Data Visualization

28 May 2018

20:10

With Pandas

* df.plot.bar()
* df.plot.line()
* df.plot.area()
* df.plot.hist()
* syntax : dataframe['column'].value\_counts().head(10).plot.bar()
* dataframe['ordinalcategory'].value\_counts.sort\_index.plot.bar()
* df.plot.scatter()
* df.plot.hex()
* df.plot.bar(stacked = TRUE)
* df.plot.line()

With Seaborn

* Bar Plot : sns.countplot() #good for nominal and small ordinal categorical data
* (Kernel Density Estimate)KDE Plot: sns.kdeplot() #good for interval data
* sns.jointplot #good for interval and some nominal categorical data.
* sns.violinplot() #good for interval data and some nominal categorical data

Example:

sns.jointplot(x='price', y='points', data=reviews[reviews['price'] < 100])

sns.jointplot(x='price', y='points', data=reviews[reviews['price'] < 100])

sns.boxplot(  
 x='variety',  
 y='points',  
 data=df )

Faceting with seaborn

* sns.FacetGrid() #good for data with atleast two categorical variables

Example:

df = footballers[footballers['Position'].isin(['ST', 'GK'])]

df = df[df['Club'].isin(['Real Madrid CF', 'FC Barcelona', 'Atlético Madrid'])]

g = sns.FacetGrid(df, row="Position", col="Club",

row\_order=['GK', 'ST'],

col\_order=['Atlético Madrid', 'FC Barcelona', 'Real Madrid CF'])

g.map(sns.violinplot, "Overall")

* sns.pairplot() #good for exploring most kinds of data

sns.pairplot(footballers[['Overall', 'Potential', 'Value']])

Multivariate Plotting

df.plot.scatter() #multivariate scatter plot

df.plot.box() #grouped box plot

sns.heatmap #heatmap

pd.plotting.parallel\_coordinates

sns.lmplot(x = ‘Value’, y = ‘Overall’, hue = ‘Position’, data = footballers.loc[‘position’].isin([‘ST’,’RW’,’LW’])],fit\_reg = Fasle, markers = [‘o’, ‘x’, ‘\*’])

sns.boxplot(x="Overall", y="Aggression", hue='Position', data=f)

f = (

footballers.loc[:, ['Acceleration', 'Aggression', 'Agility', 'Balance', 'Ball control']]

.applymap(lambda v: int(v) if str.isdecimal(v) else np.nan)

.dropna()).corr()

sns.heatmap(f, annot=True)

f = (

footballers.iloc[:, 12:17]

.loc[footballers['Position'].isin(['ST', 'GK'])]

.applymap(lambda v: int(v) if str.isdecimal(v) else np.nan)

.dropna()

)

f['Position'] = footballers['Position']

f = f.sample(200)

parallel\_coordinates(f, 'Position')

With plotly

* + Scatter Plot: go.Scatter()
  + Choropleth: go.Choropleth()
  + Heatmap: go.Heatmap()
  + Surface Plot: go.Surface()

From plotly.offline import init\_notebook\_mode(connected = True)

Import plotly.graph\_objs as go

Iplot([go.Scatter(x = column1, y = column2, mode = ‘markers’])

* + Histogram

iplot([go.Histogram2dContour(x=reviews.head(500)['points'],

y=reviews.head(500)['price'],

contours=go.Contours(coloring='heatmap')),

go.Scatter(x=reviews.head(1000)['points'], y=reviews.head(1000)['price'], mode='markers')])

* + Surface
* df = reviews.assign(n=0).groupby(['points', 'price'])['n'].count().reset\_index()
* df = df[df["price"] < 100]
* v = df.pivot(index='price', columns='points', values='n').fillna(0).values.tolist()
  + Chloropleths

df = reviews['country'].replace("US", "United States").value\_counts()

iplot([go.Choropleth(

locationmode='country names',

locations=df.index.values,

text=df.index,

z=df.values

)])

Introduction to grammar of graphics in Python

from plotnine import\*

df = top\_wines.head(1000).dropna()

(ggplot(df)) + aes(‘points’, ‘price’) + geom\_point()) + stat\_smooth() # adds a rergression line

(ggplot(df)) + aes(color = ‘points’) + aes(‘points’, ‘price’) + geom\_point()) + stat\_smooth() # adds colour

(ggplot(df)) + aes(color = ‘points’) + aes(‘points’, ‘price’) + geom\_point()) + stat\_smooth() + facet\_wrap(‘~variaety’) # adds colour

+geom\_bar() #barplot

+geom\_bin2d(bins = 20) #hexplot,2d histogram

+ggtitle(“something) #title for the plots