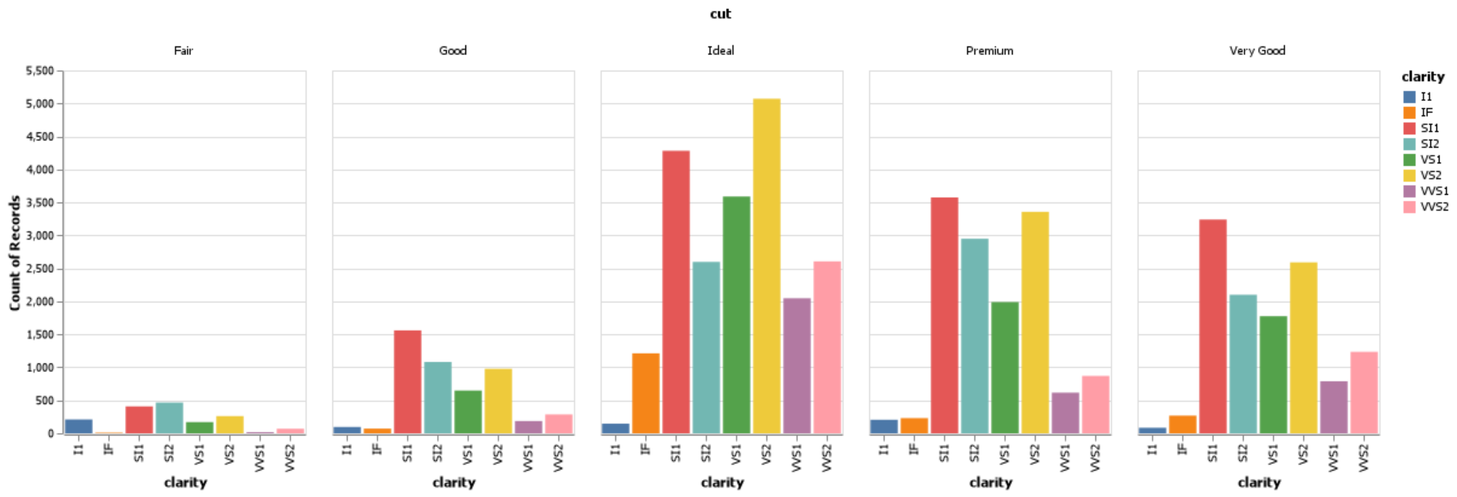
Week 1 Introduction

Cameron Hansen

Elevator pitch

The Introduction week is where the class is getting all our environments set up, so the following document is proof that the setup was successful and the readings are completed. The provided Altair chart and Python script were used in note-taking during the readings.

TECHNICAL DETAILS



The following chart is of the diamonds dataset. Using the various methods discussed in the readings this shows the number of records for each of the cuts seperated by clarity. Clarity is also color coded in the plot.dogs reate a python script and use VS Code to create the example Altair chart in the assigned readings (note that you have to type chart to see the Altair chart after you run the code). Save your Altair chart for submission.

1. Finish the readings and come to class prepared with any questions to get your environment working smoothly.

2. Create a python script and use VS Code to create the example Altair chart in the assigned readings (note that you have to type chart to see the Altair chart after you run the code). Save your Altair chart for submission.

APPENDIX A (PYTHON SCRIPT)

```
# %%
import pandas as pd
import altair as alt
import altair_saver
# %%
alt.data_transformers.enable("json")


# %%
data = pd.read_csv("https://github.com/byuidatascience/data4python4ds/raw/master/data-raw/mpg/mp
# %%
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy'
    )
    .mark_circle()
)

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy'
    )
    .mark_point()
)
data.info()

chart = (alt.Chart(data)
    .encode(
        x='cyl',
        y='hwy'
    )
    .mark_point()
)

chart = (alt.Chart(data)
    .encode(
        x='drv',
        y='class'
    )
    .mark_point()
)

# %%
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        color='class'
    )
    .mark_circle()
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        size='class'
    )
    .mark_circle()
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        opacity='class'
    )
    .mark_circle()
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        shape='class'
    )
    .mark_point()
)
# %%
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        color=alt.value("blue")
    )
    .mark_circle()
)
# %%
data.dtypes

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        color='cty'
    )
    .mark_point()
)

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        size='cty'
    )
    .mark_point()
)

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        shape='cty'
    )
    .mark_point()
)

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        color='displ'
    )
    .mark_point()
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
        stroke='cty'
    )
    .mark_point()
)
# %%
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
    )
    .mark_point()
).facet(
    facet="class",
    columns = 2
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
    )
    .mark_point()
).facet(
    row="class",
    column = 'manufacturer'
)
```

```
 # %%
 #Exercise 3.5

chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
    )
    .mark_point()
).facet(
    facet="cty",
)
chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
    )
    .mark_point()
).facet(
    row="class",
    column = 'manufacturer'
)

chart = (alt.Chart(data)
  .encode(
    x = "displ",
    y = "hwy"
    )
  .mark_circle()
  .facet(column = "drv"))

(alt.Chart(data)
  .encode(
    x = "displ",
    y = "hwy"
    )
  .mark_circle()
  .facet(row = "cyl"))
 # %%
 #Exercise 3.6 USE https://altair-viz.github.io/user_guide/marks.html for reference

 chart = (alt.Chart(data)
    .encode(
        x='displ',
        y='hwy',
    )
    .transform_loess('displ','hwy')
    .mark_line()
 )
```

```
chart = (alt.Chart(data)
    .encode(
        x='displ:O',
        y='hwy',
    )
    .mark_boxplot()
 )

chart = (alt.Chart(data)
    .encode(
        x='displ:O',
        y='hwy:Q',
        color = 'class:N'
    )
    .mark_area()
 )

chart = (alt.Chart(data)
    .encode(
        x='displ:O',
        y='hwy:Q',
        color = alt.Color("class", legend=None)
    )
    .mark_point()
 )

# %%
# 3.7 Diamonds Statiscal Transformations
url = "https://github.com/byuidatascience/data4python4ds/raw/master/data-raw/diamonds/diamonds.c
diamonds = pd.read_csv(url)

diamonds['cut'] = pd.Categorical(diamonds.cut,
  ordered = True,
  categories =  ["Fair", "Good", "Very Good", "Premium", "Ideal" ])

diamonds['color'] = pd.Categorical(diamonds.color,
  ordered = True,
  categories =  ["D", "E", "F", "G", "H", "I", "J"])


diamonds['clarity'] = pd.Categorical(diamonds.clarity,
  ordered = True,
  categories =  ["I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"])


chart = (alt.Chart(diamonds)
  .encode(
    x = "cut",
    y = alt.Y("count():Q")
    )
  .mark_bar()
  .properties(width = 400))

#%%
# Positional Adjustments 3.8
# Stroke = outline Color = Fill

# position = "identity"
chart = (alt.Chart(diamonds)
    .encode(
      x = "cut",
      y = alt.Y("count()", stack=None),
      color = "clarity",
      stroke = alt.value("none")
      )
    .mark_bar()
    .properties(width = 200))

# Position = "fill"
chart = (alt.Chart(diamonds)
    .encode(
      x = "cut",
      y = alt.Y("count()", stack='normalize'),
      color = "clarity",
      stroke = alt.value("none")
      )
    .mark_bar()
    .properties(width = 200))


# Position = "dodged"
chart = (alt.Chart(diamonds)
    .encode(
      x = "clarity",
      y = alt.Y("count()"),
      color = "clarity",
      column = 'cut'
      )
    .mark_bar()
    .properties(width = 200))



'''All charts follow this pattern
(alt.Chart(<DATA>)
  .encoding(

    )
  .facet(
    column = <FACET VARIABLE>
    )
  .<TRANFORM_FUNCTION>()
  .<MARK_FUNCTION>()
  .properties(
    width = ,
    height = 
  ))'''
# %%
chart.save("Week1.png")
# %%
chart
# %%
```