vancouver_eda

May 30, 2023

1 Exploration of Vancouver Trees (EDA)

Author : Muntakim Rahman UBC Student Number : 71065221

1.1 Introduction

This notebook will be conducting an *Exploratory Data Analysis (EDA)* for the Vancouver Trees dataset located in the `small_unique_vancouver.csv` file.

1.2 Import Packages

[1]:
```
import numpy as np
import pandas as pd
import altair as alt
import datetime as dt
```

[2]:
```
vancouver_df = pd.read_csv('small_unique_vancouver.csv', index_col = 0)
display(vancouver_df.head())
```

```
           std_street       on_street   species_name neighbourhood_name  \
10747       W 20TH AV       W 20TH AV    PLATANOIDES         Riley Park
12573       W 18TH AV       W 18TH AV     CALLERYANA      Arbutus-Ridge
29676         ROSS ST         ROSS ST          NIGRA             Sunset
8856         DOMAN ST        DOMAN ST      AMERICANA          Killarney
21098  EAST BOULEVARD  EAST BOULEVARD  HIPPOCASTANUM        Shaughnessy

      date_planted  diameter street_side_name genus_name assigned  \
10747   2000-02-23      28.5             EVEN       ACER        N
12573   1992-02-04       6.0              ODD      PYRUS        N
29676          NaN      12.0              ODD      PINUS        N
8856    1999-11-12      11.0             EVEN   FRAXINUS        N
21098          NaN      15.5              ODD   AESCULUS        Y

       civic_number plant_area curb  tree_id           common_name  \
10747            66         15    Y    21421          NORWAY MAPLE
12573          2323          7    Y   129645      CHANTICLEER PEAR
29676          7855          7    Y   154675         AUSTRIAN PINE
8856           6938          7    Y   180803   AUTUMN APPLAUSE ASH
```

```
21098          5295          N    Y    74364  COMMON HORSECHESTNUT

       height_range_id  on_street_block    cultivar_name root_barrier  \
10747                4                0              NaN            N
12573                2             2300      CHANTICLEER            N
29676                4             7800              NaN            N
8856                 4             6900  AUTUMN APPLAUSE            N
21098                4             5200              NaN            N

        latitude   longitude
10747  49.252711 -123.106323
12573  49.256350 -123.158709
29676  49.213486 -123.083254
8856   49.220839 -123.036721
21098  49.238514 -123.154958
```

[3]:
```
print(f''''There are {len(vancouver_df)} entries in the dataset.''')
```

```
'There are 5000 entries in the dataset.
```

1.2.1 Observe Outputs

Let's start by getting an understanding of the data sparsity (i.e. *NULL* values), as well as the column distributions.

[4]:
```
display(vancouver_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5000 entries, 10747 to 7450
Data columns (total 20 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   std_street          5000 non-null   object
 1   on_street           5000 non-null   object
 2   species_name        5000 non-null   object
 3   neighbourhood_name  5000 non-null   object
 4   date_planted        2363 non-null   object
 5   diameter            5000 non-null   float64
 6   street_side_name    5000 non-null   object
 7   genus_name          5000 non-null   object
 8   assigned            5000 non-null   object
 9   civic_number        5000 non-null   int64
 10  plant_area          4950 non-null   object
 11  curb                5000 non-null   object
 12  tree_id             5000 non-null   int64
 13  common_name         5000 non-null   object
 14  height_range_id     5000 non-null   int64
 15  on_street_block     5000 non-null   int64
 16  cultivar_name       2658 non-null   object
```

```
 17  root_barrier        5000 non-null   object
 18  latitude            5000 non-null   float64
 19  longitude           5000 non-null   float64
dtypes: float64(3), int64(4), object(13)
memory usage: 820.3+ KB

None
```

Data Sparsity There are *NULL* occurances in the `date_planted`, `plant_area`, `cultivar_name` columns. Let's keep these for now to visualize the data in the entries without *NULL* values.

Non-Numeric Data

[5]:
```
objects_df = vancouver_df.describe(include = 'object').T
display(objects_df)
```

```
                   count unique                       top  freq
std_street          5000    603                 CAMBIE ST    52
on_street           5000    607                 CAMBIE ST    49
species_name        5000    171                 SERRULATA   463
neighbourhood_name  5000     22      Renfrew-Collingwood    384
date_planted        2363   1599                2004-02-16     7
street_side_name    5000      4                       ODD  2554
genus_name          5000     67                      ACER  1218
assigned            5000      2                         N  4564
plant_area          4950     38                        10   736
curb                5000      2                         Y  4593
common_name         5000    361  KWANZAN FLOWERING CHERRY   383
cultivar_name       2658    176                   KWANZAN   383
root_barrier        5000      2                         N  4679
```

Observing the data stored as objects, there seem to be variation in distinct values for given columns.

The `std_street` and `on_street` column have greater than *600* distinct values and would not be good candidates for the *EDA*.

Looking at the `date_planted` column, it seems that there are only *1599* distinct values in the entire dataset. This would entail repeated dates across the entries, which is rather interesting.

The `curb` and `root_barrier` columns are binary in nature and should be one-hot encoded in our final analysis.

Numeric Data

[6]:
```
numeric_df = vancouver_df.describe(include = np.number).T
display(numeric_df)
```

```
                 count           mean           std         min  \
diameter        5000.0      12.340888      9.266600    0.000000
civic_number    5000.0    2975.707600   2078.580429    2.000000
tree_id         5000.0  128682.584600  75412.260406   36.000000
```

height_range_id	5000.0	2.734400	1.569570	0.000000
on_street_block	5000.0	2960.227000	2086.861052	0.000000
latitude	5000.0	49.247349	0.021251	49.202783
longitude	5000.0	-123.107128	0.049137	-123.220560

	25%	50%	75%	max
diameter	4.000000	10.000000	18.000000	71.000000
civic_number	1300.500000	2639.000000	4123.000000	9113.000000
tree_id	61321.500000	130130.500000	191332.000000	270750.000000
height_range_id	2.000000	2.000000	4.000000	9.000000
on_street_block	1300.000000	2600.000000	4100.000000	9100.000000
latitude	49.230152	49.247981	49.263275	49.293930
longitude	-123.144178	-123.105861	-123.063484	-123.023311

Observing the data stored as type `np.number`, there seem to be differences in `std deviation` for given columns.

Based on the `std deviation` of *75412.260406*, the `tree_id` column probably includes data for a unique identifier. We can use this to identify our trees, but it doesn't serve much other use for our *EDA*.

There is a very large `std deviation` for the `civic_number` column, with the `min` value being *2* and the `max` being *9113*. There is similar behavior in the `on_street_block` column, which very similar `mean`, `min`, and `max` values to `civic_number`. I'm not particularly interested in these columns, but we can visualize the correlation.

The `height_range_id` column has a `mean` value, as well as a `25th` and `50th` percentile *~2* which is interesting. I'd like to see the distribution of this column.

The `latitude` and `longitude` column have a `std deviation` less than *0.1*, which would entail most trees being in the same vicinity. We can try using this data to see where trees are densely concentrated.

1.3 Questions of Interest

We want to explore this dataset to understand :

- What trees are commonly found in Vancouver?
- Where are trees located in Vancouver?
- How big are these trees?
- When were these trees planted?

1.4 Columns of Interest

We are going to be visualizing the data in the following columns :

- `genus_name`
- `latitude`
- `longitude`
- `neighbourhood_name`
- `height_range_id`
- `diameter`

- plant_area
- date_planted

[7]:
```
vancouver_df = vancouver_df[
    [
        'latitude', 'longitude', 'neighbourhood_name',
        'genus_name',
        'height_range_id', 'diameter', 'plant_area',
        'date_planted'
    ]
]
```

[8]:
```
display(vancouver_df.head())
display(vancouver_df.tail())
```

```
        latitude   longitude neighbourhood_name genus_name  height_range_id  \
10747  49.252711 -123.106323         Riley Park       ACER                4
12573  49.256350 -123.158709      Arbutus-Ridge      PYRUS                2
29676  49.213486 -123.083254             Sunset      PINUS                4
8856   49.220839 -123.036721          Killarney   FRAXINUS                4
21098  49.238514 -123.154958        Shaughnessy   AESCULUS                4

       diameter plant_area date_planted
10747      28.5         15   2000-02-23
12573       6.0          7   1992-02-04
29676      12.0          7          NaN
8856       11.0          7   1999-11-12
21098      15.5          N          NaN
```

```
        latitude   longitude        neighbourhood_name    genus_name  \
6132   49.221161 -123.061023       Victoria-Fraserview        PRUNUS
5642   49.241544 -123.070644  Kensington-Cedar Cottage        CORNUS
8777   49.224511 -123.048723                 Killarney  LIRIODENDRON
23489  49.259208 -123.096905            Mount Pleasant       DAVIDIA
7450   49.243772 -123.078967  Kensington-Cedar Cottage          ACER

       height_range_id  diameter plant_area date_planted
6132                 2      17.0          9          NaN
5642                 1       3.0         10   2014-01-14
8777                 2       3.5          7   2002-04-15
23489                1       5.5          5   2003-12-02
7450                 1       3.0          8          NaN
```

1.5 Exploratory Visualizations

1.5.1 Q1 : What trees are commonly found in Vancouver?

Let's plot the count of each `genus_name` to visualize the most and least common trees within the city.

[9]:
```
genera_plot = alt.Chart(
    vancouver_df, title = 'Figure 1'
).transform_joinaggregate(
    total = 'count(*)'
).transform_calculate(
    pct = '1 / datum.total'
).mark_bar().encode(
    x = alt.X('sum(pct):Q', axis = alt.Axis(format = '.2%')),
    y = alt.Y('genus_name:N', sort = '-x'),
    tooltip = [
        alt.Tooltip('count():Q'),
        alt.Tooltip('sum(pct):Q', format = '.2%', formatType = 'number')
    ]
)

display(genera_plot)
```

`alt.Chart(…)`

From `Figure 1` :

- We observe that there are quite a few number of distinct `tree_genus`, amounting to *67*.
- It's noticable that greater than *45%* of trees are either *Acer* or *Prunus*.
- We might want to look into what other features these trees tend to share.

Q2 : Where are trees located in Vancouver? Let's bin the `latitude` and `longitude` coordinates in a heatmap to visualize the tree density within a given area.

[10]:
```
coordinates_plot = alt.Chart(
    vancouver_df, title = 'Figure 2'
).mark_bar().encode(
    x = alt.X('latitude:Q',bin = alt.Bin(maxbins = 15)),
    y = alt.Y('longitude:Q', bin = alt.Bin(maxbins = 15)),
    color = alt.Color(
        'count():Q', scale = alt.Scale(scheme = 'viridis', reverse = True),
        legend = alt.Legend(),
    ),
    tooltip = [alt.Tooltip('count():Q')]
)
```

[11]:
```
display(coordinates_plot)
```

`alt.Chart(…)`

From `Figure 2` :

- We observe that trees are generally distributed rather evenly.
- It's noticable that there are *230* trees within *49.250* <= `latitude` <= *29.260* and *-123.120* <= `longitude` <= *-123.100*.

- There are fewer trees located around the edges of the map, with the exception of *49.220* <= `latitude` <= *49.290* and *-123.040* <= `longitude` <= *-123.020*.

1.5.2 Q3 : What Sizes are Vancouver Trees? Is there a Relationship Between Diameter, Height Range ID and Plant Area?

Let's explore the `plant_area` column. The values in this column are stored as *objects*, which is interesting.

[12]:
```
def convert_to_int(unconverted) :
    try :
        converted = int(unconverted)
    except ValueError :
        converted = np.nan
    return converted

distinct_areas = [
    str(val) for val in sorted([int(area) for area in␣
 ↪list(vancouver_df['plant_area'].unique()) if convert_to_int(area) ==␣
 ↪convert_to_int(area)])
] + sorted([
    str(area) for area in list(vancouver_df['plant_area'].unique()) if␣
 ↪convert_to_int(area) != convert_to_int(area)
])
```

[13]:
```
areas_plot = alt.Chart(
    vancouver_df, title = 'Figure 3'
).mark_bar().encode(
    x = alt.X(
        'plant_area:N', sort = distinct_areas
    ),
    y = alt.Y('count():Q'),
    tooltip = [alt.Tooltip('count():Q')]
)

display(areas_plot)
```

`alt.Chart(…)`

From `Figure 3` :

- We observe that a number of trees have alphabetical `plant_area` values,.
  - *464* trees with *N* `plant_area`
- There seems to be a high number of trees with `plant_area` of *10*.
  - Trees generally seem to have a `plant_area` <= *12*.
  - Very few trees have a `plant_area` of *2* or *11*.

Let's look into the relationship between the `diameter`, `height_range_id`, and `plant_area` columns.

[14]:
```
def plot_sizes_heatmap(heatmap_df, dimension_1, dimension_2) :
    corr_heatmap = alt.Chart(
        heatmap_df
    ).mark_circle().encode(
        x = alt.X(f'{dimension_1}:Q', bin = alt.Bin(maxbins = 15)),
        y = alt.Y(f'{dimension_2}:Q', bin = alt.Bin(maxbins = 15)),
        color = alt.Color(
            'count():Q', scale = alt.Scale(
                scheme = 'viridis', reverse = True,
            ),
            legend = alt.Legend(),
        ),
        size = alt.Size('count():Q'),
        tooltip = [alt.Tooltip('count():Q')]
    )

    return corr_heatmap
```

[15]:
```
sizes_heatmaps = alt.hconcat(
    plot_sizes_heatmap(vancouver_df, 'diameter', 'height_range_id'),
    plot_sizes_heatmap(vancouver_df, 'diameter', 'plant_area'),
    plot_sizes_heatmap(vancouver_df, 'height_range_id', 'plant_area')
).properties(title = 'Figure 4')

display(sizes_heatmaps)
```

`alt.HConcatChart(…)`

From `Figure 4` :

- We observe that there is a slight positive relationship between `diameter` and `height_range_id` where *0* <= `diameter` <= *25* and *1.0* <= `height_range_id` <= *5.0*.
  - More trees seem to tend towards having feature values in the lower bins of this domain. (e.g. There are *959* trees with *0* <= `diameter` <= *5* and *1.0* <= `height_range_id` <= *2.0*.)
- Trees tend towards lower `diameter` values and *5* <= `plant_area` <=*10*, as well as *5* <= `height_range_id` <= *10*.

1.5.3 Q4 : What neighborhoods have the largest trees? What about the smallest trees?

Let's look at the breakdown of this data for both `diameter` and `height_range_id` by `neighborhood_name`.

[16]:
```
neighbourhoods_plot = alt.Chart(
    vancouver_df,
    width = 300, height = 350
).mark_boxplot().encode(
    x = alt.X(alt.repeat(), type = 'quantitative'),
```

```
    y = alt.Y('neighbourhood_name:N'),
).repeat(
    ['diameter', 'height_range_id', 'plant_area'],
    columns = 3
).properties(
    title = 'Figure 5'
)

display(neighbourhoods_plot)
```

`alt.RepeatChart(…)`

From `Figure 5` :

- For `diameter` :
  - Most regions tend to have *~10 <= median* ***`diameter`*** *<= ~15*
  - All regions have a *75th* percentile `diameter` *<= ~20*
  - The *Downtown* region has a lower range of `diameter` between *25th* and *75th* percentiles.
    - The `diameter` values here tend towards *<=10*.
- For `height_range_id` :
  - There seem to be *2* buckets of `neighbourhood_names`, whether either they have a higher *75th* percentile *>= 4* or a lower one *~3*.
- For `plant_area` :
  - The `plant_area` in between the *25th* and *75th* percentiles have a range of *< 10*.

1.5.4 Q5 : How did tree sizes change by decade?

[17]:
```
vancouver_df = vancouver_df.assign(
    decade_planted = vancouver_df['date_planted'].apply(
        lambda x : f'''{(dt.datetime.strptime(x, '%Y-%m-%d').year // 10) *␣
↪10}s''' if x == x else np.nan
    )
)

display(vancouver_df.head())
display(vancouver_df.tail())
```

```
        latitude   longitude neighbourhood_name genus_name  height_range_id  \
10747  49.252711 -123.106323         Riley Park       ACER                4
12573  49.256350 -123.158709      Arbutus-Ridge      PYRUS                2
29676  49.213486 -123.083254             Sunset      PINUS                4
8856   49.220839 -123.036721          Killarney   FRAXINUS                4
21098  49.238514 -123.154958        Shaughnessy   AESCULUS                4

       diameter plant_area date_planted decade_planted
10747      28.5         15   2000-02-23          2000s
12573       6.0          7   1992-02-04          1990s
29676      12.0          7          NaN            NaN
```

```
8856       11.0          7   1999-11-12          1990s
21098      15.5          N          NaN            NaN

        latitude   longitude        neighbourhood_name    genus_name  \
6132   49.221161 -123.061023       Victoria-Fraserview        PRUNUS
5642   49.241544 -123.070644  Kensington-Cedar Cottage        CORNUS
8777   49.224511 -123.048723                 Killarney  LIRIODENDRON
23489  49.259208 -123.096905            Mount Pleasant       DAVIDIA
7450   49.243772 -123.078967  Kensington-Cedar Cottage          ACER

       height_range_id  diameter plant_area date_planted decade_planted
6132                 2      17.0          9          NaN            NaN
5642                 1       3.0         10   2014-01-14          2010s
8777                 2       3.5          7   2002-04-15          2000s
23489                1       5.5          5   2003-12-02          2000s
7450                 1       3.0          8          NaN            NaN
```

[18]:
```
decades_plot = alt.Chart(
    vancouver_df,
    width = 300, height = 350
).mark_area(opacity = 0.5).encode(
    x = alt.X(alt.repeat(), type = 'quantitative', bin = alt.Bin(maxbins = 15)),
    y = alt.Y('count():Q', stack = None),
    color = alt.Color(
        'decade_planted:O', scale = alt.Scale(
            scheme = 'tableau10', reverse = False,
        ), legend = alt.Legend(),
    )
).repeat(
    ['diameter', 'height_range_id', 'plant_area'], columns = 3
).properties(
    title = 'Figure 6'
)


display(decades_plot)
```

```
alt.RepeatChart(…)
```

From `Figure 6` :

- For `diameter` and `height_range_id`, the distributions seem to tend towards lower values for the *2010s*.
  - The data with *null* `decade` information seems to have distributions with higher values.
- For all visualized columns, the entries with *null* `decades` seem to outnumber the other individual `decades`.
- For `plant_area`, the distributions tend towards a normal distribution centered around *~7-8*.
  - Note that the *2010s* appear to have fewer entries when looking at the `plant_area`, most likely due to trees here having alphabetically encoded values.

1.6 Concluding Remarks

I would like to explore the data in these charts when filtered for criteria including :

- `neighbourhood_names` with the most trees
- most common `genus_names`

A few questions start to emerge when looking at data for the columns we've considered for size, as well as trends over the `decades`.

Do trees of the same `genus_name` have similar numerical features? Do trees with the same `neighbourhood_name` tend to have the same `genus_names`? Where are more trees being planted over the `decades`? Has the tree density by `neighbourhood_names` changed over the `decades`?

1.7 Interactive Dashboard

Let's create a dashboard from the visuals above in order to start investigating these questions. This enables us to consider these data insights adjacent to one another. We're going to be filtering our charts with the `neighbourhood_name`, `genus_name`, and `decade_planted` fields.

[19]:
```
neighbourhoods_select = alt.selection_single(
    fields = ['neighbourhood_name'],
    bind = {
        'neighbourhood_name' : alt.binding_select(
            name = 'Neighbourhoods',
            options = list(
                vancouver_df.groupby('neighbourhood_name')['genus_name'] \
                    .agg('count').sort_values(ascending = False) \
                    .reset_index()['neighbourhood_name']
            )
        )
    }
)
```

[20]:
```
genus_select = alt.selection_single(
    fields = ['genus_name'],
    bind = {
        'genus_name' : alt.binding_select(
            name = 'Tree Genus',
            options = list(
                vancouver_df.groupby('genus_name')['neighbourhood_name'] \
                    .agg('count').sort_values(ascending = False) \
                    .reset_index()['genus_name']
            )
        )
    }
)
```

[21]:
```
decades_select = alt.selection_single(
    fields = ['decade_planted'],
```

```
    bind = {
        'decade_planted' : alt.binding_radio(
            name = 'Decades',
            options = sorted([decade for decade in␣
↪vancouver_df['decade_planted'].unique() if decade == decade])
        )
    }
)
```

[22]:
```
coordinates_plot = alt.Chart(
    vancouver_df,
    title = alt.TitleParams(
        text = f'Location of Trees in Vancouver',
        subtitle = ['Latitude and Longitude Heatmap'],
        anchor = 'start', fontSize = 25, subtitleFontSize = 20
    )
).mark_bar().encode(
    x = alt.X('latitude:Q', title = 'Latitude', bin = alt.Bin(maxbins = 15)),
    y = alt.Y('longitude:Q', title = 'Longitude', bin = alt.Bin(maxbins = 15)),
    color = alt.Color(
        'count():Q', scale = alt.Scale(scheme = 'viridis', reverse = True),
        legend = None
    ),
    tooltip = [alt.Tooltip('count():Q', title = 'Number of Trees')]
).add_selection(
    neighbourhoods_select
).add_selection(
    genus_select
).add_selection(
    decades_select
).transform_filter(
    neighbourhoods_select
).transform_filter(
    genus_select
).transform_filter(
    decades_select
)
```

[23]:
```
decades_plot = alt.Chart(
    vancouver_df,
    width = 300, height = 350
).mark_bar().encode(
    x = alt.X(alt.repeat(), type = 'quantitative', bin = alt.Bin(maxbins = 15)),
    y = alt.Y('count():Q', title = 'Number of Trees', stack = None),
    color = alt.Color(
        'decade_planted:O', scale = alt.Scale(
            scheme = 'tableau10', reverse = False,
```

```
        ),
        legend = alt.Legend(
            title = 'Decade Planted',
            titleFontSize = 14, labelFontSize = 12
        )
    )
).encode(
    opacity = alt.condition(
        decades_select, alt.value(0.75), alt.value(0.25)
    )
).repeat(
    ['diameter', 'height_range_id', 'plant_area'], columns = 3
).properties(
    title = alt.TitleParams(
        text = f'Sizes of Vancouver Tree in Different Decades',
        subtitle = ['Diameters, Height Ranges, and Plant Areas in Different␣
 ↪Decades'],
        anchor = 'start', fontSize = 25, subtitleFontSize = 20
    )
).transform_filter(
    neighbourhoods_select
).transform_filter(
    genus_select
)
```

[24]:
```
def plot_sizes_heatmap(heatmap_df, dimension_1, dimension_2) :
    corr_heatmap = alt.Chart(
        heatmap_df, title = 'Figure 4'
    ).mark_circle().encode(
        x = alt.X(f'{dimension_1}:Q', title = ' '.join([word.capitalize() for␣
 ↪word in dimension_1.split('_')]), bin = alt.Bin(maxbins = 15)),
        y = alt.Y(f'{dimension_2}:Q', title = ' '.join([word.capitalize() for␣
 ↪word in dimension_2.split('_')]), bin = alt.Bin(maxbins = 15)),
        color = alt.Color(
            'count():Q', scale = alt.Scale(
                scheme = 'viridis', reverse = True,
            ),
            legend = alt.Legend(
                title = 'Number of Trees',
                titleFontSize = 14, labelFontSize = 12
            ),
        ),
        size = alt.Size('count():Q'),
        tooltip = [alt.Tooltip('count():Q', title = 'Number of Trees')]
    )

    return corr_heatmap
```

[25]:
```
sizes_heatmaps = alt.hconcat(
    plot_sizes_heatmap(vancouver_df, 'diameter', 'height_range_id'),
    plot_sizes_heatmap(vancouver_df, 'diameter', 'plant_area'),
    plot_sizes_heatmap(vancouver_df, 'height_range_id', 'plant_area')
).properties(
  title = alt.TitleParams(
    text = f'Sizes of Trees in Vancouver',
    subtitle = ['Relationship Between Diameter, Height Range ID, and Plant␣
 ↪Area'],
    anchor = 'start', fontSize = 25, subtitleFontSize = 20
  )
).transform_filter(
    neighbourhoods_select
).transform_filter(
    genus_select
).transform_filter(
    decades_select
)
```

[26]:
```
genera_plot = alt.Chart(
    vancouver_df,
    title = alt.TitleParams(
        text = f'Genera of Vancouver Trees',
        anchor = 'start', fontSize = 25, subtitleFontSize = 20
    )
).transform_joinaggregate(
    total = 'count(*)'
).transform_calculate(
    pct = '1 / datum.total'
).mark_bar().encode(
    x = alt.X('genus_name:N', title = 'Tree Genera', sort = '-y'),
    y = alt.Y('sum(pct):Q', axis = alt.Axis(format = '.2%'), title = '% of␣
 ↪Total Trees'),
    tooltip = [
        alt.Tooltip('count():Q', title = 'Number of Trees'),
        alt.Tooltip('sum(pct):Q', format = '.2%', formatType = 'number', title␣
 ↪= '% of Total Trees')
    ]
).transform_filter(
    neighbourhoods_select
).transform_filter(
    decades_select
)
```

[27]:
```
neighbourhoods_plot = alt.Chart(
    vancouver_df,
    title = alt.TitleParams(
```

```
        text = f'Neighbourhoods of Vancouver Trees',
        anchor = 'start', fontSize = 25, subtitleFontSize = 20
    )
).transform_joinaggregate(
    total = 'count(*)'
).transform_calculate(
    pct = '1 / datum.total'
).mark_bar().encode(
    x = alt.X('sum(pct):Q', axis = alt.Axis(format = '.2%'), title = '% of␣
↪Total Trees'),
    y = alt.Y('neighbourhood_name:N', title = 'Neighbourhoods', sort = '-x'),
    tooltip = [
        alt.Tooltip('count():Q', title = 'Number of Trees'),
        alt.Tooltip('sum(pct):Q', format = '.2%', formatType = 'number', title␣
↪= '% of Total Trees')
    ]
).transform_filter(
    genus_select
).transform_filter(
    decades_select
)
```

[28]:
```
eda_dashboard = (
  (
    (coordinates_plot | neighbourhoods_plot) & genera_plot
  ) | (
    decades_plot & sizes_heatmaps
  ).resolve_scale(
    color = 'independent', size = 'independent'
  )
).configure_mark(
  stroke = 'black', strokeOpacity = 1, strokeWidth = 0.5
).configure_axis(
    labelFontSize = 15, titleFontSize = 17.5
)

display(eda_dashboard)
```

```
alt.HConcatChart(…)
```

1.8 References

These resources provide the data, theory and code segments for the *EDA* exploration in this notebook.

- Data Visualization
- Machine Learning Final Project