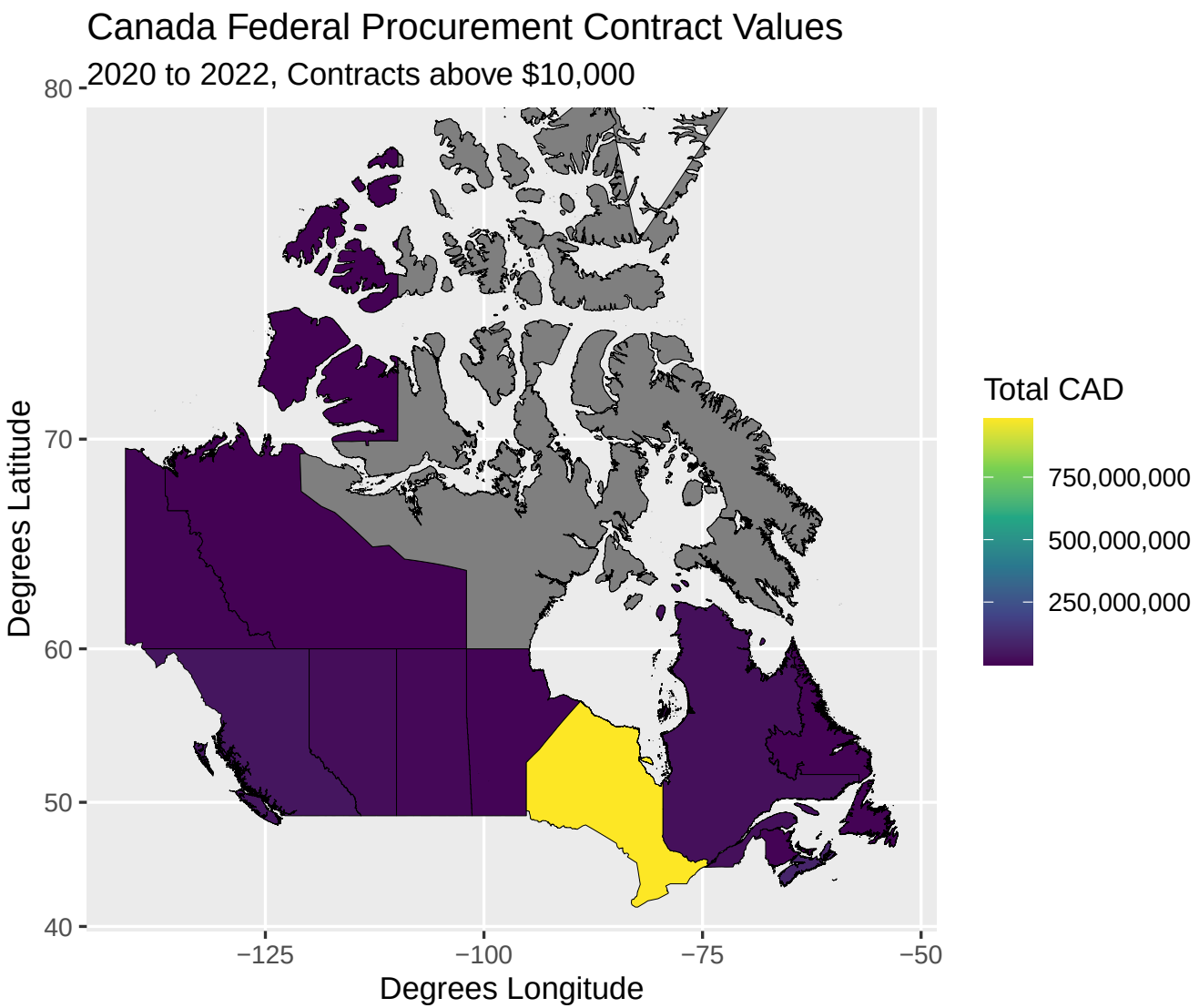
GADM Data

- Database of Global Administrative Areas
- https://gadm.org/
- ArcGIS shape file package for Canada

Statistics Canada Data

- 2016 Census Boundary Files
- https://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/bound-limit-2016-eng.cfm
- ArcGIS shape file for provinces/territories

Contracts Data Set (GADM Data)



MEMORIAL UNIVERSITY

Contracts Data Set (GADM Data)

```
canada.gadm <- getData("GADM",country="CAN",level=1)
canada.gadm.tidy <- tidy(canada.gadm)

d.gadm.id <- as.data.frame(
        d.clean %>% mutate(id = case_when(
                vendor_province == 'NL' ~ '9',
                vendor_province == 'PE' ~ '2',
                vendor_province == 'NS' ~ '11',
                vendor_province == 'NB' ~ '8',
                vendor_province == 'QC' ~ '3',
                vendor_province == 'ON' ~ '13',
                vendor_province == 'MB' ~ '7',
                vendor_province == 'SK' ~ '4',
                vendor_province == 'AB' ~ '1',
                vendor_province == 'BC' ~ '6',
                vendor_province == 'NT' ~ '10',
                vendor_province == 'YK' ~ '5',
                vendor_province == 'NU' ~ '12',
                TRUE ~ '-1')) %>%
        filter(vendor_province != 'UNKWN') %>%
        filter(contract_date >= '2020-01-01') %>%
        filter(contract_date <= '2022-12-31') %>%
        group_by(id) %>%
        summarize(totalvalue = sum(contract_value) %>%
        ungroup())
```

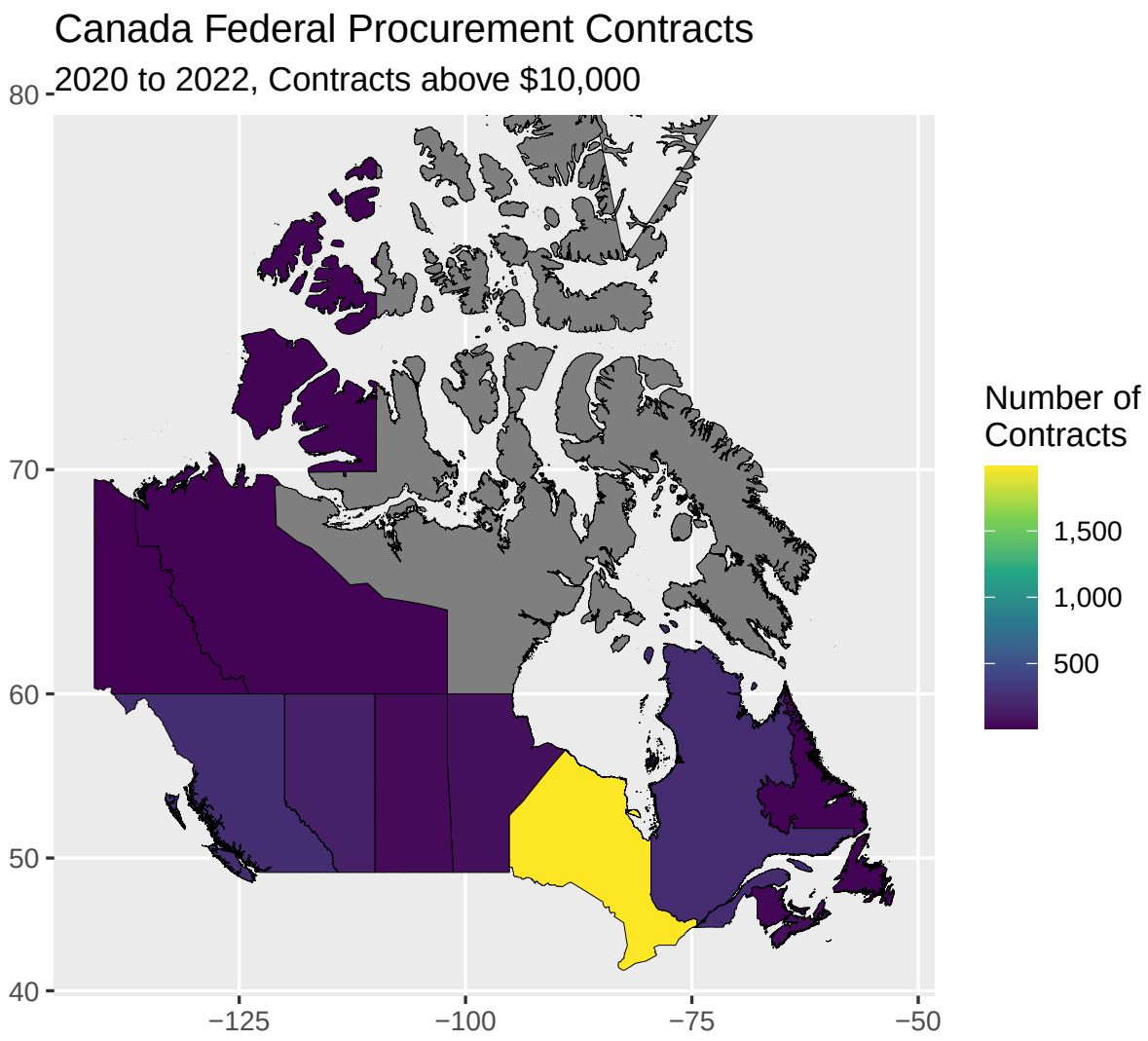
MEMORIAL UNIVERSITY

Contracts Data Set (GADM Data)

```
canada.gadm.data <- left_join(canada.gadm.tidy, d.gadm.id, by='id')

ggplot(canada.gadm.data, aes(x=long,y=lat,group=group))+
  geom_polygon(aes(fill=totalvalue), linewidth=0.1, color='black') +
  coord_map() +
  scale_fill_continuous(type='viridis', labels=scales::comma) +
  guides(x='none', y='none') +
  labs(x='Degrees Longitude', y='Degrees Latitude', fill='Total CAD',
       title='Canada Federal Procurement Contract Values',
       subtitle='2020 to 2022, Contracts above $10,000')
```

MEMORIAL UNIVERSITY

Contracts Data Set (GADM Data)



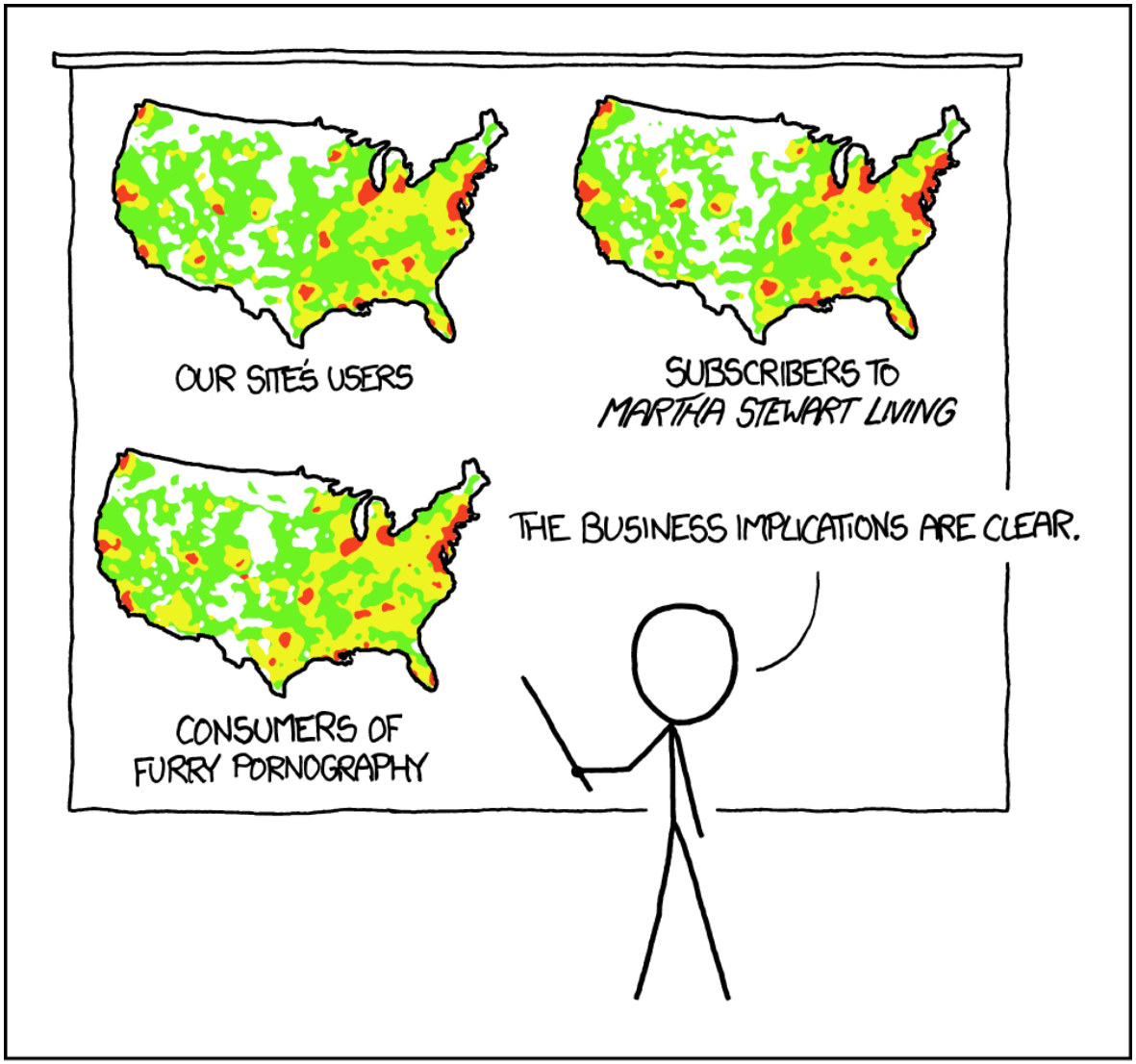
MEMORIAL UNIVERSITY

Contracts Data Set (GADM Data)

```
canada.gadm <- shapefile('gadm41_CAN_1.shp')
canada.gadm.tidy <- tidy(canada.gadm)

...
```

MEMORIAL UNIVERSITY

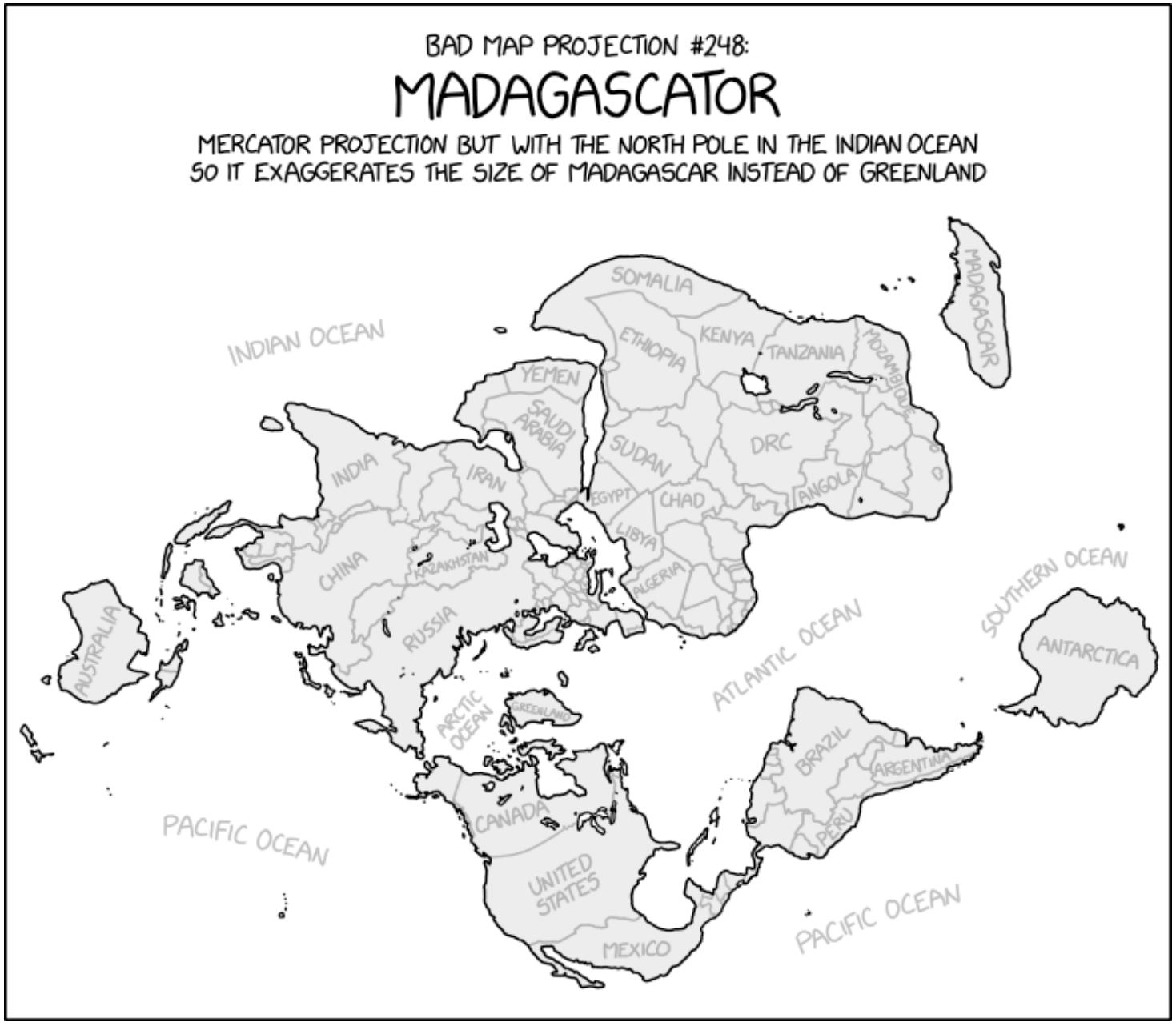
What not do with maps (XKCD)



PET PEEVE #208:

GEOGRAPHIC PROFILE MAPS WHICH ARE BASICALLY JUST POPULATION MAPS

What not do with maps (XKCD)

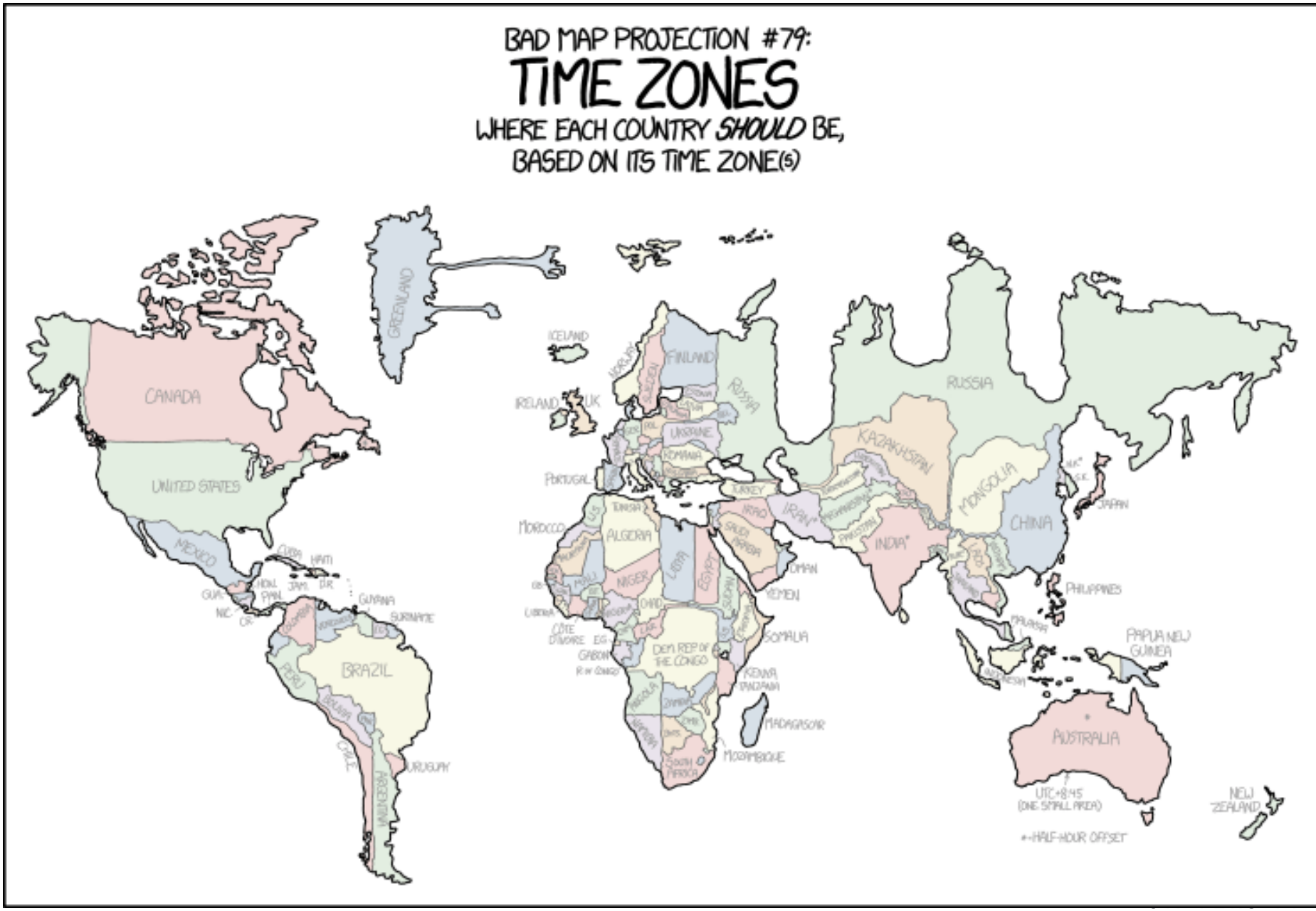


MEMORIAL UNIVERSITY

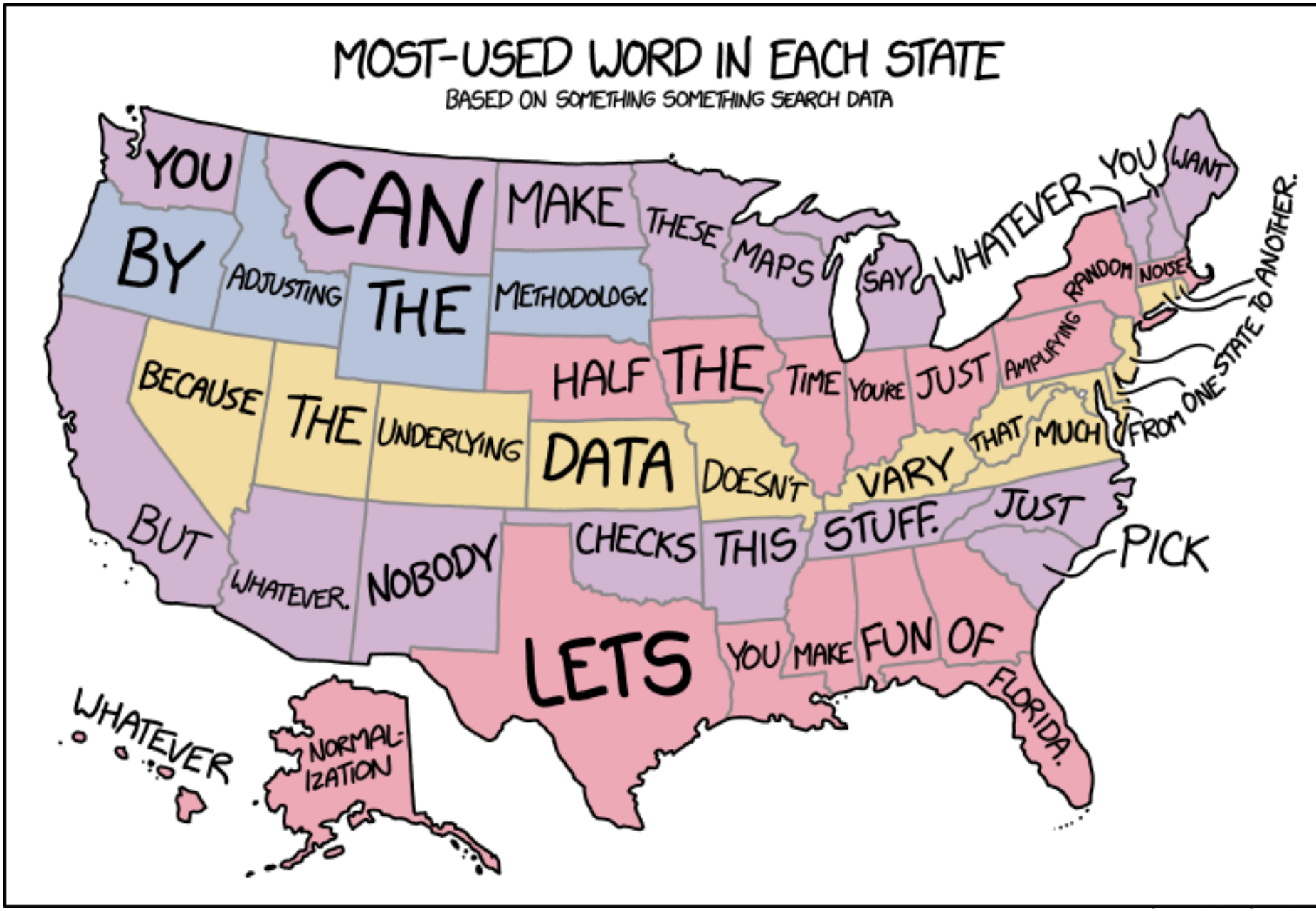
What not do with maps (XKCD)

MEMORIAL UNIVERSITY

What not do with maps (XKCD)



What not do with maps (XKCD)



Using Plotly Express and Plotly Dash

Pre-Processing

```
contractsData = pd.read_csv('contracts.cleaned.csv', low_memory=False)

contractsData['contract_date'] =
   pd.to_datetime(contractsData['contract_date'])
contractsData['contract_period_start'] =
   pd.to_datetime(contractsData['contract_period_start'])
contractsData['economic_object_code'] =
   contractsData['economic_object_code'].astype('category')
contractsData['commodity_type'] =
   contractsData['commodity_type'].astype('category')
contractsData['country_of_vendor'] =
   contractsData['country_of_vendor'].astype('category')
contractsData['solicitation_procedure'] =
   contractsData['solicitation_procedure'].astype('category')
contractsData['limited_tendering_reason'] =
   contractsData['limited_tendering_reason'].astype('category')
contractsData['trade_agreement_exceptions'] =
   contractsData['trade_agreement_exceptions'].astype('category')
contractsData['award_criteria'] =
   contractsData['award_criteria'].astype('category')
```

MEMORIAL UNIVERSITY

Pre-Processing

```
def prov_from_postal(row):
    if str(row['vendor_postal_code']).startswith('A'): return 'NL'
    if str(row['vendor_postal_code']).startswith('B'): return 'NS'
    if str(row['vendor_postal_code']).startswith('C'): return 'PE'
    if str(row['vendor_postal_code']).startswith('E'): return 'NB'
    if str(row['vendor_postal_code']).startswith('G'): return 'QC'
    if str(row['vendor_postal_code']).startswith('J'): return 'QC'
    if str(row['vendor_postal_code']).startswith('L'): return 'ON'
    if str(row['vendor_postal_code']).startswith('M'): return 'ON'
    if str(row['vendor_postal_code']).startswith('N'): return 'ON'
    if str(row['vendor_postal_code']).startswith('K'): return 'ON'
    if str(row['vendor_postal_code']).startswith('P'): return 'ON'
    if str(row['vendor_postal_code']).startswith('R'): return 'MB'
    if str(row['vendor_postal_code']).startswith('S'): return 'SK'
    if str(row['vendor_postal_code']).startswith('T'): return 'AB'
    if str(row['vendor_postal_code']).startswith('V'): return 'BC'
    if str(row['vendor_postal_code']).startswith('X'): return 'NT'
    if str(row['vendor_postal_code']).startswith('Y'): return 'YK'
    return 'UNKWN'

contractsData['vendor_province'] =
   contractsData.apply(prov_from_postal, axis=1)
contractsData['vendor_province'] =
   contractsData['vendor_province'].astype('category')
```

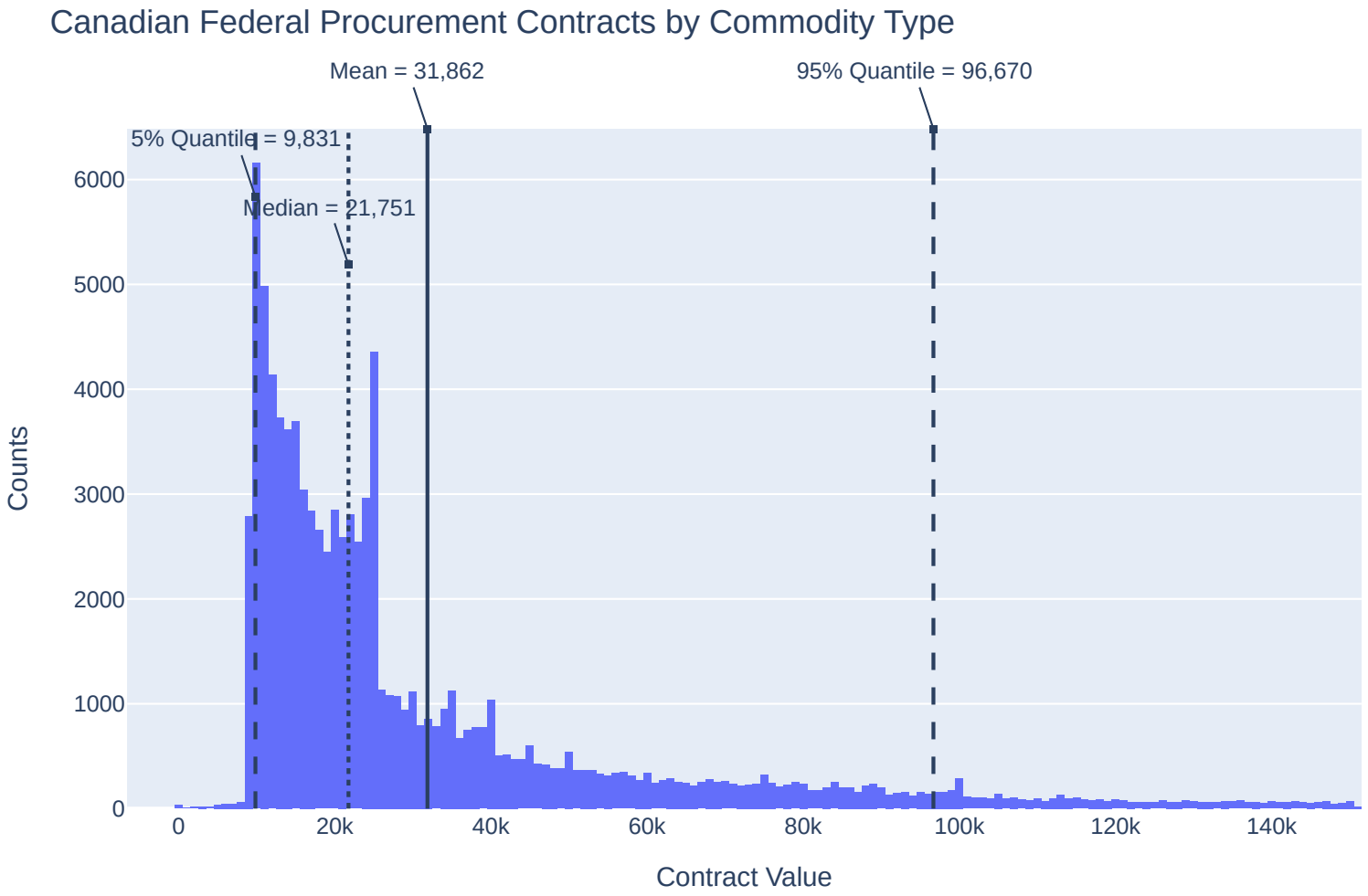
MEMORIAL UNIVERSITY

Pre-Processing

```
def region_from_prov(row):
    if str(row['vendor_province']) in ['NL', 'NS', 'PE', 'NB']: return
    if str(row['vendor_province']) in ['YK', 'NT']: return 'Northern'
    if str(row['vendor_province']) in ['QC', 'ON']: return 'Central'
    if str(row['vendor_province']) in ['AB', 'SK', 'MB']: return 'Prai
    if str(row['vendor_province']) in ['BC']: return 'Western'
    if str(row['vendor_province']) in ['UNKWN']: return 'Unknown'
    return 'Unknown'

contractsData['vendor_region'] =
   contractsData.apply(region_from_prov, axis=1)
contractsData['vendor_region'] =
   contractsData['vendor_region'].astype('category')

contractsData = contractsData[
   (contractsData['original_value'] <
        contractsData['original_value'].quantile(.90)) &
   (contractsData['contract_value'] <
        contractsData['contract_value'].quantile(.90))]
```

MEMORIAL UNIVERSITY

Histogram



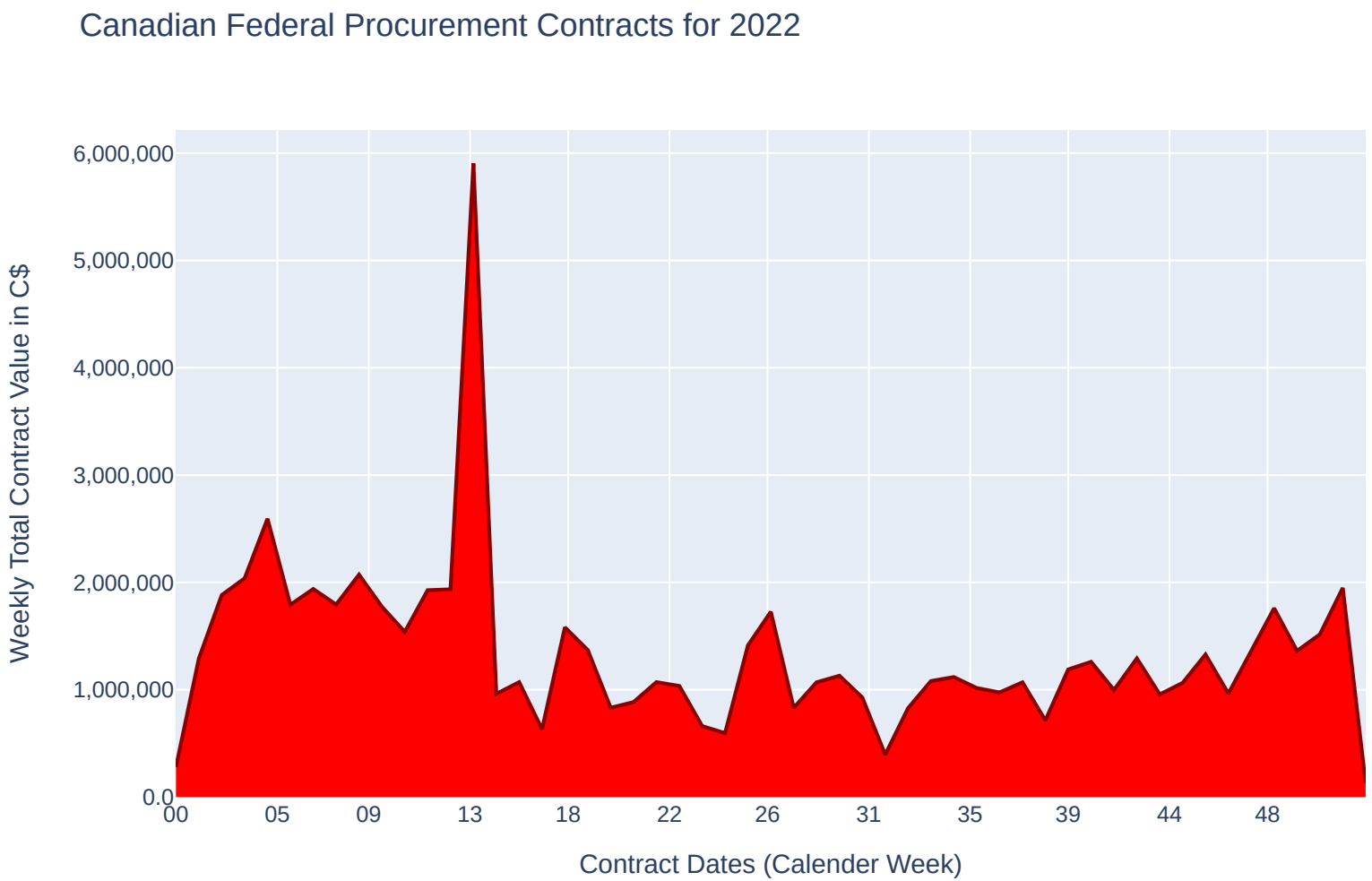
MEMORIAL UNIVERSITY

Histogram

```
fig = px.histogram(
          data_frame = df,
          x = 'contract_value',
          histfunc = 'count',
          title = 'Canadian Federal Procurement Contracts by Commodity
          labels = {'contract_value': 'Contract Value', 'y': 'Count'})
fig.add_vline(x = df.contract_value.mean(), line_dash='solid')
fig.add_vline(x = df.contract_value.median(), line_dash='dot')
fig.add_vline(x = df.contract_value.quantile(.05), line_dash='dash')
fig.add_vline(x = df.contract_value.quantile(.95), line_dash='dash')
fig.add_annotation(
             x=df.contract_value.mean(),
             y=1,
             xref='x',
             yref='paper',
             text="Mean = {:,.0f}".format(df.contract_value.mean()),
             showarrow=True,
             arrowhead=7)

...

fig.update_layout(yaxis_title='Counts')
```

MEMORIAL UNIVERSITY

Area – One Series



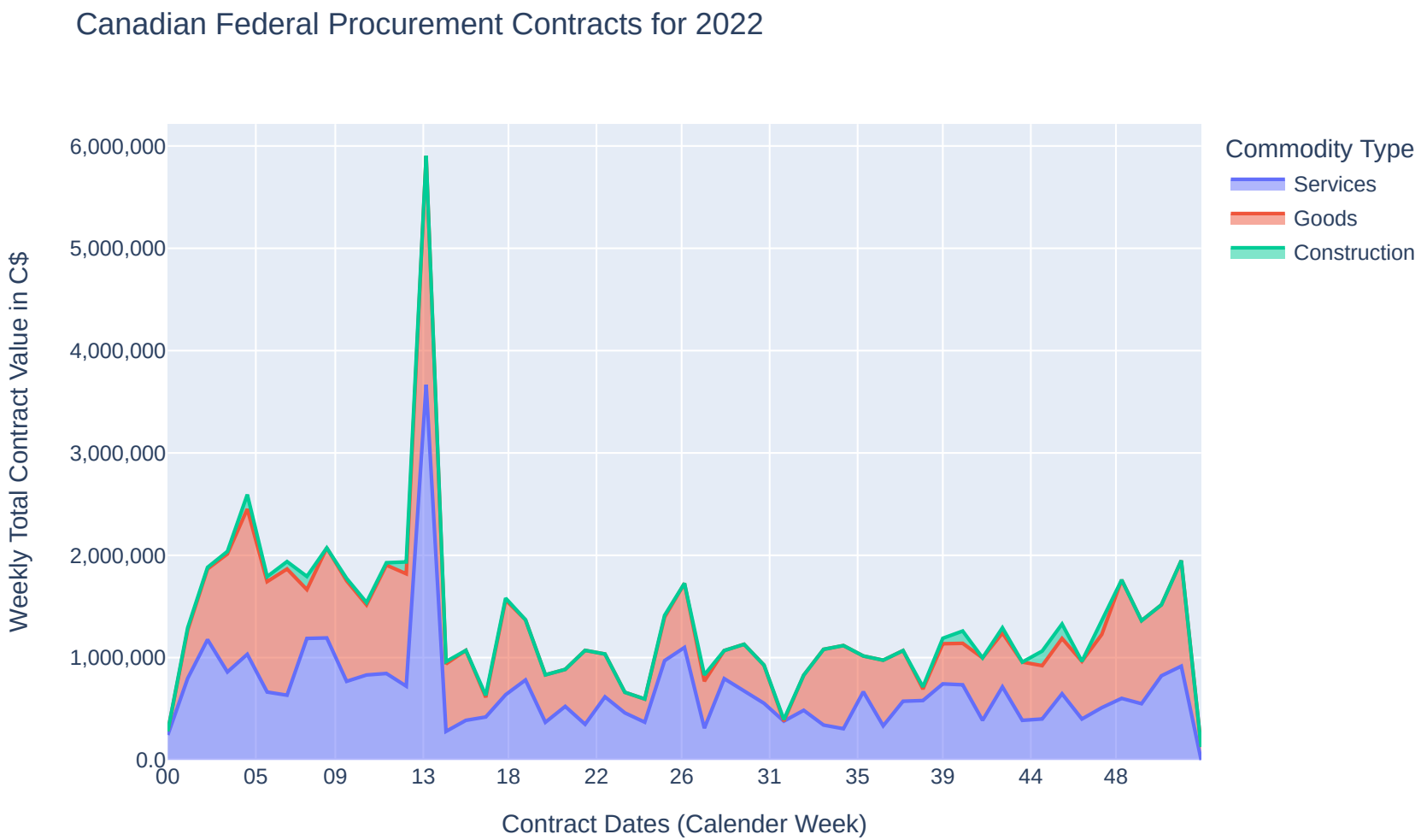
MEMORIAL UNIVERSITY

Area – One Series

```
df = contractsData[
      (contractsData['contract_date'] >= pd.to_datetime('2022-01-01'))
      (contractsData['contract_date'] <= pd.to_datetime('2022-12-31'))
df = df.groupby(pd.Grouper(key='contract_date', axis=0, freq='W'))
      ['contract_value'].sum().reset_index()

fig = px.area(
          data_frame = df,
          x = 'contract_date',
          y = 'contract_value',
          title = 'Canadian Federal Procurement Contracts for 2022',
          labels = {'contract_value': 'Weekly Total Contract Value in
                    'contract_date': 'Contract Dates (Calender Week)'}
fig.update_xaxes(dtick="M1", tickformat='%W', tick0='2022-01-02')
fig.update_yaxes(tickformat=',.2r')
fig.update_traces(fillcolor='red', line=dict(color='darkred'))
```

MEMORIAL UNIVERSITY

Area – Three Series



MEMORIAL UNIVERSITY

Area – Three Series

```
df = contractsData[
      (contractsData['contract_date'] >= pd.to_datetime('2022-01-01'))
      (contractsData['contract_date'] <= pd.to_datetime('2022-12-31'))
df = df.groupby([
          pd.Grouper(key='contract_date', axis=0, freq='W'),
          pd.Grouper(key='commodity_type', axis=0)])
      ['contract_value'].sum().reset_index()
df['commodity_type'] = df['commodity_type'].astype(str)

fig = px.area(
        data_frame = df,
        x = 'contract_date',
        y = 'contract_value',
        color = 'commodity_type',
        title = 'Canadian Federal Procurement Contracts for 2022',
        labels = {'contract_value': 'Weekly Total Contract Value in C$
                  'contract_date': 'Contract Dates (Calender Week)',
                  'commodity_type': 'Commodity Type'})
fig.update_xaxes(dtick="M1", tickformat='%W', tick0='2022-01-02')
fig.update_yaxes(tickformat=',.2r')
```

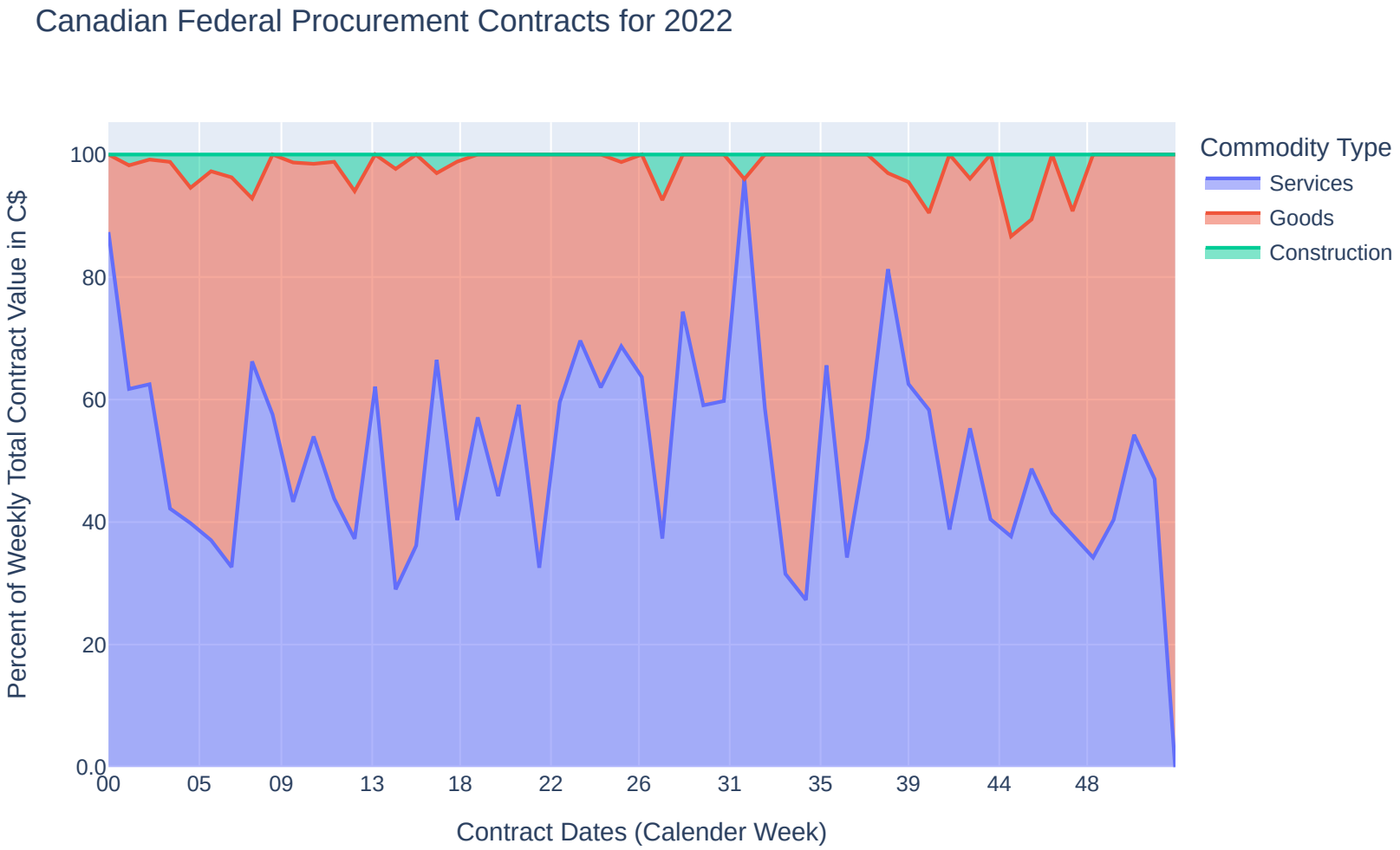
MEMORIAL UNIVERSITY

Area – Three Series

```
newnames = {'S':'Services', 'G': 'Goods', 'C': 'Construction'}
fig.for_each_trace(lambda t: t.update(
                name = newnames[t.name],
                legendgroup = newnames[t.name],
                hovertemplate =
                   t.hovertemplate.replace(t.name, newnames[t.name])
                                       ))
```

MEMORIAL UNIVERSITY

Area – Three Series, Normalized



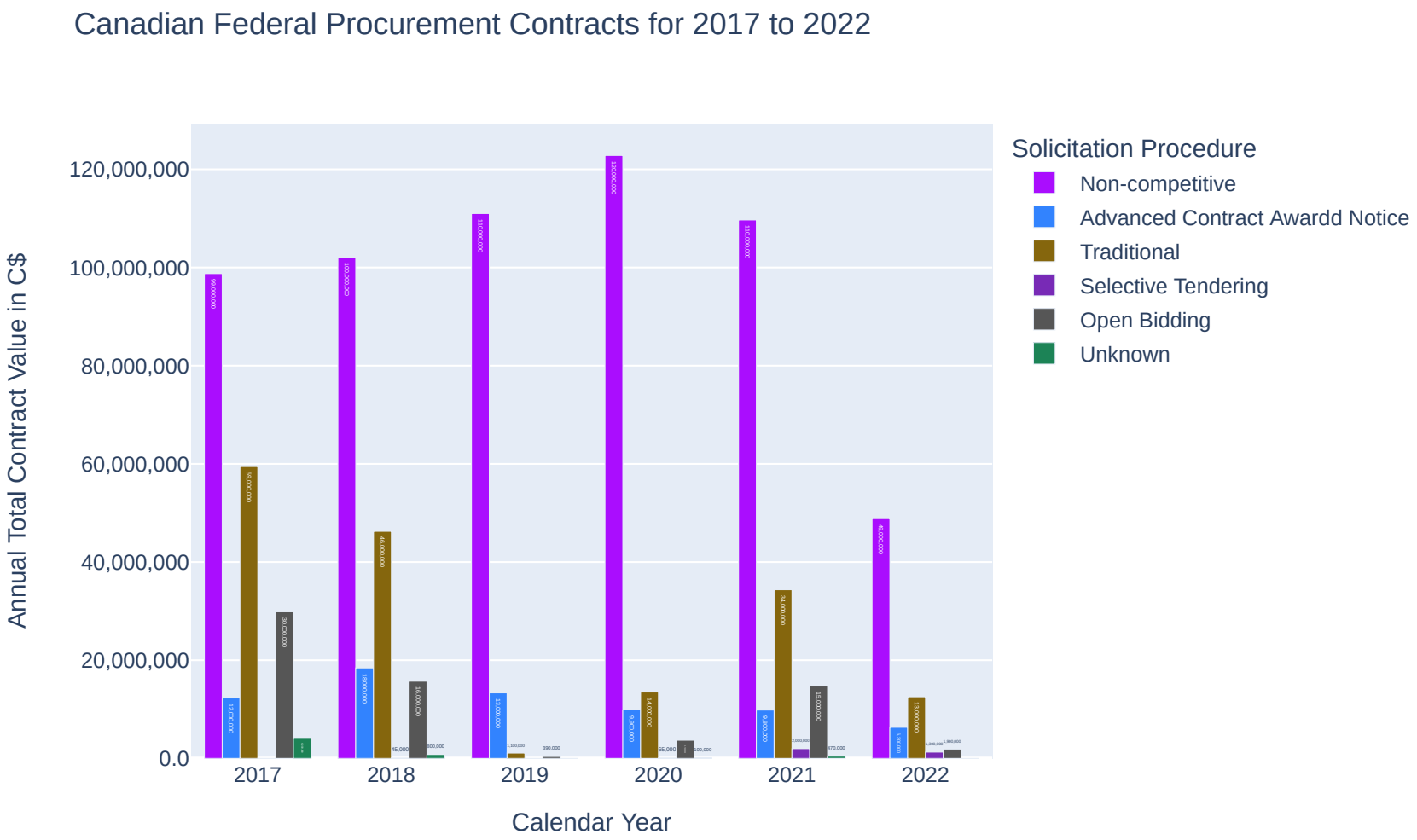
MEMORIAL UNIVERSITY

Area – Three Series, Normalized

```
...
fig = px.area(data_frame = df,
              x = 'contract_date',
              y = 'contract_value',
              color = 'commodity_type',
              title = 'Canadian Federal Procurement Contracts for 2022',
              groupnorm = 'percent',
              labels = {'contract_value': 'Percent of Weekly Total Contra
                 'contract_date': 'Contract Dates (Calender Week)',
                 'commodity_type': 'Commodity Type'})

...
```

MEMORIAL UNIVERSITY

Bar – Three Series, Grouped



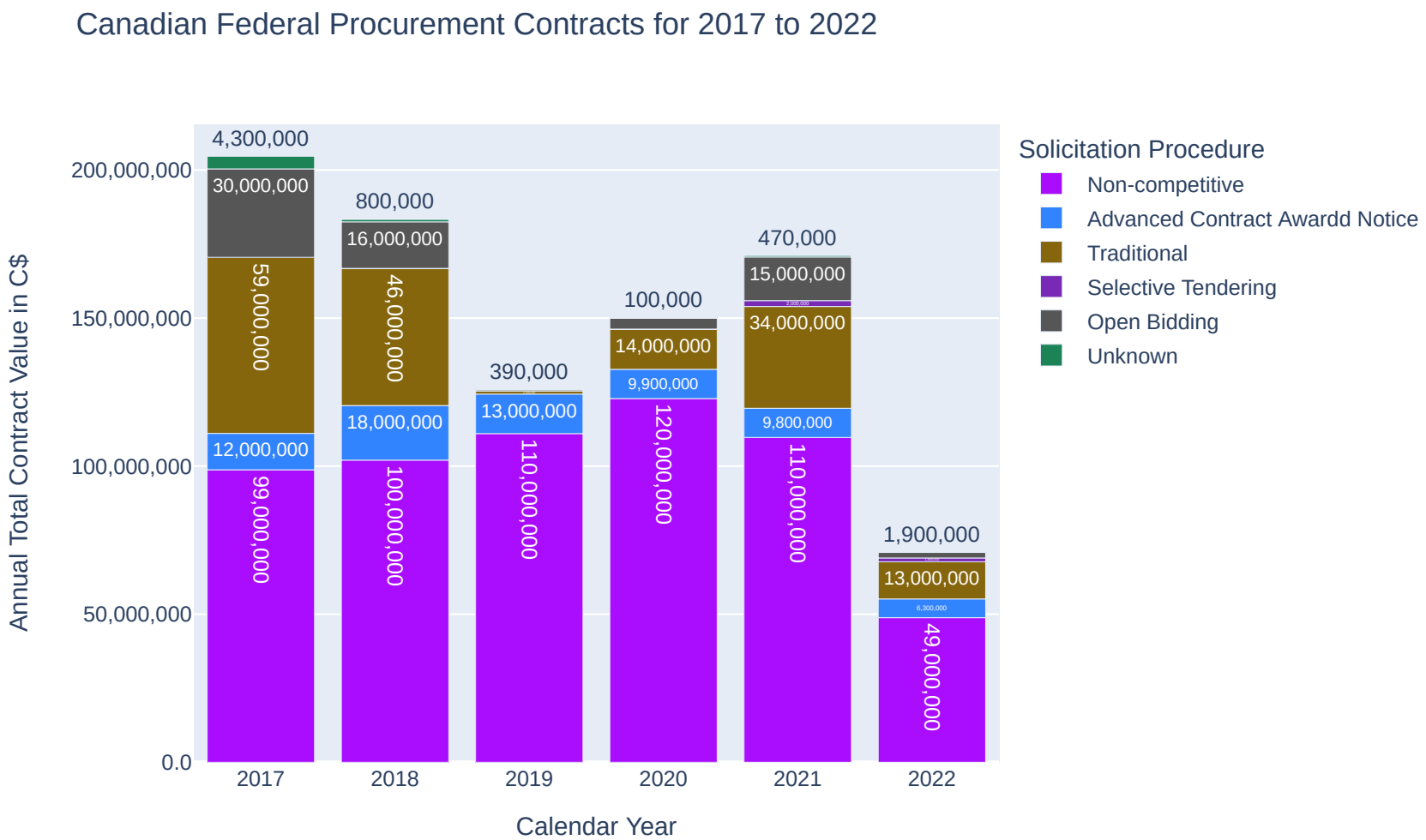
MEMORIAL UNIVERSITY

Bar – Three Series, Grouped

```
...

fig = px.bar(
        data_frame = df,
        x = 'commodity_type',
        y = 'contract_value',
        color = 'solicitation_procedure',
        barmode = 'group',
        title = 'Canadian Federal Procurement Contracts for 2017 to 20
        text_auto=True,
        color_discrete_sequence = px.colors.qualitative.Alphabet,
        labels = {'contract_value': 'Annual Total Contract Value in C$
            'year': 'Calendar Year',
            'solicitation_procedure': 'Solicitation Procedure'})

...
```

MEMORIAL UNIVERSITY

Bar – Three Series, Stacked

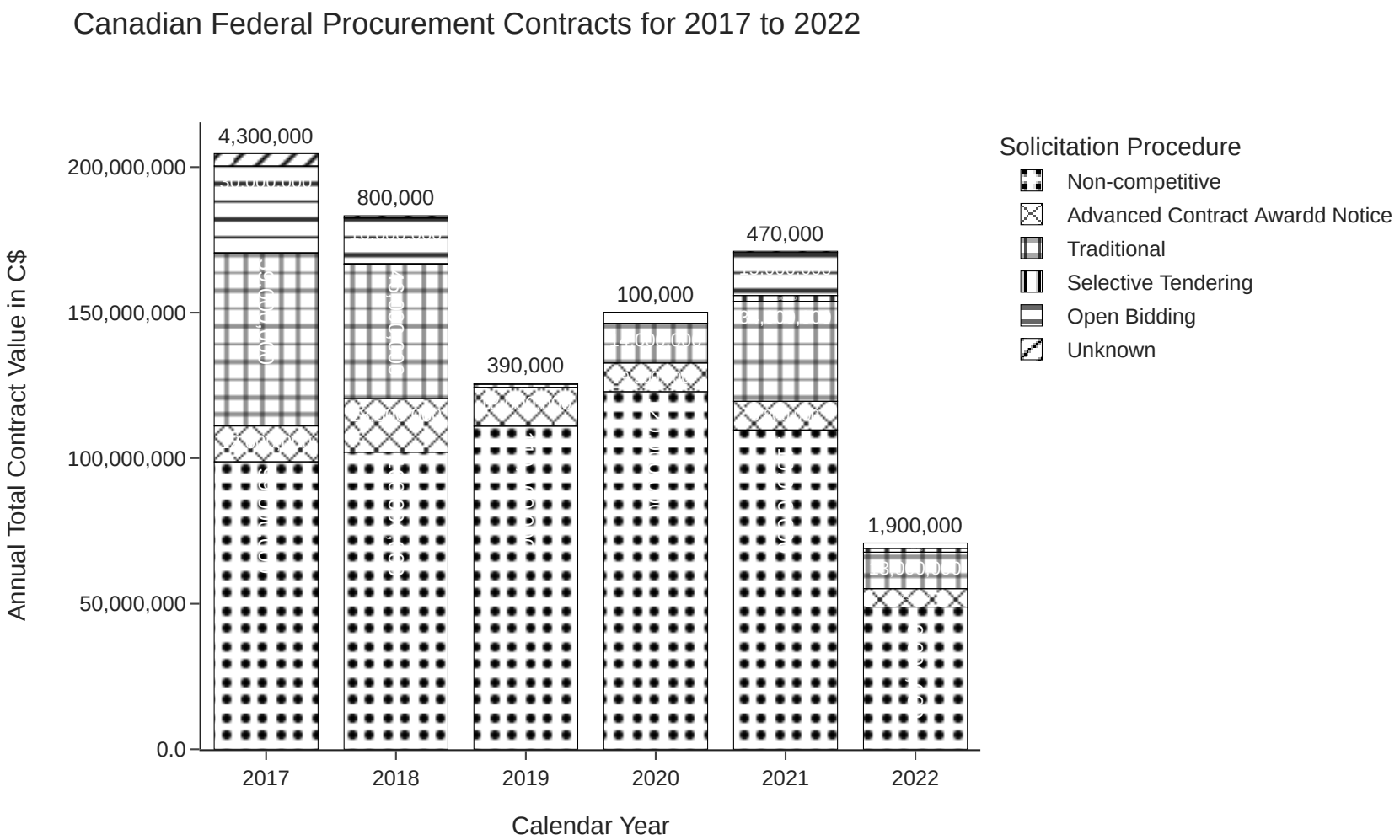


MEMORIAL UNIVERSITY

Bar – Three Series, Stacked

```
fig.update_layout(barmode = 'stack')
```

Bar – Three Series, Pattern

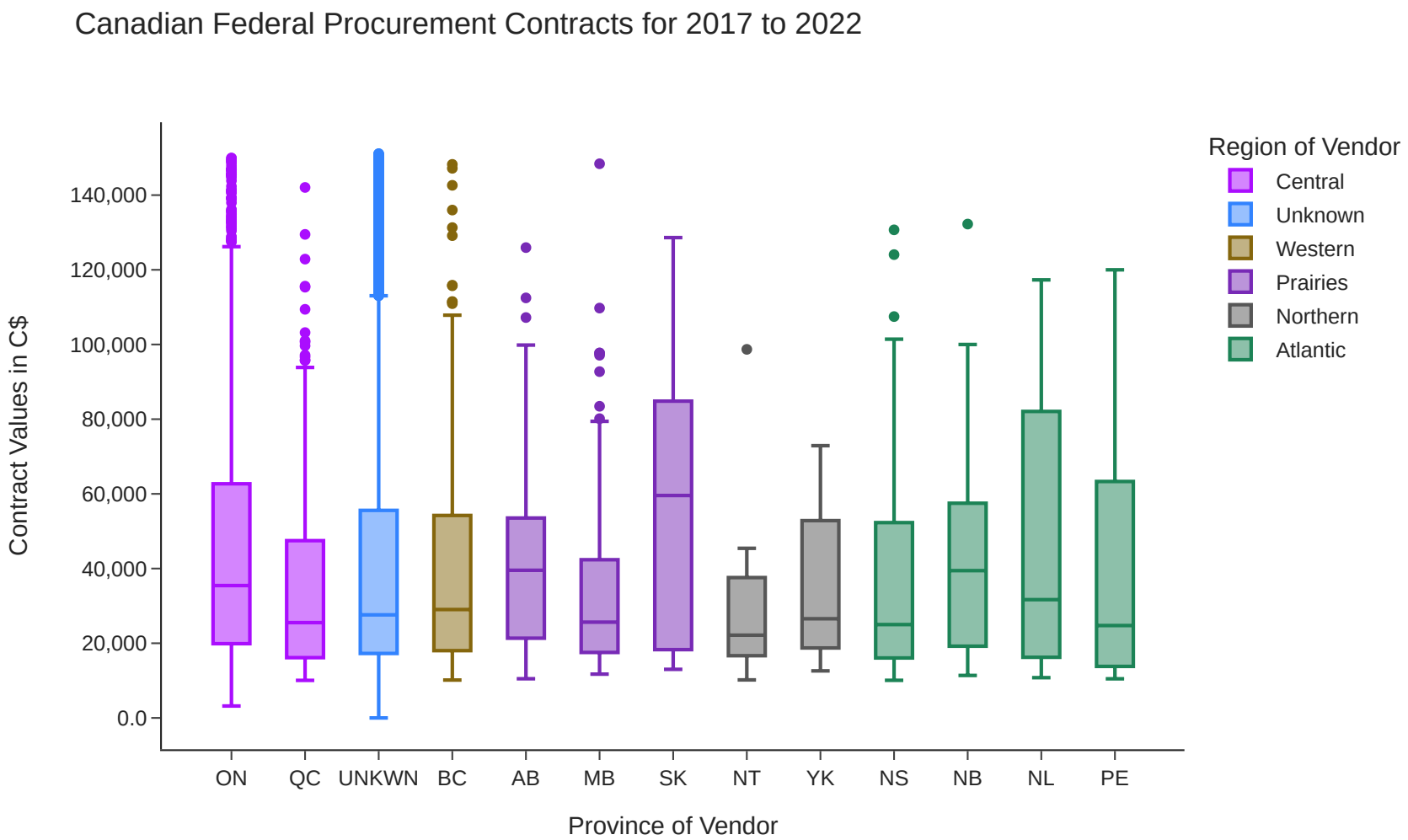


Bar – Three Series, Pattern

```
...

fig = px.bar(data_frame = df,
             x = 'year',
             y = 'contract_value',
             pattern_shape = 'solicitation_procedure',
             pattern_shape_sequence = ['.', 'x', '+', '|', '-', '/'],
             title = 'Canadian Federal Procurement Contracts for 2017 to
             text_auto=True,
             template="simple_white",
             labels = {
               'contract_value': 'Annual Total Contract Value in C$',
               'year': 'Calendar Year',
               'solicitation_procedure': 'Solicitation Procedure'})

fig.update_traces(marker=dict(color='black',
                              line_color='black',
                              pattern_fillmode='replace'))
```

MEMORIAL UNIVERSITY

Box Plot



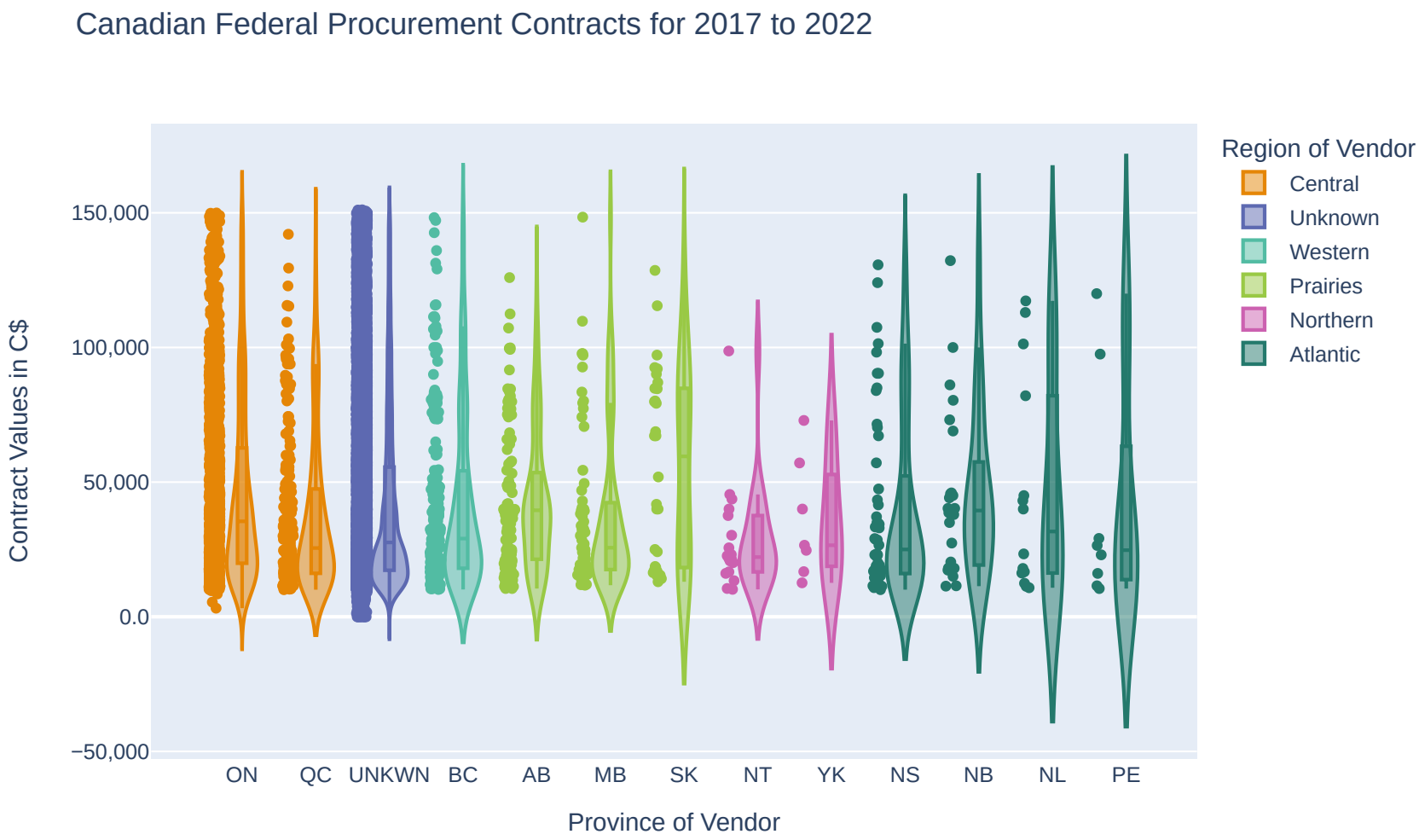
MEMORIAL UNIVERSITY

Box Plot

```
...

fig = px.box(data_frame = df,
             x = 'vendor_province',
             y = 'contract_value',
             color = 'vendor_region',
             title = 'Canadian Federal Procurement Contracts for 2017 to
             color_discrete_sequence=px.colors.qualitative.Alphabet,
             template="presentation",
             labels = {'contract_value': 'Contract Values in C$',
                       'vendor_province': 'Province of Vendor',
                       'vendor_region': 'Region of Vendor'})

fig.update_yaxes(tickformat=',.2r')
fig.update_traces(width = 0.5)
```

MEMORIAL UNIVERSITY

Violin Plot



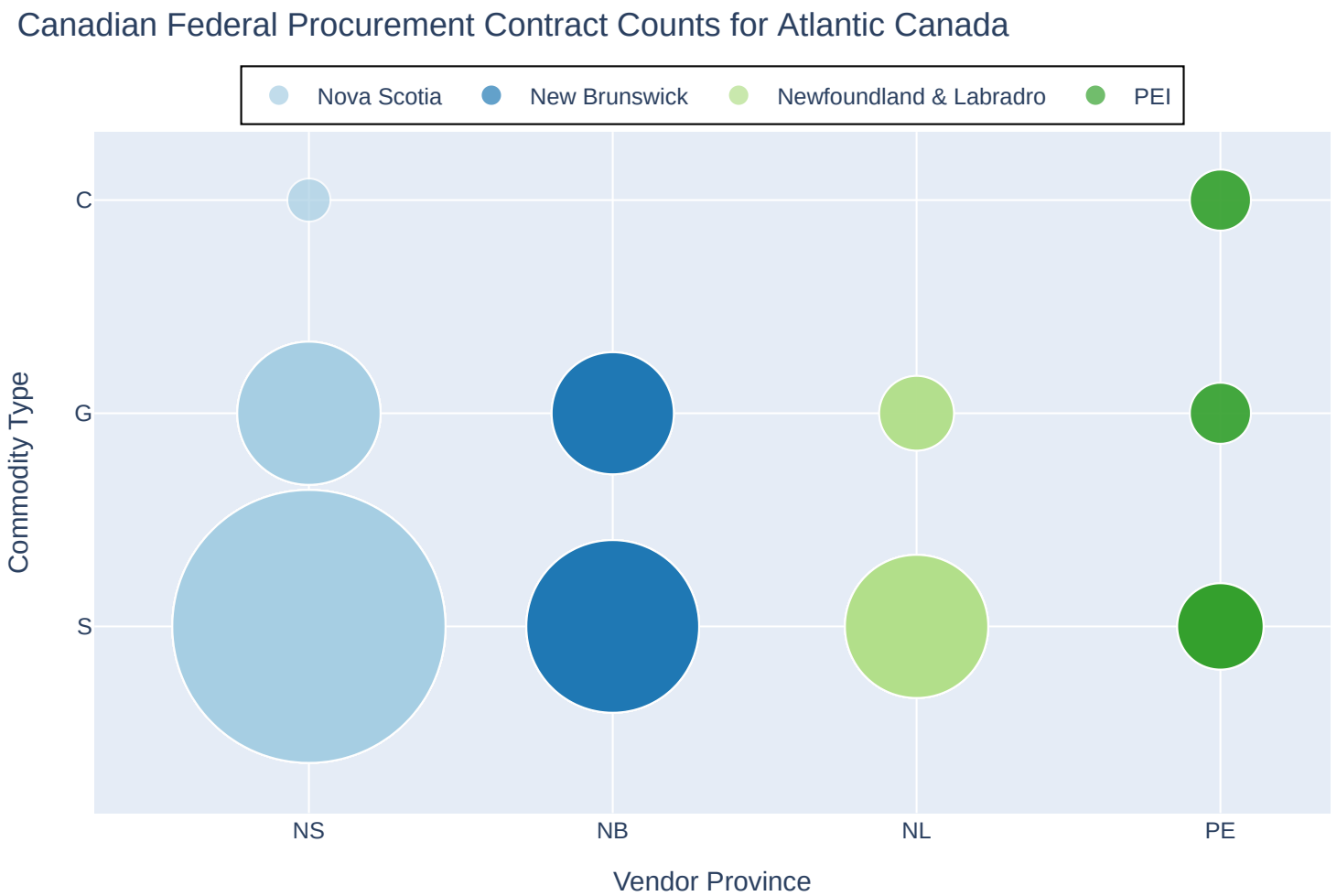
MEMORIAL UNIVERSITY

Violin Plot

```
fig = px.violin(data_frame = df,
          x = 'vendor_province',
          y = 'contract_value',
          color = 'vendor_region',
          points = 'all',
          title = 'Canadian Federal Procurement Contracts for 2017 to
          color_discrete_sequence=px.colors.qualitative.Vivid,
          box = True,
          labels = {'contract_value': 'Contract Values in C$',
             'vendor_province': 'Province of Vendor',
             'vendor_region': 'Region of Vendor'})

fig.update_yaxes(tickformat=',.2r')
fig.update_traces(width = 0.5)
```

MEMORIAL UNIVERSITY

Counts/Bubbles



MEMORIAL UNIVERSITY

Counts/Bubblest

```
df = contractsData[
      (contractsData['contract_date'] >= pd.to_datetime('2017-01-01'))
      (contractsData['contract_date'] <= pd.to_datetime('2022-12-31'))
      (contractsData['vendor_province'] != 'UNKWN') &
      (contractsData['vendor_region'] == 'Atlantic')]
df['count'] = df['vendor_province']
              .groupby([df['vendor_province'], df['commodity_type']])
              .transform('count')

df['commodity_type'] = df['commodity_type'].astype(str)
df['vendor_region'] = df['vendor_region'].astype(str)
df['vendor_province'] = df['vendor_province'].astype(str)
```

MEMORIAL UNIVERSITY

Counts/Bubblest

```
fig = px.scatter(df,
            x='vendor_province',
            y='commodity_type',
            color='vendor_province',
            size='count',
            size_max=100,
            color_discrete_sequence=px.colors.colorbrewer.Paired,
            labels={'vendor_province': 'Vendor Province',
                    'commodity_type': 'Commodity Type',
                    'count': 'Count'},
            title='Canadian Federal Procurement Contract Counts for Atla
fig.update_layout(
      legend=dict(orientation="h",
                  y=1.01, yanchor="bottom",
                  x=0.5, xanchor="center"),
      legend_title_text=None,
      legend_bgcolor="white",
      legend_bordercolor="black",
      legend_borderwidth=1)
```

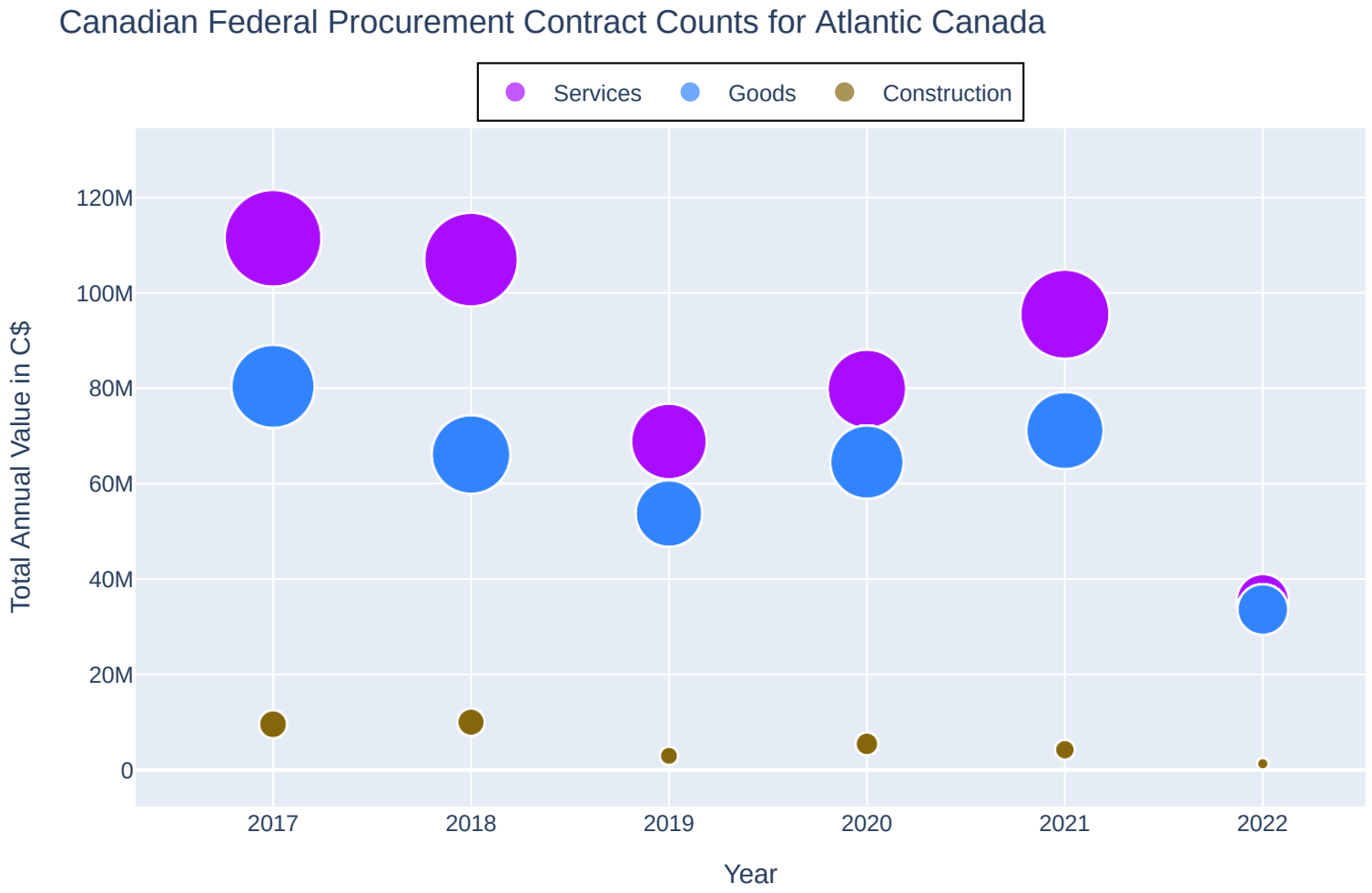
MEMORIAL UNIVERSITY

Counts/Bubblest

```
newnames = {'G': 'Goods', 'S':'Services', 'C': 'Construction',
            'NS': 'Nova Scotia', 'NB': 'New Brunswick',
            'NL': 'Newfoundland & Labradro', 'PE': 'PEI'}

fig.for_each_trace(lambda t:
        t.update(name = newnames[t.name],
                 legendgroup = newnames[t.name],
                 hovertemplate =
                    t.hovertemplate.replace(t.name, newnames[t.name])))

fig.update_layout(legend= {'itemsizing': 'constant'})
fig.update_yaxes(categoryorder="array",
                 categoryarray=['Construction', 'Goods', 'Services'])
```

MEMORIAL UNIVERSITY

Bubble Chart



MEMORIAL UNIVERSITY

Bubble Chart

```
df = contractsData[
        (contractsData['contract_date'] >= pd.to_datetime('2017-01-01')
        (contractsData['contract_date'] <= pd.to_datetime('2022-12-31')
        (contractsData['commodity_type'] != 'UNKWN')]

df['year'] = df['contract_date'].astype(str).str[0:4]
df['year'] = df['year'].astype(float)
df['count'] = df['vendor_province']
              .groupby([df['commodity_type'], df['year']])
              .transform('count')
df['total_value'] = df['contract_value']
                    .groupby([df['commodity_type'], df['year']])
                    .transform('sum')

df['commodity_type'] = df['commodity_type'].astype(str)
df['vendor_region'] = df['vendor_region'].astype(str)
df['vendor_province'] = df['vendor_province'].astype(str)
```

MEMORIAL UNIVERSITY

Bubble Chart

```
fig = px.scatter(df,
            x='year',
            y='total_value',
            color='commodity_type',
            size='count',
            size_max=50,
            color_discrete_sequence=px.colors.qualitative.Alphabet,
            labels={'year': 'Year', 'commodity_type': 'Commodity Type',
                    'count': 'Count', 'total_value': 'Total Annual Valu
            title='Canadian Federal Procurement Contract Counts for Atl

fig.update_layout(legend=dict(orientation="h",
                              y=1.01, yanchor="bottom",
                              x=0.5, xanchor="center"),
                  legend_title_text=None,
                  legend_bgcolor="white",
                  legend_bordercolor="black",
                  legend_borderwidth=1)
```

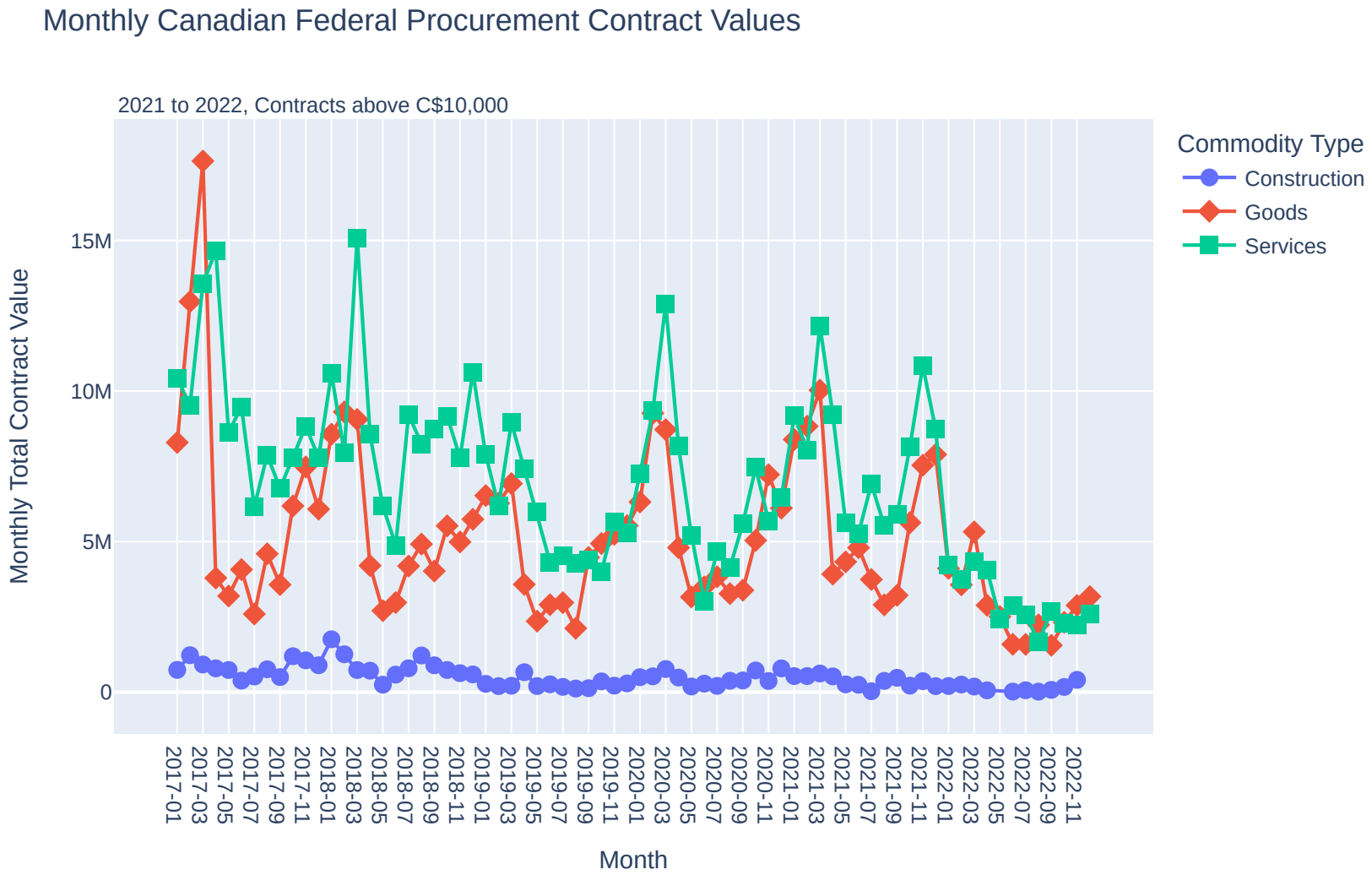
MEMORIAL UNIVERSITY

Bubble Chart

```
newnames = {'G': 'Goods', 'S':'Services', 'C': 'Construction',
            'NS': 'Nova Scotia', 'NB': 'New Brunswick',
            'NL': 'Newfoundland & Labradro', 'PE': 'PEI'}
fig.for_each_trace(lambda t:
       t.update(name = newnames[t.name],
                legendgroup = newnames[t.name],
                hovertemplate =
                   t.hovertemplate.replace(t.name, newnames[t.name])
       ) )

fig.update_layout(legend={'itemsizing': 'constant'})
```

MEMORIAL UNIVERSITY

Line Chart



MEMORIAL UNIVERSITY

Line Chart

```
...

grouped_df = df.groupby(['contract_month', 'commodity_type'])
                .agg(totalcount=('contract_value', 'count'),
                     totalvalue=('contract_value', 'sum'))
                .reset_index()

fig = px.line(grouped_df,
              x='contract_month',
              y='totalvalue',
              color='commodity_type',
              symbol='commodity_type',
              labels={'contract_month': 'Month',
                      'totalvalue': 'Monthly Total Contract Value',
                      'commodity_type': 'Commodity Type'},
              title='Monthly Canadian Federal Procurement Contract Val
```

MEMORIAL UNIVERSITY

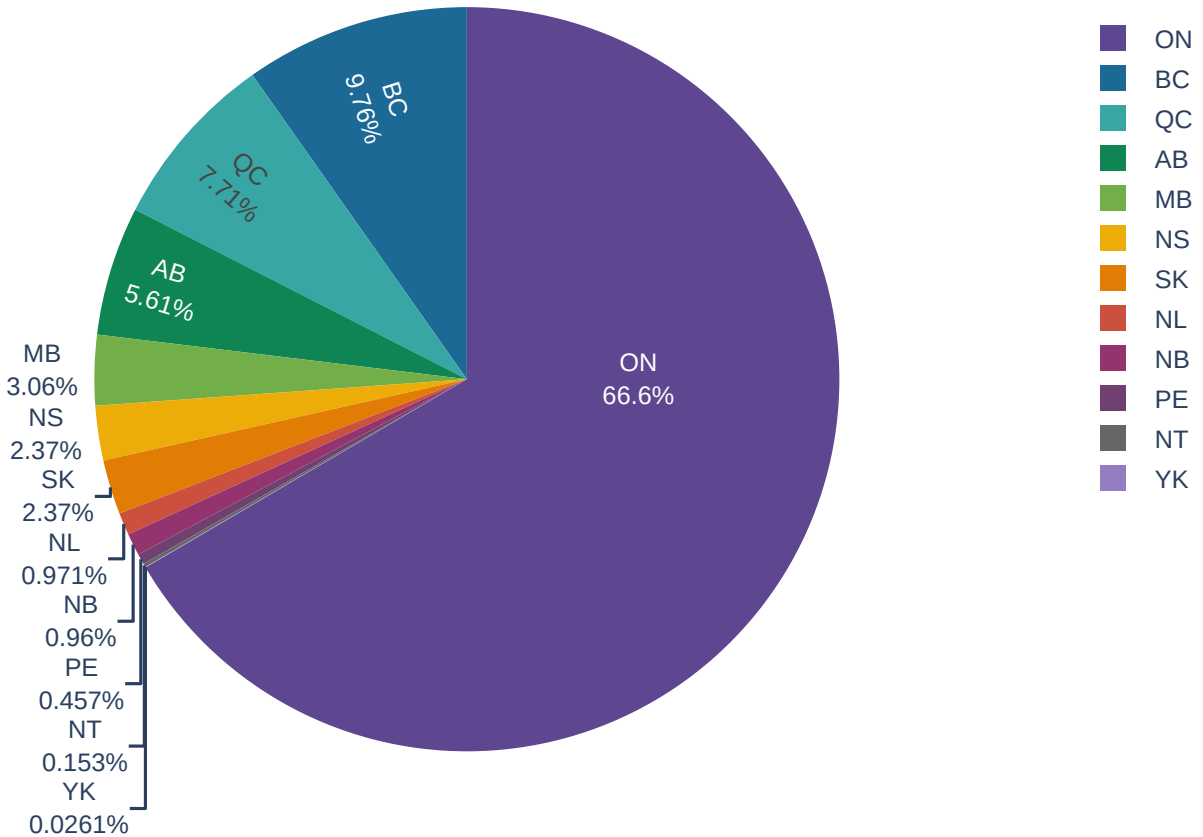
Line Chart

```
fig.update_xaxes(type='category',
                 categoryorder='category ascending',
                 tickformat="%b-%y")

fig.update_traces(marker=dict(size=10))

fig.for_each_trace(lambda trace:
      trace.update(name='Services' if trace.name == 'S' else
                   'Construction' if trace.name == 'C' else 'Goods'))

fig.add_annotation(text='2021 to 2022, Contracts above C$10,000',
                   showarrow=False,
                   xref='paper', x=0, xanchor='left',
                   yref='paper', y=1.05)
```

MEMORIAL UNIVERSITY

Pie Chart

2022 Canadian Federal Contract Award Values by Province



MEMORIAL UNIVERSITY

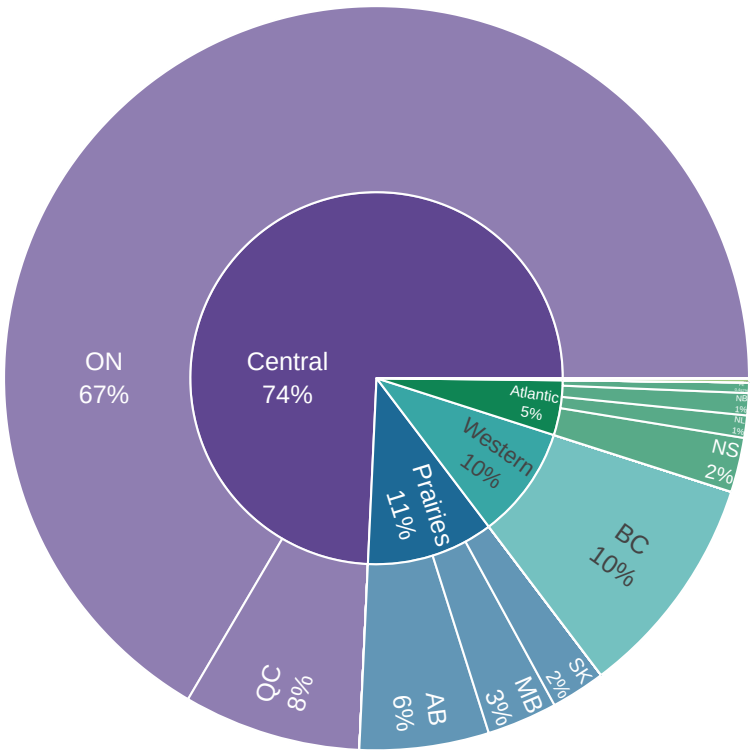
Pie Chart

```
grouped_df = df.groupby(['vendor_province'])
               .agg(totalvalue=('contract_value', 'sum'))
               .reset_index()

fig = px.pie(data_frame=grouped_df,
             values='totalvalue',
             names='vendor_province',
             color_discrete_sequence=px.colors.qualitative.Prism,
             labels={'vendor_province': 'Province of Vendor',
                     'totalvalue': '2022 Annual Total Contract Value'}
             title = '2022 Canadian Federal Contract Award Values by P

fig.update_traces(textinfo='percent+label',
                  insidetextorientation='radial')
```

MEMORIAL UNIVERSITY

Sunburst Chart

2022 Canadian Federal Contract Award Values by Province



MEMORIAL UNIVERSITY

Sunburst Chart

```
grouped_df = df.groupby(['vendor_region', 'vendor_province'])
               .agg(totalvalue=('contract_value', 'sum'))
               .reset_index()

fig = px.sunburst(data_frame=grouped_df,
             path=['vendor_region', 'vendor_province'],
             values='totalvalue',
             names='vendor_province',
             color_discrete_sequence=px.colors.qualitative.Prism,
             labels={'vendor_province': 'Province of Vendor',
                     'totalvalue': '2022 Annual Total Contract Value'}
             title = '2022 Canadian Federal Contract Award Values by P

fig.update_traces(textinfo='label+percent entry',
                  insidetextorientation='radial')
```

MEMORIAL UNIVERSITY

Treemap Chart

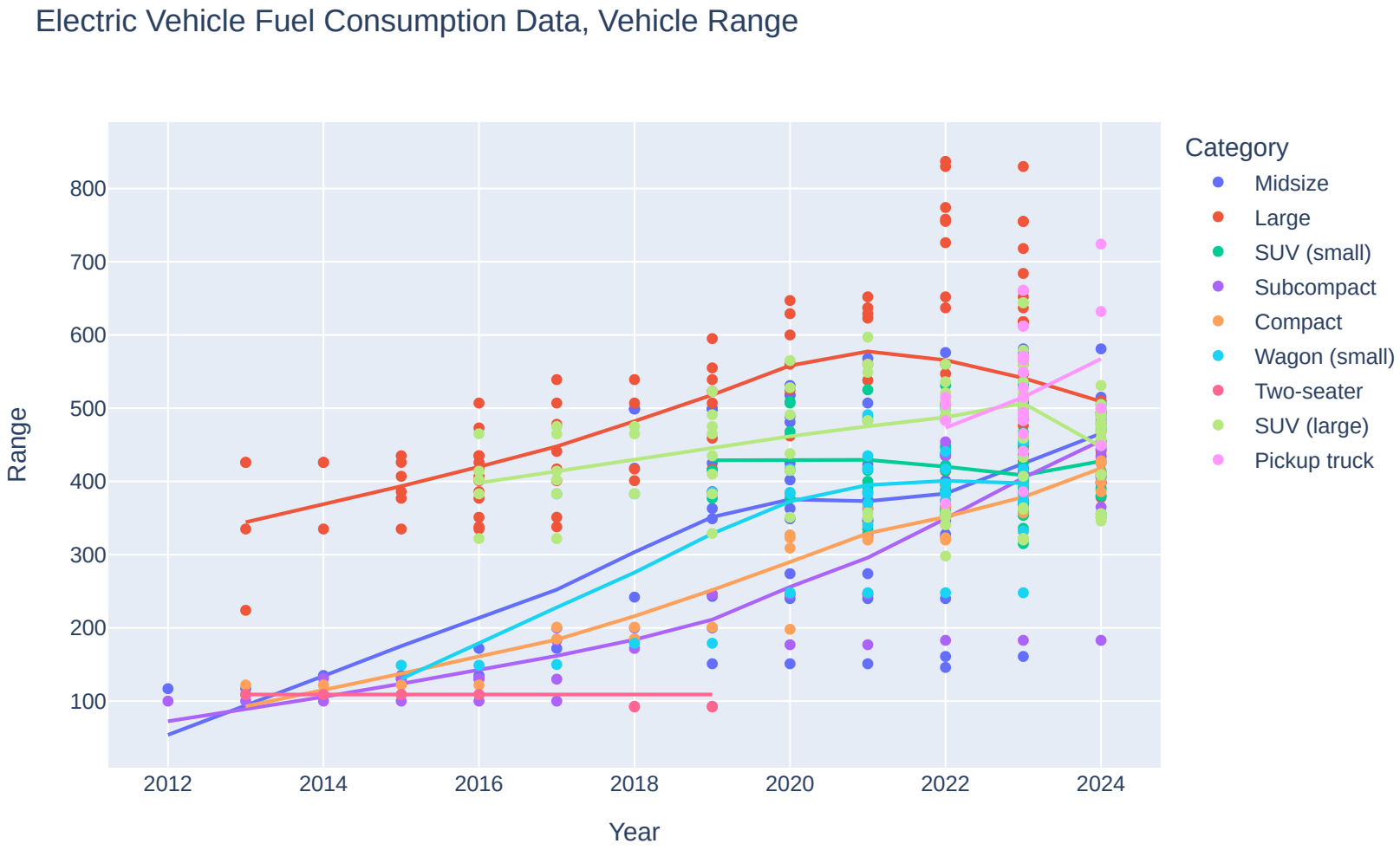
2022 Canadian Federal Contract Award Values by Province

Treemap Chart

```
grouped_df = df.groupby(['vendor_region', 'vendor_province'])
               .agg(totalvalue=('contract_value', 'sum'))
               .reset_index()

fig = px.treemap(data_frame=grouped_df,
           path=[px.Constant('Canada'),
                 'vendor_region',
                 'vendor_province'],
           values='totalvalue',
           names='vendor_province',
           color_discrete_sequence=px.colors.qualitative.Prism,
           labels={'vendor_province': 'Province of Vendor',
                   'totalvalue': '2022 Annual Total Contract Value'},
           title = '2022 Canadian Federal Contract Award Values by Pro

fig.update_traces(textinfo='label+percent entry')
```

MEMORIAL UNIVERSITY

Trendline Plot



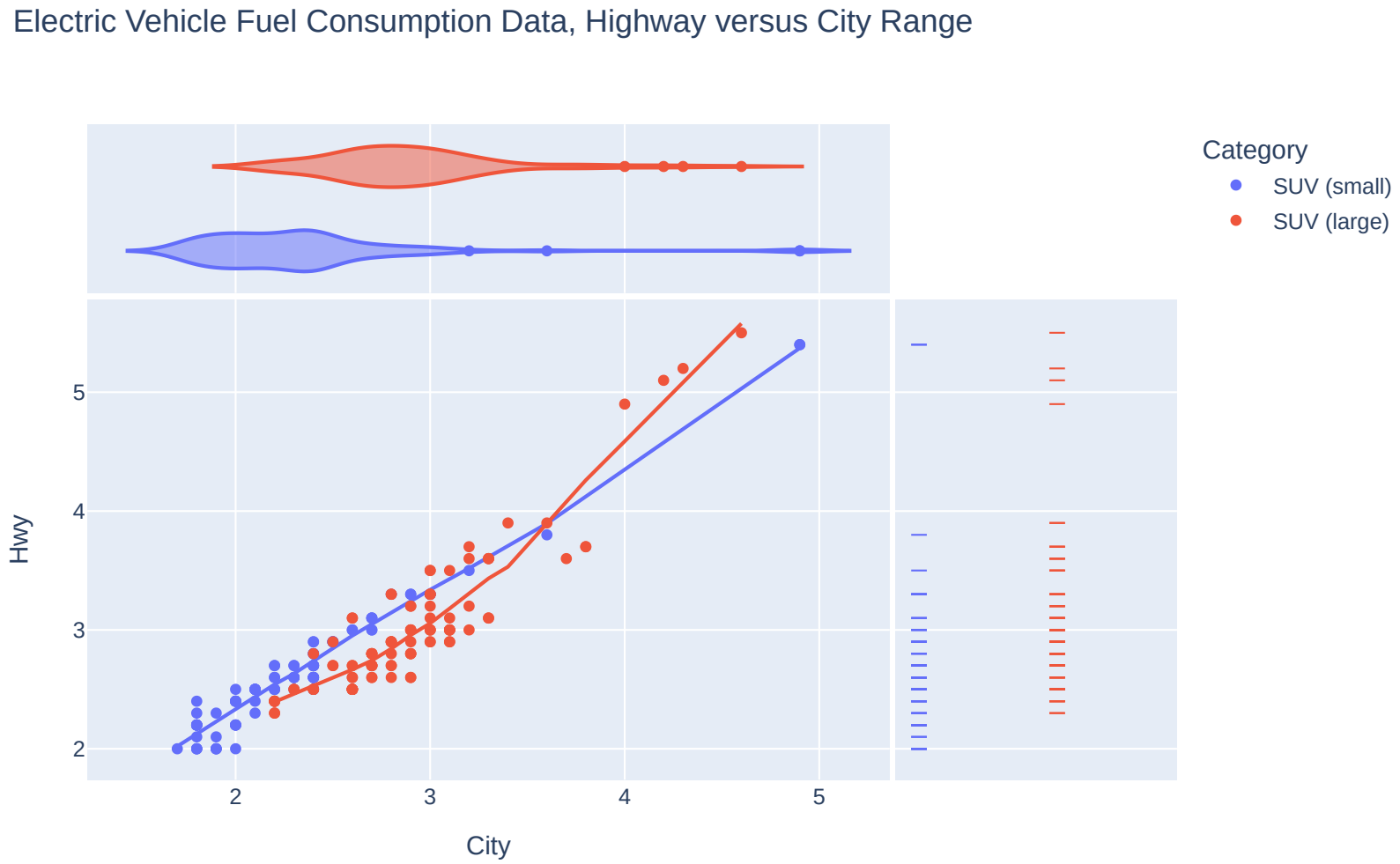
MEMORIAL UNIVERSITY

Trendline Plot

```
fig = px.scatter(fuelData,
            x='Year',
            y='Range',
            color='Category',
            title='Electric Vehicle Fuel Consumption Data, Vehicle Rang
            trendline='lowess',
            trendline_options=dict(frac=0.6666))

...
```

MEMORIAL UNIVERSITY

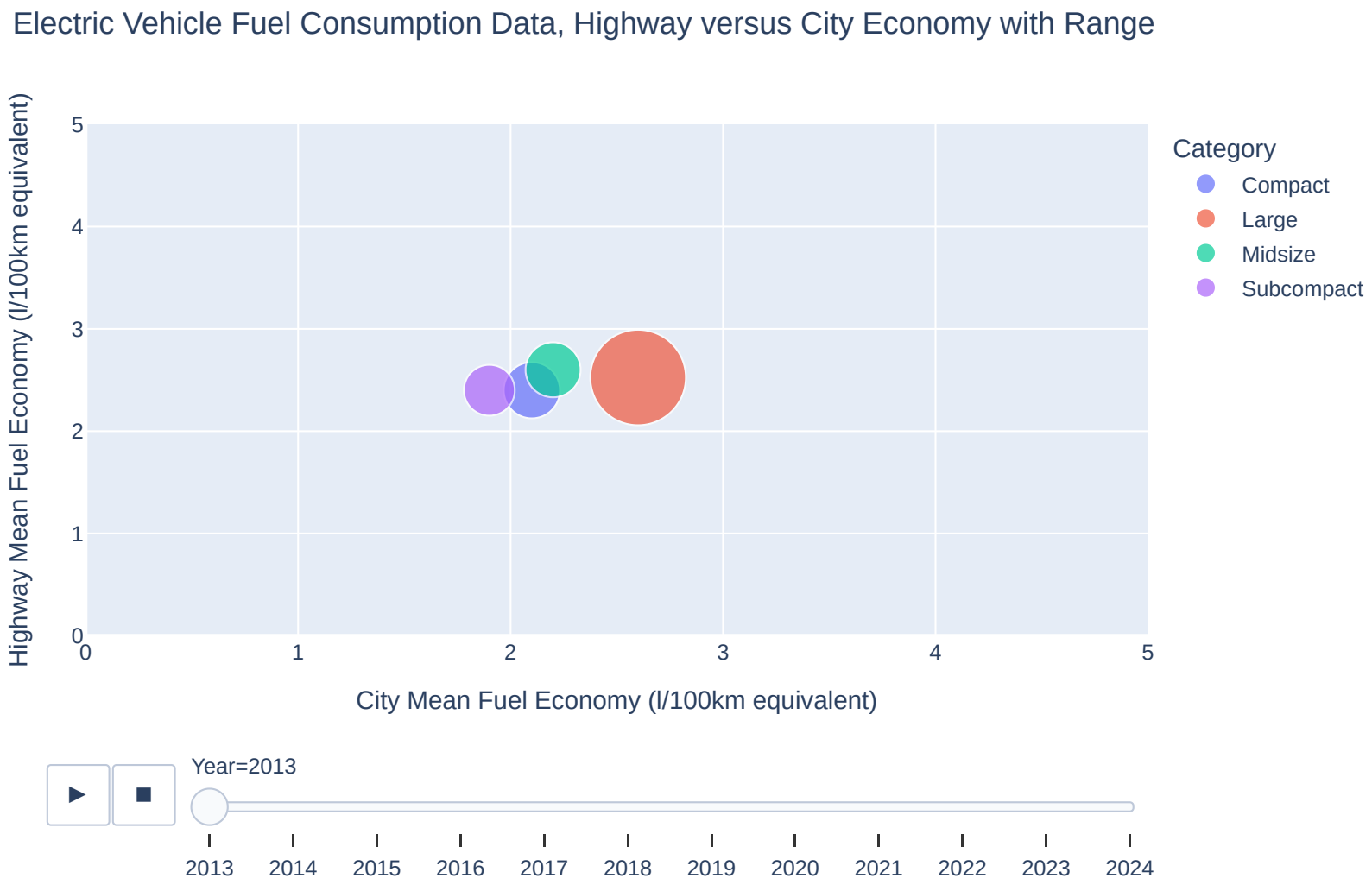
Trendlines and Marginals



Trendlines and Marginals

```
...

df = fuelData[fuelData['Category'].isin(['US', 'UL'])]
fig = px.scatter(df,
            x='City',
            y='Hwy',
            color='Category',
            title='Electric Vehicle Fuel Consumption Data, Highway vers
            trendline='lowess',
            marginal_x='violin',
            marginal_y='rug',
            trendline_options=dict(frac=0.6666))

...
```

Bubbles, Interactive



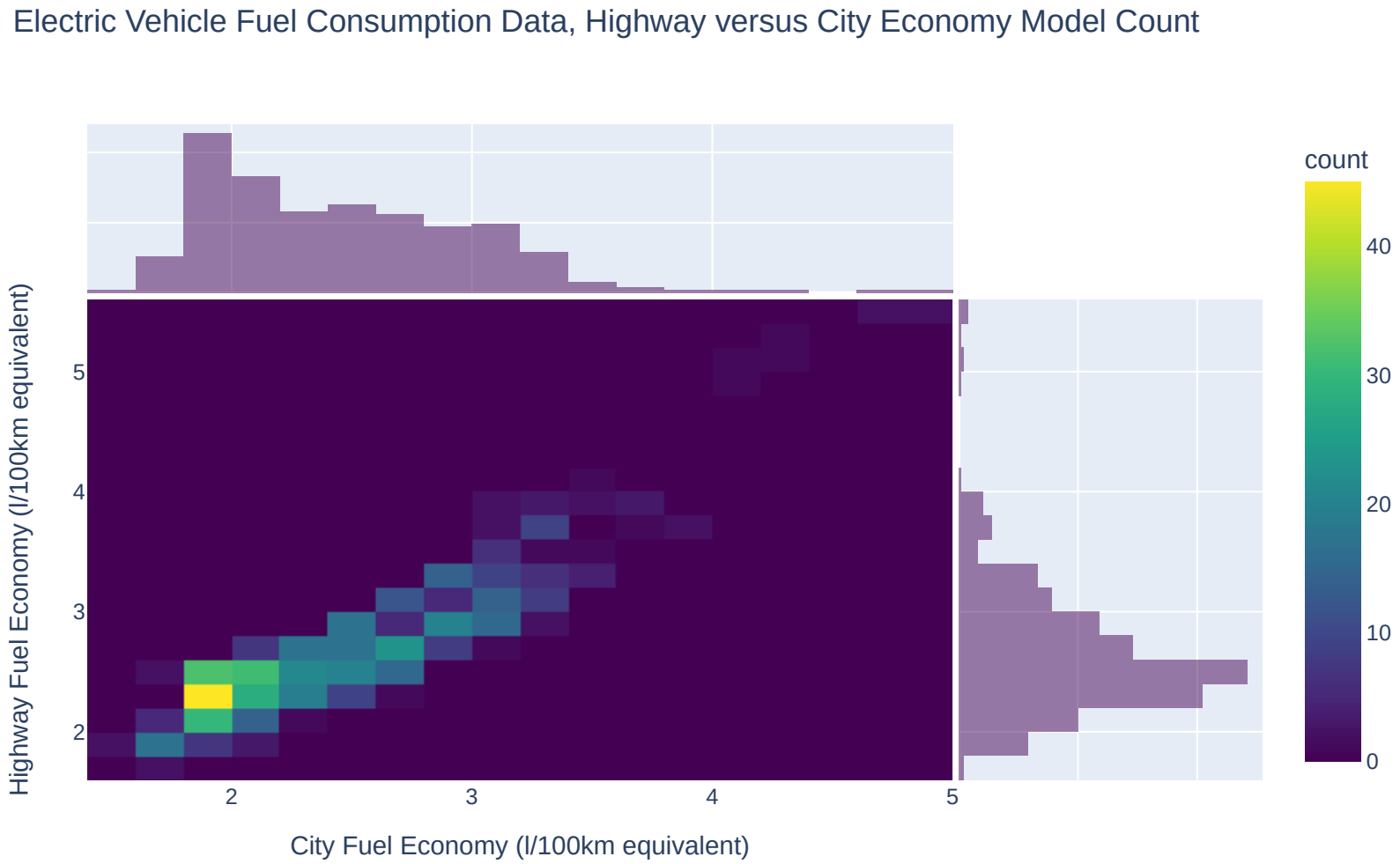
Interactive Demo

MEMORIAL UNIVERSITY

Bubbles, Interactive

```
df = fuelData
df = fuelData[(fuelData['Category'].isin(['S', 'M', 'L', 'C', 'UL']))
              & (fuelData['Year'] > 2012)]
grouped_df = df.groupby(['Category', 'Year'])
              .agg(range=('Range', 'mean'),
                   meanHwy=('Hwy', 'mean'),
                   meanCity=('City', 'mean'))
              .reset_index()

fig = px.scatter(grouped_df,
           x = 'meanCity',
           y = 'meanHwy',
           color = 'Category',
           size = 'range',
           animation_frame = 'Year',
           animation_group = 'Category',
           range_x = [0, 5],
           range_y = [0, 5],
           title='Electric Vehicle Fuel Consumption Data, Highway vers
           template='presentation',
           labels={"range" : "Range",
                   "meanHwy": "Highway Mean Fuel Economy (l/100km equi
                   "meanCity": "City Mean Fuel Economy (l/100km equiva
                 size_max = 50)
```

Heatmap with Marginals



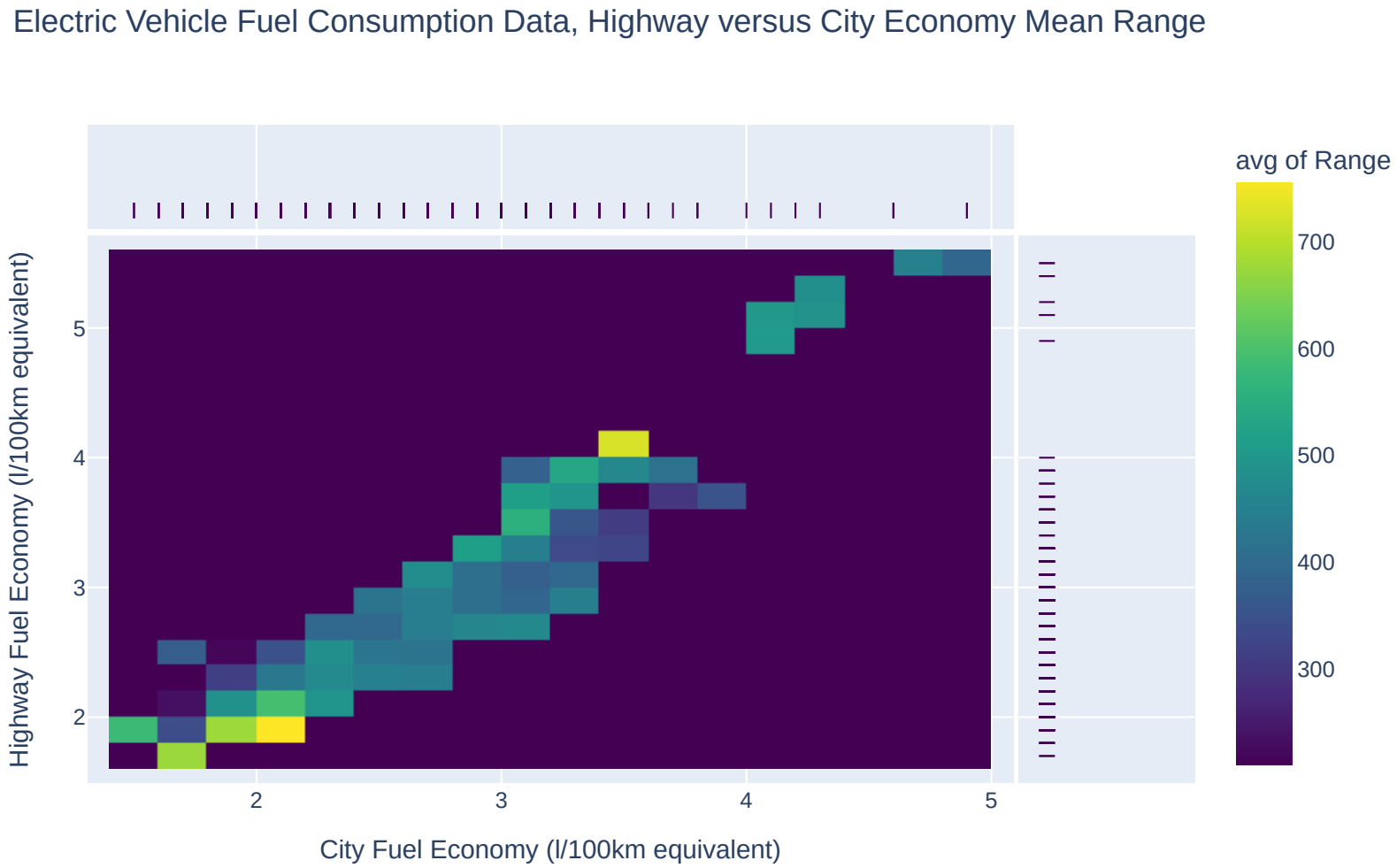
MEMORIAL UNIVERSITY

Heatmap with Marginals

```
df = fuelData

fig = px.density_heatmap(df,
            x = 'City',
            y = 'Hwy',
            nbinsx=20,
            nbinsy=20,
            color_continuous_scale=px.colors.sequential.Viridis,
            marginal_x="histogram",
            marginal_y="histogram",
            title='Electric Vehicle Fuel Consumption Data, Highway vers
            labels={"range" : "Range",
                    "Hwy": "Highway Fuel Economy (l/100km equivalent)",
                    "City": "City Fuel Economy (l/100km equivalent)"})
```

MEMORIAL UNIVERSITY

Heatmap with Marginals



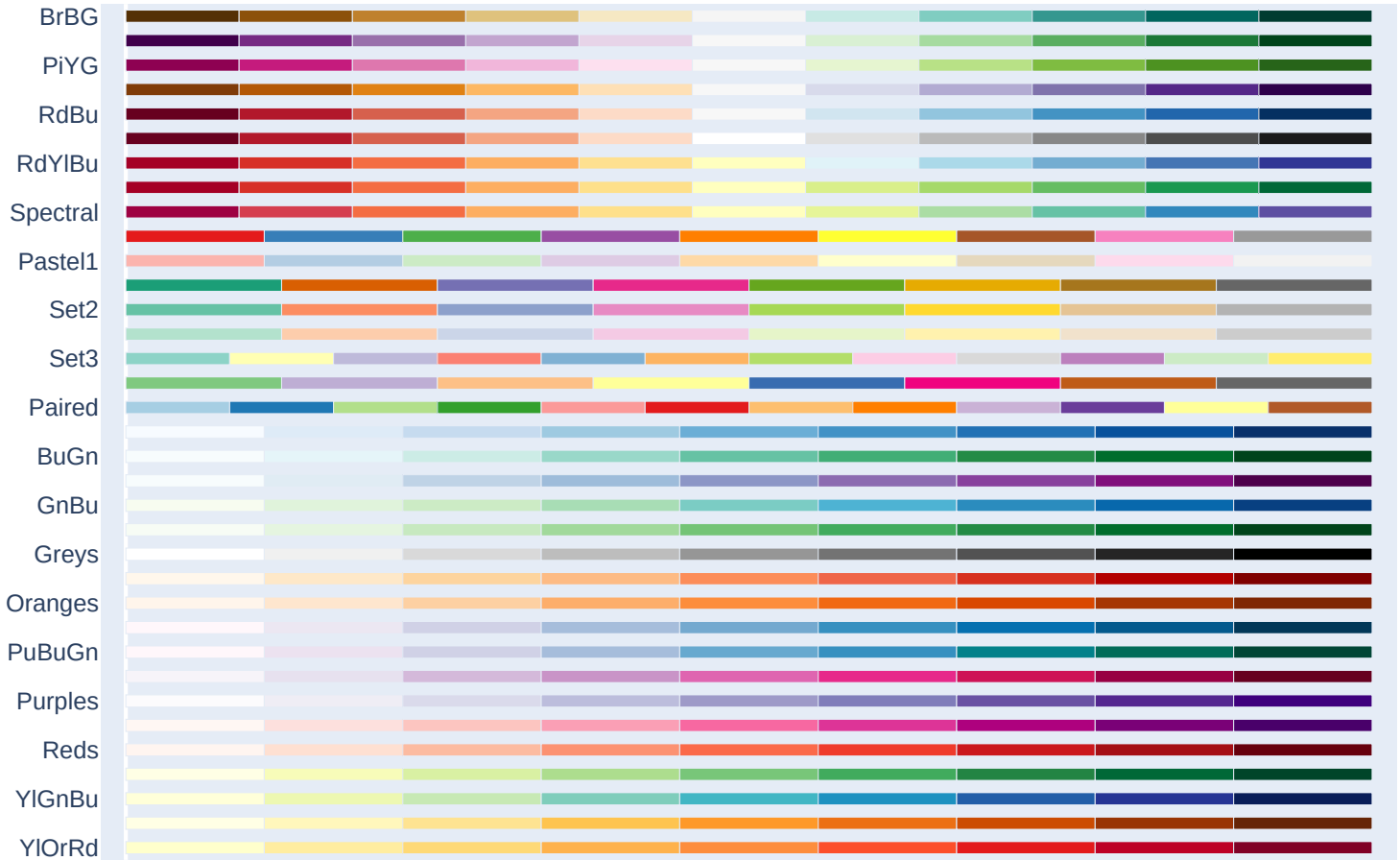
MEMORIAL UNIVERSITY

Heatamp with Marginals

```
df = fuelData

fig = px.density_heatmap(df,
            x = 'City',
            y = 'Hwy',
            nbinsx=20,
            nbinsy=20,
            color_continuous_scale=px.colors.sequential.Viridis,
            marginal_x="rug",
            marginal_y="rug",
            title='Electric Vehicle Fuel Consumption Data, Highway vers
            labels={"range" : "Range",
                    "Hwy": "Highway Fuel Economy (l/100km equivalent)",
                    "City": "City Fuel Economy (l/100km equivalent)"})
```

Colorbrewer Color Swatches

plotly.colors.colorbrewer



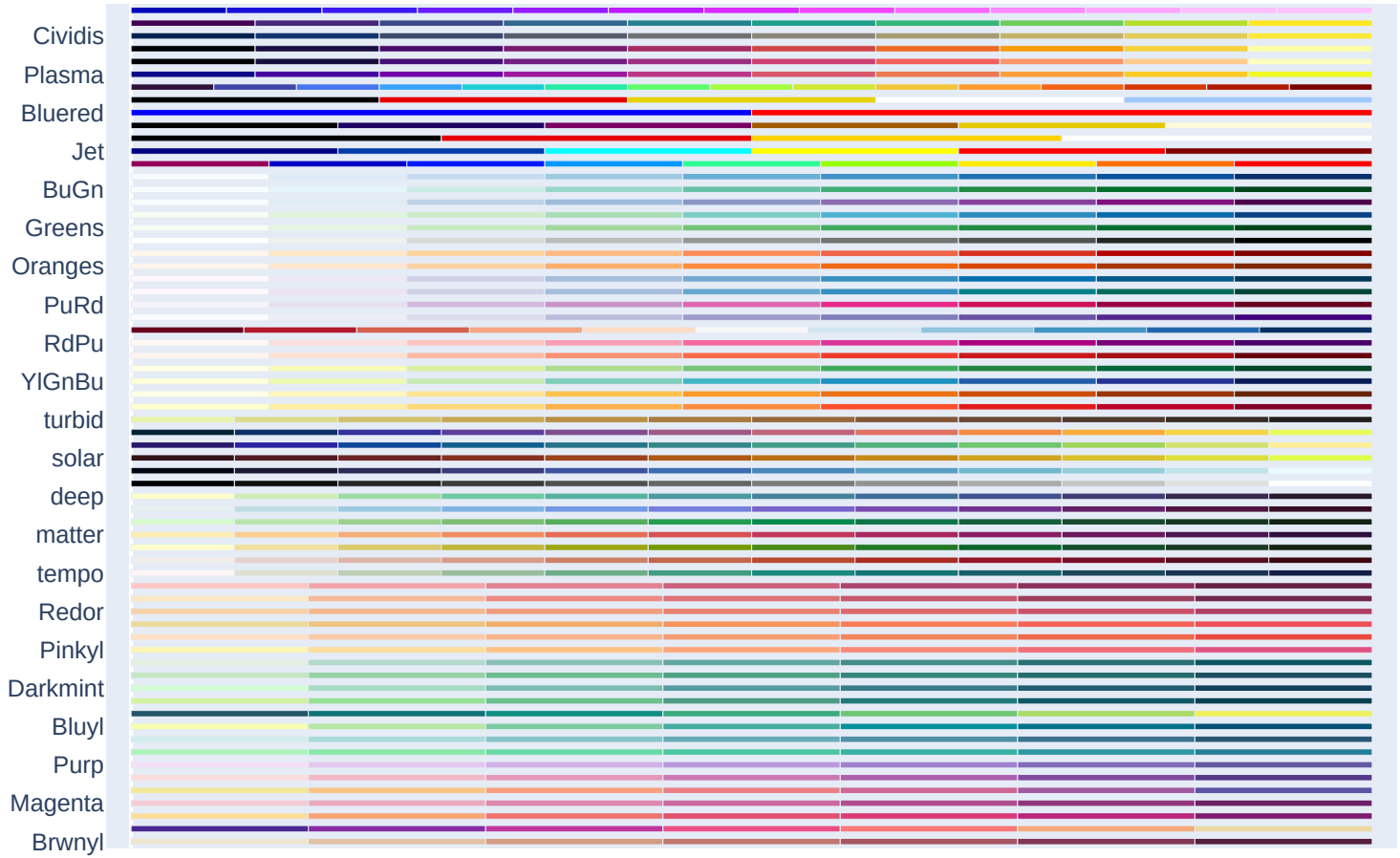
MEMORIAL UNIVERSITY

Discrete Color Swatches

plotly.colors.qualitative

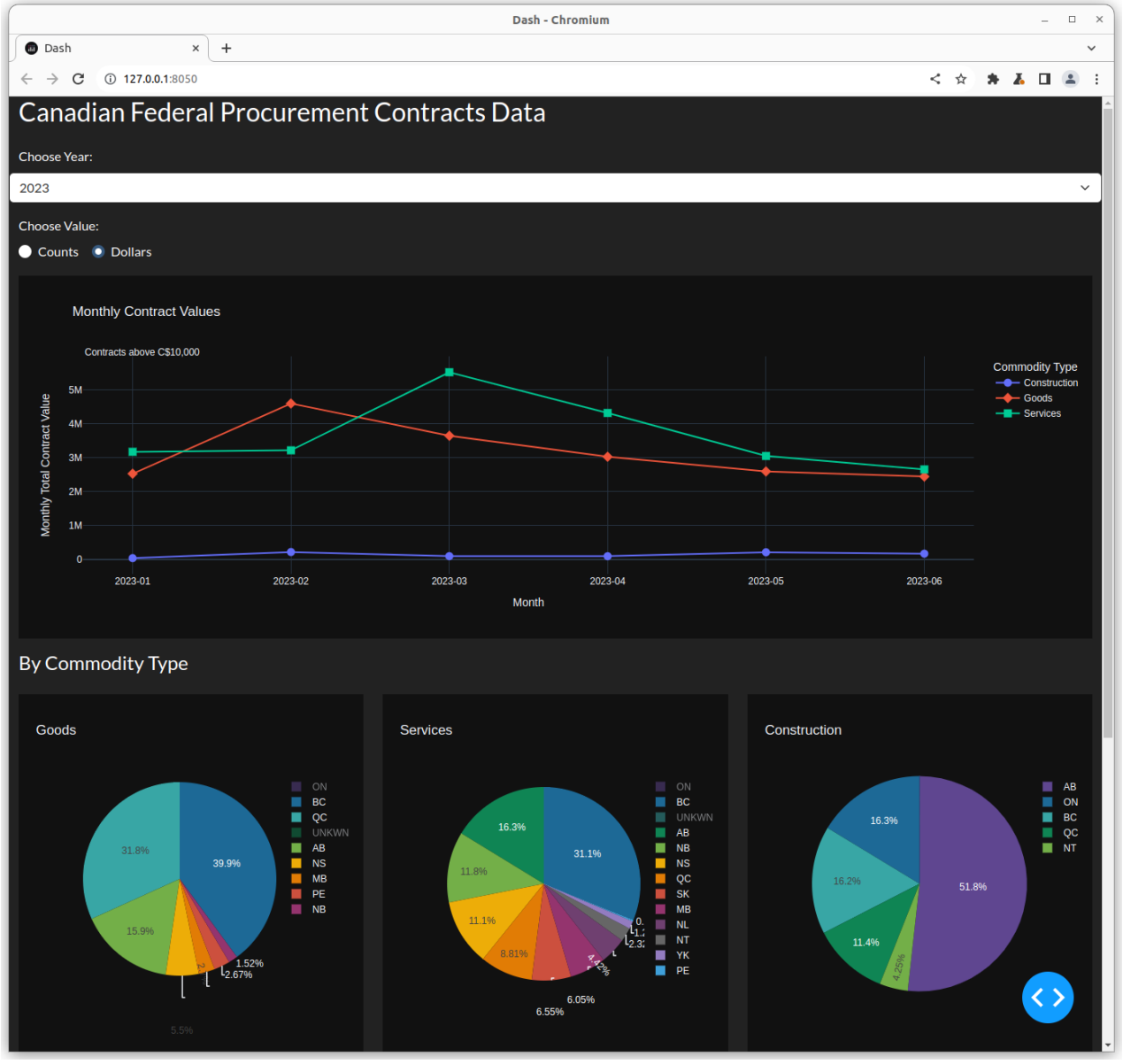
Sequential Color Swatches

plotly.colors.sequential



MEMORIAL UNIVERSITY

Dashboards – Live Demo



MEMORIAL UNIVERSITY

http://joerg.evermann.ca/
DataVisualization.zip