Pandas Cheatsheet

Imports

```
import pandas as pd
import altair as alt
import datetime
```

Creating DataFrames

```
df1 = pd.read_csv(  # From file
    'world_countries.csv')
df2 = pd.DataFrame({  # From Python dict
  'col0': [0, 1, 2],  'col1': [3, 4, 5],
  'col2': ['ab', 'cd', 'ef'],
  'col3': [datetime.datetime.now()] * 3})
```

Inspecting DataFrames

```
df1.head()      # First 5 rows
df1.tail()      # Last 5 rows
df1.columns     # Columns names
len(df1)        # Number of rows
df1.shape       # Number of rows and cols
df1.describe()  # Stats about each column
df1.info()      # Summary info
```

Summarizing columns

```
# Rename a column
df1 = df1.rename(
  columns={'Population': 'Pop'})
df1.Pop.sum()     # Sum
df1.Pop.mean()    # Average
df1.Pop.std()     # Standard deviation
df1.Pop.median()  # Median
df1.Pop.min()     # Minimum
df1.Pop.max()     # Maximum
```

Filtering rows

```
df1[5:11]      # Select rows 5 through 10
# Rows with Spain in the Country column
df1[df1.Country == "Spain"]
# Removing nulls
df1 = df1[~df1.Pop.isnull()]
# Convert strings to integers
df1.ConSal = df1.Pop.astype('int64')
# Booleans operators are &, | and ~
df1[(df1.Pop > 100) &
    ~(df1.Area.isnull())]
```

Column manipulations

```
# Arithmetic operations on columns
df2['col0'] + df2['col1']
# Even if they're strings
df2['col2'] + df2['col2']
# Create new column from the result
df2['col4'] = df2['col0'] / df2['col1']
# String methods and attributes can be
# accessed via .str.
df2['col2'].str.replace('a', 'b')
# And datetime methods and attributes
# via .dt.
df2.col3.dt.date
# Select just some columns from DataFrame
df1[['Country', 'Pop']]
```

Dealing with missing values

```
# Drop rows with any missing values
df1.dropna()
# Drop columns with any missing value
df1.dropna(axis=1)
# Fill missing values with 0s
df1.fillna(0)
# Fill missing values with ''
df1.fillna('')
```

Grouping

```
# Get the average salary for each country
df1.groupby('Country').agg(
    {'Pop': 'mean'})
# Get the average and minimum salary
df1.groupby('Country').agg(
    {'Pop': ['mean', 'min']})
# Keep grouping column as a column
df1.groupby(
    'Country', as_index=False).agg(
    {'Pop': ['mean', 'min']})
```

Miscellaneous

```
# Reorder from top salary to lowest
df1.sort_values('Pop',
                ascending=False)
# Remove a column
df1.drop(columns='Phones')
# Randomly select a sample of 45 rows
df1 = df1.sample(45)
```

Merging

```
df3 = pd.DataFrame(
    {'col5': ['ab', 'cd', 'ef'],
     'col6': [100, 200, 300]})
# Create a new DataFrame matching col2
# of df2 and col5 of df3.
df2.merge(df3, left_on='col2',
          right_on='col5')
```

Graphing

```
alt.Chart(df1).mark_point().encode(
  x='Country', y='Area', size='Pop',
  color='Birthrate')
```