project_report

May 12, 2022

1 Fun Facts about the distribution of Vacouver Trees

May 08, 2022

Final Project Notebook by Xiao Juan Li

1.1 Foreword

This notebook will be showing exploratory data analysis for the subset of the Vancouver Street Trees dataset located here.

1.2 Introduction

1.2.1 Motivation

As we know, Vancouver is one of the most livable cities in Canada and in the world. It is also one of Canada's warmest cities in the winter.No doubt this is a wonderful place for various kinds of trees to live and grow. I am really excited about finding some fun facts about these trees in such a fabulous city.

Vancouver plans to become the greenest city in the world. I am curious about how many trees have been planted over the years and how the distribution of trees has been evolved through different neighbourhoods.Besides, the Vancouver streets are famous for the breathtaking trees view, are there any distribution rules or not? Moreover, Vancouver is one of the most expensive cities in terms of housing affordability in Canada. Are there any prestigious communities and what the natural enviornment around them would be like? Are they related to the trees distribution as well?Last but not least, Which range of tree size is most prominent in the whole city? Is that be small,medium or big? I just can't wait to find all the answers through the following data exploraton and hope we might find some hidden treasures too. We will be able to address these questions using an interactive dashboard.

1.2.2 Questions of interest

- 1.How is the distribution of trees evolved over the years through different neighbourhoods in Vancouver?

- 2.How do trees differ among street sides in Vancouver?
- 3.Which neighbourhood is surrounded by the most giant trees in Vancouver?
- 4.Which range of tree size is most prominent in Vancouver?

1.3 Analysis

1.3.1 Data Imports

[39]:
```python
# Import libraries needed for this assignment

import altair as alt
import pandas as pd
import os
import json

# alt.data_transformers.enable("data_server")
```

Let's import the subset of the Vancouver Street Trees data. Since this is a new dataset,let's take a good first step to get familiar with it by glancing at the values in the dataframe.

[40]:
```python
trees_df = pd.read_csv('small_unique_vancouver.csv')
trees_df.head()
```

[40]:
```
   Unnamed: 0      std_street       on_street   species_name  \
0       10747       W 20TH AV       W 20TH AV    PLATANOIDES
1       12573       W 18TH AV       W 18TH AV     CALLERYANA
2       29676         ROSS ST         ROSS ST          NIGRA
3        8856        DOMAN ST        DOMAN ST      AMERICANA
4       21098  EAST BOULEVARD  EAST BOULEVARD  HIPPOCASTANUM

  neighbourhood_name date_planted  diameter street_side_name genus_name  \
0         Riley Park   2000-02-23      28.5             EVEN       ACER
1      Arbutus-Ridge   1992-02-04       6.0              ODD      PYRUS
2             Sunset          NaN      12.0              ODD      PINUS
3          Killarney   1999-11-12      11.0             EVEN   FRAXINUS
4        Shaughnessy          NaN      15.5              ODD   AESCULUS

  assigned  …  plant_area curb tree_id             common_name  \
0        N  …          15    Y   21421            NORWAY MAPLE
1        N  …           7    Y  129645        CHANTICLEER PEAR
2        N  …           7    Y  154675            AUSTRIAN PINE
3        N  …           7    Y  180803     AUTUMN APPLAUSE ASH
4        Y  …           N    Y   74364  COMMON HORSECHESTNUT

  height_range_id  on_street_block    cultivar_name root_barrier   latitude  \
0               4                0              NaN            N  49.252711
1               2             2300      CHANTICLEER            N  49.256350
2               4             7800              NaN            N  49.213486
3               4             6900  AUTUMN APPLAUSE            N  49.220839
4               4             5200              NaN            N  49.238514

    longitude
```

```
0 -123.106323
1 -123.158709
2 -123.083254
3 -123.036721
4 -123.154958

[5 rows x 21 columns]
```

1.3.2 Data Summary Tables and Methods

Now let's check the type of data in each column and how many missing values there are.

[41]:
```
trees_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          5000 non-null   int64
 1   std_street          5000 non-null   object
 2   on_street           5000 non-null   object
 3   species_name        5000 non-null   object
 4   neighbourhood_name  5000 non-null   object
 5   date_planted        2363 non-null   object
 6   diameter            5000 non-null   float64
 7   street_side_name    5000 non-null   object
 8   genus_name          5000 non-null   object
 9   assigned            5000 non-null   object
 10  civic_number        5000 non-null   int64
 11  plant_area          4950 non-null   object
 12  curb                5000 non-null   object
 13  tree_id             5000 non-null   int64
 14  common_name         5000 non-null   object
 15  height_range_id     5000 non-null   int64
 16  on_street_block     5000 non-null   int64
 17  cultivar_name       2658 non-null   object
 18  root_barrier        5000 non-null   object
 19  latitude            5000 non-null   float64
 20  longitude           5000 non-null   float64
dtypes: float64(3), int64(5), object(13)
memory usage: 820.4+ KB
```

From the above infomation,the datatype of date_planted is object, we need to parse dates as numbers. We can specify parse_dates=['date_planted'] to read_csv again.

Also, it looks like there are some NaNs in three of the columns, and the date_planted and cultivar_name seem to have the most: about half rows are missing a value.

Now we are parsing the dates and then we'll reprint the info of the dataset.

[42]:
```python
# parsing the dates
trees_df = pd.read_csv('small_unique_vancouver.
↪csv',parse_dates=['date_planted'])
trees_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          5000 non-null   int64
 1   std_street          5000 non-null   object
 2   on_street           5000 non-null   object
 3   species_name        5000 non-null   object
 4   neighbourhood_name  5000 non-null   object
 5   date_planted        2363 non-null   datetime64[ns]
 6   diameter            5000 non-null   float64
 7   street_side_name    5000 non-null   object
 8   genus_name          5000 non-null   object
 9   assigned            5000 non-null   object
 10  civic_number        5000 non-null   int64
 11  plant_area          4950 non-null   object
 12  curb                5000 non-null   object
 13  tree_id             5000 non-null   int64
 14  common_name         5000 non-null   object
 15  height_range_id     5000 non-null   int64
 16  on_street_block     5000 non-null   int64
 17  cultivar_name       2658 non-null   object
 18  root_barrier        5000 non-null   object
 19  latitude            5000 non-null   float64
 20  longitude           5000 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(5), object(12)
memory usage: 820.4+ KB
```

1.3.3 Visualizing Missing Values

Visualizing missing values helps us identify potential issues with the data.

[43]:
```python
alt.data_transformers.disable_max_rows();
trees_nans = trees_df.isna().reset_index().melt(id_vars='index',␣
↪var_name='column', value_name='NaN')
trees_nans
```

[43]:

	index	column	NaN
0	0	Unnamed: 0	False
1	1	Unnamed: 0	False

```
2           2  Unnamed: 0  False
3           3  Unnamed: 0  False
4           4  Unnamed: 0  False
…         …           …     …
104995   4995   longitude  False
104996   4996   longitude  False
104997   4997   longitude  False
104998   4998   longitude  False
104999   4999   longitude  False

[105000 rows x 3 columns]
```

[44]:
```python
color_scale = alt.Scale(range=['#dde8f1', 'steelblue'][::1])

nan_heatmap = (
        alt.Chart(trees_nans, title='Individual NaNs').mark_rect(height=17).
 ↪encode(
            alt.X('index:O', axis=None),
            alt.Y('column', title=None),
            alt.Color('NaN', scale=color_scale, sort=[False, True],
                      legend=alt.Legend(orient='top', offset=13), title=None),
            alt.Stroke('NaN', scale=color_scale, sort=[False, True],␣
 ↪legend=None))
        .properties(width=900))
nan_heatmap
```

[44]:
```
alt.Chart(…)
```

By visualizing the missing values for each column next to each other, we can quickly see if there are similar patterns between columns.From the above plot we find that the missing values from cultivar_name and date_planted are not exactly the same rows,although they both have about half rows missing a value.The column plant_area has only 1% rows missing a value.

Since cultivar_name and plant_area are categorical columns showing trees description information,we are not dropping these NaN values if we are not interested in them.For the column date_planted,we can drop the NaN values when we focus on the statistics related to the time. Considering almost half of rows missing a value in date_planted, we might keep the NaN values rather than drop them when we deal with time unrelated statistics.

1.3.4 Early Data Analysis

A statistical summary is useful to complement visualizations.Let's print out the summary statistics for the numerical columns.

[45]:
```python
trees_df.describe()
```

[45]:
```
        Unnamed: 0     diameter  civic_number      tree_id  \
count  5000.000000  5000.000000   5000.000000  5000.000000
```

```
mean   14861.920400    12.340888   2975.707600  128682.584600
std     8680.023278     9.266600   2078.580429   75412.260406
min        2.000000     0.000000      2.000000      36.000000
25%     7192.750000     4.000000   1300.500000   61321.500000
50%    14870.000000    10.000000   2639.000000  130130.500000
75%    22366.750000    18.000000   4123.000000  191332.000000
max    29992.000000    71.000000   9113.000000  270750.000000

       height_range_id  on_street_block     latitude    longitude
count       5000.00000      5000.000000  5000.000000  5000.000000
mean           2.73440      2960.227000    49.247349  -123.107128
std            1.56957      2086.861052     0.021251     0.049137
min            0.00000         0.000000    49.202783  -123.220560
25%            2.00000      1300.000000    49.230152  -123.144178
50%            2.00000      2600.000000    49.247981  -123.105861
75%            4.00000      4100.000000    49.263275  -123.063484
max            9.00000      9100.000000    49.293930  -123.023311
```

Visualizing the distributions of all numerical columns helps us understand the data.

The first column unnamed:0 seems like the id for each row in the original dataset,we have not much interest in it when discovering the numerical columns relationships through visualization. We are going to ignore this column in the following numerical columns exploring.

[46]:
```
# remove the first column (unnamed:0)from numerical columns
numerical_columns = trees_df.iloc[:,1:].select_dtypes('number').columns.tolist()

(alt.Chart(trees_df)
 .mark_bar().encode(
     alt.X(alt.repeat(), type='quantitative', bin=alt.Bin(maxbins=25)),
     y='count()')
 .properties(width=220, height=150)
 .repeat(numerical_columns,columns=3))
```

[46]: alt.RepeatChart(…)

This overview tells us that most trees have a diameter of less than 5 inches, and height between 10 to 30 feet. As trees get bigger and taller,the count numbers are going down.Also, the civic number and street blocks number seem to share the same distribution.Last but not least, the horizontal distribution of trees concentrates on the middle part, while the vertical distribution concentrates on the upper part.

Repeating columns of both X and Y lets us effectively explore pairwise relationships between columns.

[47]:
```
# Scroll right on the plot to see the last column
(alt.Chart(trees_df)
 .mark_point(size=10).encode(
     alt.X(alt.repeat('column'), type='quantitative'),
```

```
    alt.Y(alt.repeat('row'), type='quantitative'))
 .properties(width=80, height=120)
 .repeat(column=numerical_columns, row=numerical_columns))
```

[47]: alt.RepeatChart(…)

Unfortunately, these plots are saturated, so although we can see that there might be some correlative relationships, we should remake this plot as a 2D histogram heatmap.

[48]:
```
# Scroll right on the plot to see more columns
(alt.Chart(trees_df)
 .mark_rect().encode(
    alt.X(alt.repeat('column'), type='quantitative', bin=alt.Bin(maxbins=30)),
    alt.Y(alt.repeat('row'), type='quantitative', bin=alt.Bin(maxbins=30)),
    alt.Color('count()', title=None))
 .properties(width=110, height=110)
 .repeat(column=numerical_columns, row=numerical_columns)).
 ↪resolve_scale(color='independent')
```

[48]: alt.RepeatChart(…)

From the above heatmaps, we find that diameter and height might have a positive relationship when diameter is less than 25 inches. Also,we can learn that civic number and block number are related to longitude and latitude and it provides some interesting aspects related to geographic distribution.

Besides, visualizing the counts of all categorical columns helps us understand the data.Considering some columns have too many values and here we just select a subset of categorical columns to explore.

[49]:
```
categorical_columns =␣
 ↪['street_side_name','curb','neighbourhood_name','root_barrier']
(alt.Chart(trees_df)
 .mark_bar().encode(
    alt.X('count()'),
    alt.Y(alt.repeat(), type='nominal', sort='x',title=''))
 .properties(width=80, height=200)
 .repeat(categorical_columns))
```

[49]: alt.RepeatChart(…)

We learn that some distributions are interesting such as how trees were planted in different street sides and neighbourhoods.Now we are going to explore more fun aspects of the data further in the following exploratory visualizaions.

1.4 Exploratory Visualizations

After the above early data analysis,we are going to keep exploring and focus on fun facts about tree distribution in the report. Some of these are inspired by the quick and dirty EDA plots in the introduction part.Some columns of interest are date_planted,neighbourhood_name,diameter,height_range_id and street_side_name.

1.4.1 Question 1: How is the distribution of trees evolved over the years through different neighbourhoods in Vancouver?

Since this question is related to the time,we'd better drop the missing values of date_planted and creat a new column of year_planted.

[50]:

```python
trees_df = trees_df.assign(year_planted=(trees_df['date_planted'].dt.year.
 ↪astype('Int64')))
trees_with_date_df = trees_df[trees_df['date_planted'].notna()]
trees_with_date_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2363 entries, 0 to 4998
Data columns (total 22 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Unnamed: 0          2363 non-null   int64
 1   std_street          2363 non-null   object
 2   on_street           2363 non-null   object
 3   species_name        2363 non-null   object
 4   neighbourhood_name  2363 non-null   object
 5   date_planted        2363 non-null   datetime64[ns]
 6   diameter            2363 non-null   float64
 7   street_side_name    2363 non-null   object
 8   genus_name          2363 non-null   object
 9   assigned            2363 non-null   object
 10  civic_number        2363 non-null   int64
 11  plant_area          2328 non-null   object
 12  curb                2363 non-null   object
 13  tree_id             2363 non-null   int64
 14  common_name         2363 non-null   object
 15  height_range_id     2363 non-null   int64
 16  on_street_block     2363 non-null   int64
 17  cultivar_name       1678 non-null   object
 18  root_barrier        2363 non-null   object
 19  latitude            2363 non-null   float64
 20  longitude           2363 non-null   float64
 21  year_planted        2363 non-null   Int64
dtypes: Int64(1), datetime64[ns](1), float64(3), int64(5), object(12)
memory usage: 426.9+ KB
```

[51]:

```python
trees_per_year = (
    alt.Chart(trees_with_date_df)
    .mark_bar().encode(
        alt.X('year_planted:O',title='Year',scale=alt.Scale(zero=False)),
```

```
        alt.Y('count()',title='Number of Trees',scale=alt.Scale(zero=False)),
        tooltip=[alt.Tooltip("count()", title='Number of Trees'),
                 alt.Tooltip("year_planted", title="Year Planted")])
    .properties(width=500)
)
trees_per_year.properties(title="Fig 1. Number of trees planted each year from␣
↪1989-2019")
```

[51]: alt.Chart(…)

[52]:
```
# Using widget radio button to choose from top 5 and bottom 5 year ranking

year_top5=trees_with_date_df.groupby('year_planted').size().nlargest(n=5).index.
↪to_list()

year_bottom5=trees_with_date_df.groupby('year_planted').size().nsmallest(n=5).
↪index.to_list()

radiobuttons_year = alt.binding_radio(name='Year Ranking', options=␣
↪[year_top5,year_bottom5],
                                      labels=['Top 5','Bottom 5'])

select_top_or_bottom = alt.selection_single(
    fields=['year_planted'],
    bind={'year_planted': radiobuttons_year})

trees_per_year.add_selection(select_top_or_bottom).encode(
    opacity=alt.condition(select_top_or_bottom, alt.value(0.7), alt.value(0.
↪05)),
    text=alt.condition(select_top_or_bottom, 'year_planted', alt.
↪value('steelblue'))
).properties(title={
        "text" : "Fig 2. Ranking of number of trees planted each year from␣
↪1989-2019",
        "subtitle" : ["Click on the radio button to select the ranking option."]
    })
```

[52]: alt.Chart(…)

From the above interactive plot,we can easily find that most trees were planted in 1996,1998,2002,2004 and 2013.On the other hand, least trees were planted in 1989,1991,2016,2017 and 2018. We are going to find out more about trees planted in different neighbourhood over these years.

[53]:
```
neighbourhood_order = trees_with_date_df.groupby('neighbourhood_name').size().
↪sort_values().index.tolist()
```

```
neighbourhood_heatmap_plot = alt.Chart(trees_with_date_df).mark_rect().encode(
     alt.X('year_planted:O',title=None),
     alt.Y('neighbourhood_name',sort=neighbourhood_order,title='neighbourhood'),
     alt.Color('count()',title='Number of Trees')).
 ↪properties(width=200,height=410)

neighbourhood_bar_plot= alt.Chart(trees_with_date_df).mark_bar().encode(
alt.X('count()',title='Number of Trees 1989-2019'),
alt.Y('neighbourhood_name',sort=neighbourhood_order,title=None,scale=alt.
 ↪Scale(zero=True)),
color=alt.condition(alt.FieldOneOfPredicate('neighbourhood_name',␣
 ↪neighbourhood_order[:-3]),
                          alt.value('steelblue'),
                          alt.value('coral'))
)

neighbourhood_bar_2= alt.Chart(trees_df).mark_bar().encode(
alt.X('count()',title='Number of Trees All The Time'),
alt.Y('neighbourhood_name',sort='x',title=None,scale=alt.Scale(zero=True)),
color=alt.condition(alt.FieldOneOfPredicate('neighbourhood_name',␣
 ↪neighbourhood_order[:-3]),
                          alt.value('steelblue'),
                          alt.value('coral'))).properties(width=150,height=410)
# interactive

multi = alt.selection_multi(fields=['neighbourhood_name'], empty='all')

neighbourhood_heatmap = neighbourhood_heatmap_plot.encode(
opacity=alt.condition(multi, alt.value(0.8), alt.value(0.05))
).properties(
    selection=multi
).add_selection(multi)

neighbourhood_bar = neighbourhood_bar_plot.encode(
opacity=alt.condition(multi, alt.value(0.9), alt.value(0.1)),
tooltip='count()'
).properties(
    selection=multi,width=150,height=410
).add_selection(multi)

(neighbourhood_heatmap | neighbourhood_bar | neighbourhood_bar_2).
 ↪properties(title=alt.TitleParams(
    "Fig 3. Number of Trees planted each year through neighbourhoods from␣
 ↪1989-2019",
     subtitle = ["Click on a row of the heatmap to select the neighbourhood,"
```

```
                  ,"Or click on a bar to select the neighbourhood.
↪"],anchor='middle'))
```

[53]: alt.HConcatChart(…)

From the Fig 3, we learn that most trees were planted in Renfrew-Collingwood,Kensington-Cedar Cottage and Hastings-Sunrise over the years.We find those neighbourhoods which planted most trees over the years are also the areas with most trees nowadays.

Besides,we would like to make some observations about the distribution of tree heights over the years as a bonus to question 1.

[54]:
```
median_line = alt.Chart(trees_with_date_df).mark_line().encode(
alt.X('year_planted:Q',title=None),
alt.Y('median(height_range_id)',title='Median height range id')
)
median_points = alt.Chart(trees_df).mark_circle(size=60).encode(
alt.X('year_planted',title=None),
alt.Y('median(height_range_id)')
)

average_line = alt.Chart(trees_with_date_df).mark_line().encode(
alt.X('year_planted:Q',title=None),
alt.Y('mean(height_range_id)',title='Average height range id')
)
average_points = alt.Chart(trees_df).mark_circle(size=60).encode(
alt.X('year_planted',title=None),
alt.Y('mean(height_range_id)')
)

height_per_year = (median_line + median_points).properties(width=350,height=200
) | (average_line + average_points).properties(width=350,height=200)

height_per_year.properties(title=alt.TitleParams(
    "Fig 4. Trees planted in 1991 are ourliers in average height",
    subtitle = ["Median/average height of trees is going down obviously since␣
↪2005 "],anchor='middle'))
```

[54]: alt.HConcatChart(…)

[55]:
```
trees_1991_df =␣
↪trees_with_date_df[trees_with_date_df['year_planted']==1991][['genus_name','species_name','
trees_1991_df.sort_values(['height_range_id'],ascending=False)
```

[55]:
```
     genus_name species_name  year_planted  diameter  height_range_id  \
2876    QUERCUS   ACUTISSIMA          1991     19.50                7
2677    QUERCUS   ACUTISSIMA          1991     16.50                6
1687    QUERCUS      PHELLOS          1991     24.00                5
```

```
3400     QUERCUS        PHELLOS           1991      20.00                5
569        TILIA   EUCHLORA    X          1991      10.50                4
1799       TILIA   EUCHLORA    X          1991      15.00                4
2808    FRAXINUS       OXYCARPA           1991      17.50                4
4671        ACER      CAMPESTRE           1991       8.00                4
1984      PRUNUS     CERASIFERA           1991      11.50                3
2337      PRUNUS     CERASIFERA           1991      10.00                3
4398      PRUNUS     CERASIFERA           1991      10.00                3
2971       TILIA        CORDATA           1991      13.00                2
4271       PYRUS     CALLERYANA           1991       9.00                2
4379      PRUNUS     CERASIFERA           1991      12.00                2
4517       TILIA        CORDATA           1991       8.75                2

            neighbourhood_name
2876  Kensington-Cedar Cottage
2677  Kensington-Cedar Cottage
1687  Kensington-Cedar Cottage
3400  Kensington-Cedar Cottage
569        Renfrew-Collingwood
1799          Hastings-Sunrise
2808       Renfrew-Collingwood
4671                  Downtown
1984          Hastings-Sunrise
2337  Kensington-Cedar Cottage
4398  Kensington-Cedar Cottage
2971                Riley Park
4271          Hastings-Sunrise
4379       Renfrew-Collingwood
4517                Riley Park
```

From the figure 4, we find that average trees planted in 1991 are typically tallest.I am curious about this and explore some more in it. Remember in figure 2, we have learned that 1991 is among the bottom 5 in trees planted per year and only 15 trees were planted. Besides, we find around 4 trees planted are very big and tall from the above tress_1991_df.These giant trees are almost 1/3 of the total trees planted. These might be the reasons why trees planted in 1991 are outliers in average height.

1.4.2 Question 2: How do trees differ among street sides in Vancouver?

To answer this question, we'll explore the differences among the strees sides.

[56]:
```
street_trees_bar = alt.Chart(trees_df).mark_bar().encode(
alt.X('count()',title='Number of Trees'),
alt.Y('street_side_name',title='Street Side'),
alt.Color('street_side_name',title='Street Side'),
tooltip='count()'
).properties(width=350,height=100)
```

```
street_trees_bar=street_trees_bar + street_trees_bar.mark_text(align='left',␣
↪dx=2).encode(text=alt.Text('count()'))

street_trees_height_bar = alt.Chart(trees_df).mark_bar().encode(
alt.X('mean(height_range_id)',title='Average height range id'),
alt.Y('street_side_name',title=None),
alt.Color('street_side_name')
).properties(width=320,height=100)

street_trees_bars = (street_trees_bar | street_trees_height_bar).
↪properties(title=alt.TitleParams(
    "Fig 5. Trees evenly distributed on both sides of the streets",
     subtitle = ["Trees in the middle of the streets are less and shorter",
                 "Trees in the bike area are the least and␣
↪shortest"],anchor='middle'))

street_trees_bars
```

[56]:
```
alt.HConcatChart(…)
```

We find that trees evenly planted on both sides of the street are bigger and taller than those planted in the middle of the street. Trees are usually smallest especially in the bike area. It makes sense when we are looking at the trees on the street we usually feel the same way as the above plot shows us.

1.4.3 Question 3: Which neighbourhood is surrounded by the most giant trees in Vancouver?

Now we are exploring the most wonderful neighbourhoods where there are most aboundant giant trees.Considering the exploration of neighbourhoods, it would be more fun and efficient to show the messages by using maps.

[57]:
```
url_geojson = 'https://raw.githubusercontent.com/UBC-MDS/exploratory-data-viz/
↪main/data/local-area-boundary.geojson'
```

[58]:
```
data_geojson_remote = alt.Data(url=url_geojson, format=alt.
↪DataFormat(property='features',type='json'))

data_geojson_remote
```

[58]:
```
Data({
  format: DataFormat({
    property: 'features',
    type: 'json'
  }),
  url: 'https://raw.githubusercontent.com/UBC-MDS/exploratory-data-
viz/main/data/local-area-boundary.geojson'
})
```

```
})
```

[59]:
```
vancouver_map = alt.Chart(data_geojson_remote).mark_geoshape(
    color = 'gray', opacity= 0.5, stroke='white').encode(
).project(type='identity', reflectY=True)

vancouver_map.properties(title='Vancouver Base Map')
```

[59]:
```
alt.Chart(...)
```

[22]:
```
alt.data_transformers.enable('default', max_rows=1000000)
```

[22]:
```
DataTransformerRegistry.enable('default')
```

[23]:
```
mean_df = trees_df.groupby('neighbourhood_name'
                       ).mean().reset_index(
).rename(columns={'neighbourhood_name':'name'})[['name',
                                                  'diameter',
                                                  'height_range_id',
                                                  'latitude',
                                                  'longitude']]
mean_df
```

[23]:
```
                         name   diameter  height_range_id   latitude  \
0               Arbutus-Ridge  12.598571         2.800000  49.247602
1                    Downtown   7.480117         2.380117  49.280103
2           Dunbar-Southlands  16.078115         3.335463  49.244179
3                    Fairview  13.910821         3.298507  49.263516
4          Grandview-Woodland  12.603627         2.689119  49.272888
5            Hastings-Sunrise  12.185441         2.620588  49.275697
6    Kensington-Cedar Cottage  12.005600         2.704000  49.246254
7                  Kerrisdale  13.904960         3.023810  49.226844
8                   Killarney  10.030000         2.570000  49.222677
9                   Kitsilano  15.080855         3.382900  49.264396
10                    Marpole  12.419492         2.338983  49.212498
11             Mount Pleasant  13.401759         2.718593  49.262018
12                   Oakridge  10.236263         2.210526  49.226898
13        Renfrew-Collingwood  10.308724         2.356771  49.245873
14                 Riley Park  12.676829         2.743902  49.246407
15                Shaughnessy  14.162611         3.265487  49.246301
16               South Cambie  12.402542         2.661017  49.246633
17                 Strathcona  12.447333         2.920000  49.277899
18                     Sunset  11.147249         2.427184  49.221903
19        Victoria-Fraserview  10.456678         2.318493  49.221300
20                   West End  12.842520         2.897638  49.286638
21           West Point Grey  13.256250         2.954545  49.264610
```

```
     longitude
0  -123.161488
1  -123.118926
2  -123.183673
3  -123.131065
4  -123.064619
5  -123.041677
6  -123.074008
7  -123.154462
8  -123.037422
9  -123.163117
10 -123.129457
11 -123.097173
12 -123.125679
13 -123.040694
14 -123.102549
15 -123.139998
16 -123.120934
17 -123.088975
18 -123.092476
19 -123.063233
20 -123.134735
21 -123.204269
```

[24]:
```
diameter_choropleth = alt.Chart(data_geojson_remote).mark_geoshape().
 ↪transform_lookup(
    lookup='properties.name',
    from_=alt.LookupData(mean_df, 'name', ['diameter', 'name'])).encode(
    color='diameter:Q',
    tooltip='name:N').project(type='identity', reflectY=True
                              ).properties(width=300,title="Neighbourhood Trees␣
 ↪Diameter Map")

height_choropleth = alt.Chart(data_geojson_remote).mark_geoshape().
 ↪transform_lookup(
    lookup='properties.name',
    from_=alt.LookupData(mean_df, 'name', ['height_range_id','name'])).encode(
    color=alt.Color('height_range_id:Q', title='Height range id'),
    tooltip='name:N').project(type='identity', reflectY=True
                              ).properties(width=300,title="Neighbourhood Trees␣
 ↪Height Map")


(diameter_choropleth| height_choropleth).resolve_scale(color='independent')
```

[24]: alt.HConcatChart(…)

From the above maps we find that most areas in the Vancouver westside are the great neighbourhoods where there are most giant trees. It is facinating that these neighbourhoods are usually the most prestigious areas and have the highest housing price as well. To explore more about this interesting relationship,I am going to import a dataset about the benchmark price updated in 2022 Apr. through all the Vancouver communities.

[25]:
```
# import a dataset related to home prices in 2022 Apr.
home_price_df = pd.read_csv('vancouver_benchmark_price.csv')
home_price_df = home_price_df.rename(columns={'neighbourhood_name':'name'})
home_price_df
```

[25]:

	Unnamed: 0	name	benchmark_price
0	0	Arbutus-Ridge	3884000
1	1	Downtown	769200
2	2	Dunbar-Southlands	3630650
3	3	Fairview	937400
4	4	Grandview-Woodland	2024000
5	5	Hastings-Sunrise	1917800
6	6	Kensington-Cedar Cottage	2019400
7	7	Kerrisdale	3619900
8	8	Killarney	2098300
9	9	Kitsilano	2608000
10	10	Marpole	2848100
11	11	Mount Pleasant	2230550
12	12	Oakridge	4015100
13	13	Renfrew-Collingwood	1790500
14	14	Riley Park	2609200
15	15	Shaughnessy	5441300
16	16	South Cambie	4829600
17	17	Strathcona	1705500
18	18	Sunset	1676000
19	19	Victoria-Fraserview	1823900
20	20	West End	746900
21	21	West Point Grey	3793600

[26]:
```
hover = alt.selection_single(fields=['name'], on='mouseover')

price_choropleth = (alt.Chart(data_geojson_remote).mark_geoshape().
↪transform_lookup(
    lookup='properties.name',
    from_=alt.LookupData(home_price_df, 'name', ['benchmark_price', 'name']))
.encode(
    color=alt.Color('benchmark_price:Q', title='Benchmark price'),
    opacity=alt.condition(hover, alt.value(1), alt.value(0.1)),
    tooltip=[alt.Tooltip('name:N', title='Neighbourhood'), alt.
↪Tooltip('benchmark_price:Q', format= '$.3s',title='Benchmark price')])
.add_selection(hover)
```

```
.project(type='identity', reflectY=True)
.properties(height=150, width=300))

height_choropleth = (height_choropleth.encode(
    opacity=alt.condition(hover, alt.value(1), alt.value(0.1)),
    tooltip=[alt.Tooltip('name:N', title='Neighbourhood'),
             alt.Tooltip('height_range_id:Q',title='Height range id',format='.
 ↪2s')]
    ).add_selection(hover).project(type='identity', reflectY=True).
 ↪properties(height=150, width=300,title="")
                    )

(price_choropleth | height_choropleth).resolve_scale(color='independent').
 ↪properties(title=alt.TitleParams(
    "Vancouver westside prestigious neighbourhoods surrounded by the most giant␣
 ↪trees",anchor='middle'))
```

[26]: alt.HConcatChart(…)

This is amazing that most prestigious neighbourhoods are abundant with moset giant trees. These tall trees are providing the best privacy to the homes and also bringing the most wonderful views. They contribute a lot to the beautiful and peaceful environments where are full of around 4 to 5 millions of dollars houses.

Furthermore,let's take a look at how the trees are distributed in these top neighbourhoods by subplots.

[27]:
```
top_neighbourhood_trees_df = trees_df[trees_df['neighbourhood_name'].
 ↪isin(['Kitsilano',␣
 ↪'Dunbar-Southlands','Fairview','Shaughnessy','Kerrisdale'])]


alt.Chart(top_neighbourhood_trees_df).mark_bar().encode(
    alt.X('diameter', bin=alt.Bin(maxbins=30)),
    alt.Y('count()'),
    alt.Color('neighbourhood_name')
).properties(width=200, height=150
).facet('neighbourhood_name',columns=3)
```

[27]: alt.FacetChart(…)

[28]:
```
alt.Chart(top_neighbourhood_trees_df).mark_bar().encode(
    alt.X('height_range_id', bin=alt.Bin(maxbins=30)),
    alt.Y('count()',title='Number of trees'),
    alt.Color('neighbourhood_name',title=None)
).properties(width=200, height=150
).facet('neighbourhood_name',columns=3)
```

[28]: alt.FacetChart(…)

From these subplots, Fairview has the most fairly distributed trees of different sizes just like its name “Fairview”! What a fun fact! Let’s explore more in the fairview map.

[29]:
```
url_geojson_fair = 'https://raw.githubusercontent.com/UBC-MDS/
↪exploratory-data-viz/main/data/vancouver_neighbourhoods/fairview.geojson'
```

[30]:
```
data_geojson_remote_fair = alt.Data(url=url_geojson_fair, format=alt.
↪DataFormat(property='features',type='json'))
```

[31]:
```
fairview_map = alt.Chart(data_geojson_remote_fair).mark_geoshape(
    color = 'gray', opacity= 0.5, stroke='white').encode(
).project(type='identity', reflectY=True)

df_fair = trees_df[trees_df['neighbourhood_name'] == 'Fairview']
```

[32]:
```
points_fair = alt.Chart(df_fair).mark_circle(color='green').encode(
    longitude='longitude',
    latitude='latitude',size=alt.Size('diameter',legend=alt.
↪Legend(orient='bottom'),title='Diameter(in)',scale=alt.
↪Scale(range=(1,100)))).project(type= 'identity', reflectY=True)

fairview_trees_map = (fairview_map + points_fair).properties(title="Fairview␣
↪has a fair view of trees")
fairview_trees_map
```

[32]: alt.LayerChart(…)

We can see from the above map that there are evenly distibuted trees in Fairview. Trees are distributed here not only geographically evenly, but also physically evenly with similar tree sizes.It is really fair to view these trees. “Fairview” has the best fair view of trees!

1.4.4 Question 4:Which range of tree size is most prominent in Vancouver?

Let’s take a look at the tree sizes distribution.

[33]:
```
tree_size_points =(alt.Chart(trees_df).mark_circle(size=500).encode(
     alt.X('height_range_id', type='quantitative', bin=alt.
↪Bin(maxbins=30),title='Height range id(binned)'),
     alt.Y('diameter', type='quantitative', bin=alt.
↪Bin(maxbins=30),title='Diameter(binned)'),
     color = alt.Color('count()',legend=alt.
↪Legend(orient='left',offset=3),title=None),
     size = alt.Size('count()',legend=None),
     tooltip = alt.Tooltip('count()',title="Number of trees")
     ).properties(width=510, height=310,
```

```
    title='Tree size of diameter less than 5 inches is most prominent in␣
↪Vancouver')
    )
tree_size_points
```

[33]: alt.Chart(…)

Using both the colour and marker size to indicate the count creates an effective visualization in the above plot.We can easily learn that diameter less than 5 inches and height range between 1 and 1.5 are the most poluplar size of the trees in Vancouver. The trees with the diameter between 5 and 10 inches and height range between 2 and 2.5 go to the second place.

Now let's explore which genura are most prominent.

[34]:
```
click = alt.selection_multi()
genus_top5_list=trees_with_date_df.groupby('genus_name').size().nlargest(n=5).
↪index.to_list()

genus_top5_df=trees_df[(trees_df['genus_name'].isin (genus_top5_list))]

bars = (alt.Chart(genus_top5_df).mark_bar().encode(
    alt.X('count()', title='Number of trees'),
    alt.Y('genus_name', title='Genus',sort='x'),
    alt.Color('genus_name', sort='-x',title=None),
    opacity=alt.condition(click, alt.value(0.9), alt.value(0.2)))
.add_selection(click)).properties(width=300)

brush = alt.selection_interval()
click = alt.selection_multi(fields=['genus_name'])
points = (alt.Chart(genus_top5_df).mark_point().encode(
    alt.X('height_range_id:Q', title='Height',scale=alt.Scale(zero=False)),
    alt.Y('median(diameter):Q', title='Median Diameter'),
    color=alt.condition(brush, 'genus_name:N', alt.value('lightgray')),
    size=alt.Size('count()',title="Number of trees",
    legend=alt.Legend(orient='bottom',offset=3)),tooltip='count()'
).add_selection(brush)).properties(width=510, height=310,title={
        "text" : "Trees get less when size going up",
        "subtitle" : ["Drag with the mouse to filter data,","click legend to␣
↪select genus."]
    })


bars = bars.add_selection(click)
points = points.encode(opacity=alt.condition(click, alt.value(0.9), alt.value(0.
↪01)))
click_legend = alt.selection_multi(fields=['genus_name'], bind='legend')
```

```
points = points.encode(opacity=alt.condition(click_legend, alt.value(0.9), alt.
↪value(0.02))).add_selection(click_legend)

bars = bars.encode(opacity=alt.condition(click_legend, alt.value(0.9), alt.
↪value(0.02)))
(points & bars).add_selection(click_legend)

bars = bars.transform_filter(brush)

bar_chart_max = genus_top5_df.groupby('genus_name').size().max()

bars = bars.encode(alt.X('count()', title='Number of Trees',scale=alt.
↪Scale(domain=[0, bar_chart_max]))
                  ).properties(title="Most prominent genura")

(points & bars).add_selection(click_legend)
```

[34]: alt.VConcatChart(…)

From the above bar we find that Acer is most prominent genus and Prunus goes to the 2nd place.It's goes very well with our common sense because maple trees in Acer genus are most prominent tress in Canada.Also, Vancouver is famous for the beautiful cherry trees which are in Prunus genus. Besides, we find that trees are less and less when the tree size is going up through the above interactive plot.

Lastly, we focus on the relationship between the neighbourhoods and tree sizes by using the map.

[35]:
```
points_genus = alt.Chart(trees_df).mark_circle(size=10,color='green').encode(
    longitude='longitude',
    latitude='latitude' ,tooltip='neighbourhood_name',
    size=alt.Size('diameter:Q',title='Diameter(in)',legend=alt.
↪Legend(orient='left'),scale=alt.Scale(range=(1,150))
    )
    ).project(type= 'identity', reflectY=True).add_selection(click_legend)

slider_diameter = alt.binding_range(name='Diameter(inch) bigger than the slider␣
↪value ')

select_diameter = alt.selection_single(
    fields=['diameter'],
    bind=slider_diameter)

points_genus = points_genus.encode(
    opacity=alt.condition(alt.datum.diameter > select_diameter.diameter, alt.
↪value(0.7), alt.value(0.01))
).add_selection(select_diameter)
```

```
vancouver_genus_map = (vancouver_map + points_genus
                    ).properties(width=500,height=280,
                  title={
        "text" : "Most giant trees are centralized in Vacouver westside",
        "subtitle" : ["Highlighting trees bigger than a slider value",
                     "Shaughnessy wins when trees bigger than 60 inches"]
    })

vancouver_genus_map
```

[35]: alt.LayerChart(…)

2 Discussion

Vancouver is a beautiful city and is known for its perfect balance of city and nature. Trees here play a key role in this balance. In our analysis, we have been focused on 4 different aspects through our 4 questions of interests about fun facts of the distribution of trees.

In Fig 1, we see different number of trees planted in each year. To see it more clearly we rank them by 2 groups: top 5 and bottom 5 in Fig 2. Then we find some fun facts by connecting them to average height of trees planted in each year and to neighbourhood trees planted between 1989 and 2019. One fun fact is that average trees planted in 1991 are typically the tallest. we have learned that 1991 is among the bottom 5 ranking in Fig 2 and only 15 trees were planted. Besides, we found around 4 trees planted are very big and tall from the table tress_1991_df. These giant trees are almost 1/3 of the total trees planted that year. These might be the reasons why trees planted in 1991 are outliers in average height. Another fun fact is that those neighbourhoods which have planted most trees between 1989 and 2019 are also the ones with most trees nowadays and interestingly, they are all in the the eastside of the city. It might reflect that eastside is likly to be in the developping stage having more newly planned city areas, thus having more spaces for newly planted trees in recent years. But how about the westside? We will cover it later.

After the exploration of the time related aspect, we then go to find some interesting rules in the distribution of trees on street sides. In Fig 5, we have seen that trees evenly planted on both sides of the street are bigger and taller than those planted in the middle of the street. Trees are usually smallest especially in the bike area. The same rule goes to the numbers of trees. Trees on both sides of streets are dramatically more than those in the middle. There are even less in the bike area. This might reflect the reality when we walk on the streets. The evenly distribution on both sides of streets brings us a beauty of balance, while the unequal distribution among other different street sides brings us a beauty of contrast.

Furthermore, we have loaded maps and other data resources to explore our most exciting fun facts about the relationship between the distribution of trees and the home prices or land values of the communities. By comparing the maps of tree sizes and the map of the neighbourhood home benchmark price(updated in 2022 Apr.),we discovered that there might be some connections between the giant trees and the prestigous neighbourhoods in westside of Vancouver. The neighbourhoods where there are lots of mansions are usually the areas surrounded with most epic trees. Those trees are much older and bigger than trees planted in the eastside, though the number is less. Besides, we have found the probablly most fun fact that one neighbourhood “Fairview” has a really fair view of trees because the differences of tree sizes are not much and they are evenly distributed over the area. We can see it clearly in the Fairview_trees_map.

Last but not least, we focus on the tree size exploration. We see that trees diameter less than 5 inches and height range id between 1 and 1.5 are the most poluplar tree size in Vancouver. 5 to 10 inches go to the 2nd place and as size is going up, the number is going down. Also, we take a close look at the top 5 genus group and find that most of them share the same size rule that tree sizes less than 10 inches are most popular except for Prunus. In Prunus, most prominent diameter size is bwtween 10 to 20 inchese. As we know, Prunus is a genus including the fruits plums, cherries, peaches and etc. They have beautiful blossoms and grow fast, so the popular size might tend to be bigger than other genus. We save the best for the last. Let's see the magic map which shows that the giant trees are centralized in westside of the city especially when the diameter bigger than 45 inches. When the diameter goes bigger than 60 inches, we make a fascinating obervation that the most prestigious neighbourhood "Shaughnassy" where the benchmark home price is around 5.5M dollars owns the most giant trees. Incredibly, Shaughassy is the champion of both land values and tree sizes!

This has been a very interesting dive into the Vancouver trees! We have found so many fun facts and it is absolutely a wonderful journey to experience the beauty of data visallization.

2.1 Dashboard

[36]:
```
# Resizing and reorganizing first 4 plots related to plant year to make them␣
↪accommodate to each other

select_year_click = alt.selection_single(fields=['year_planted'])   # On mouse␣
↪click

# plot 1: trees_per_year_click
trees_per_year_click = (
    trees_per_year.encode(
        opacity=alt.condition(select_year_click, alt.value(0.9), alt.value(0.
↪2)))
    .properties(height=200, width=350)
    .add_selection(select_year_click)
    .properties(title={
        "text" : "Number of trees planted each year from 1989 to 2019",
        "subtitle" : ["Click on a bar to select the year"]
    })
)

# plot 2: average_height_per_year
average_height_per_year = ((average_line + average_points
    .encode(
        opacity=alt.condition(select_year_click, alt.value(0.9), alt.value(0.
↪2)),
        stroke=alt.condition(select_year_click, alt.value('black'), alt.
↪value('#ffffff')),
```

```
        tooltip=[alt.Tooltip("year_planted:Q", title="Year")],
        color=alt.value('coral'))
                          ).properties(title="Trees planted in 1991 are␣
↪ourliers in average height",
                     height=200, width=450).add_selection(select_year_click))

# plot 3: neighbourhood_tree_heatmap
neighbourhood_tree_heatmap = neighbourhood_heatmap_plot.encode(
                        opacity=alt.condition(select_year_click, alt.value(0.
↪9), alt.value(0.1)),
                        color=alt.Color('count()',legend=alt.
↪Legend(orient='bottom', title='Number of trees'), title=None)
                        ).add_selection(select_year_click
                        ).properties(width=350,height=300,title={
        "text" : 'Neighbourhoods trees heatmap from 1989 to 2019',
        "subtitle" : ["Click on a column to select the year"]
    })

# plot 4: neighbourhood_tree_bar
neighbourhood_tree_bar = neighbourhood_bar_plot.
↪transform_filter(select_year_click
                        ).add_selection(select_year_click
                        ).properties(width=350,height=300,title='Vancouver␣
↪eastside planted most trees from 1989 to 2019')

neighbourhood_bar_chart_max = trees_with_date_df.groupby('neighbourhood_name').
↪size().max()   # fixing the extent of the x-axis

neighbourhood_tree_bar = neighbourhood_tree_bar.encode(
                            alt.X('count()', title='Number of Trees'
                            ,scale=alt.Scale(domain=[0,␣
↪neighbourhood_bar_chart_max])))

# Figure layout
(trees_per_year_click | average_height_per_year) & (neighbourhood_tree_heatmap␣
↪| neighbourhood_tree_bar
) & street_trees_bars & (fairview_trees_map.properties(width=400,height=300
).resolve_scale(size='independent') | (height_choropleth & price_choropleth).
↪resolve_scale(color='independent'
).properties(title='Vancouver prestigious westside surrounded by most giant␣
↪trees')
) & ((tree_size_points.properties(width=400,height=200) & vancouver_genus_map.
↪properties(width=400).resolve_scale(size='independent')
) | ((points.properties(width=230) & bars.properties(width=230)).
↪add_selection(click_legend)).resolve_scale(size='independent'))
```

```
[36]: alt.VConcatChart(…)
```

2.2 References

Not all the work in this notebook is original. Parts that were borrowed from other resources are as follows:

2.2.1 Resources used

- Programming in Python for Data Science sample final project for inspiration
- Data Source-Vancouver Street Trees
- Data Source-Vancouver Benchmark Home Prices Through Neighbourhoods
- Altair documentation including, but not limited to,
  - Compound Charts
  - Multiple Interactions
  - Customizing Visualizations
- Altair Ally:API Reference-nan
- Pandas:API Reference-nlargest
- Image of Shaughnassy street from the Faith Wilson
- Wikipedia article on the Ecology of Vancouver

Image credit to this website.