DataCraft

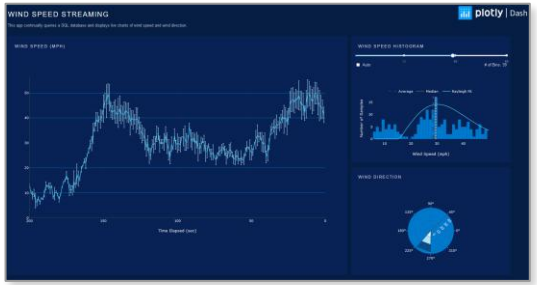
12

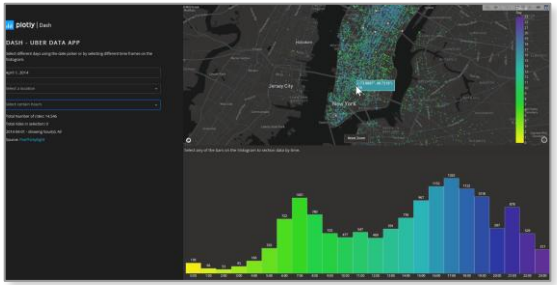
Interaktive Grafiken mit Plotly Dash

Plotly Dash ist ein Package für Python, dass interaktive Dashboards ermöglicht. Installation mit conda install dash.

Es gibt eine Version für Unternehmen (Dash Enterprise), aber auch eine Open Source Version.

In der Dash App Gallery findet man einige Beispiele, was mit diesem Framework möglich ist.





Eine erste App ist einfach zu erzeugen. Dazu legt man eine py-Datei mit folgendem Inhalt an:

```
from dash import Dash, html

app = Dash(__name__)
app.layout = html.H1(children="Willkommen zu Dash!")
app.run_server(debug=True)
```

Mit Start des Programms wird ein flask-Server auf dem PC gestartet, der unter http://127.0.0.1:8050 zu erreichen ist.

Beenden des Servers mit Strg + C

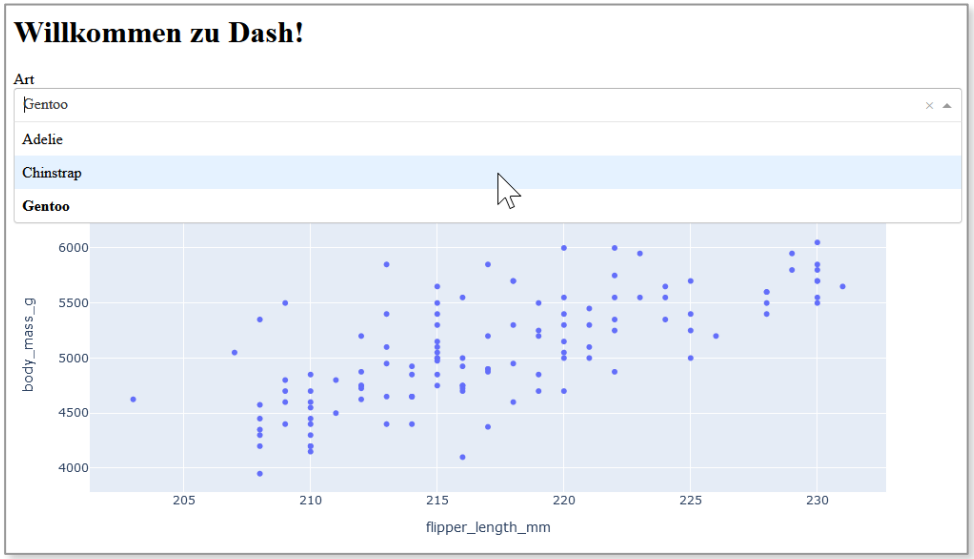
DataCraft

Die Variable app.layout definiert, wie das Dashboard aussieht. Dabei wird das Layout durch HTML-Blöcke zusammengesetzt:

```
from dash import Dash, html, dcc
import seaborn as sns
import plotly.express as px

app = Dash(__name__)
penguins = sns.load_dataset("penguins")
fig = px.scatter(penguins, x="flipper_length_mm", y="body_mass_g")

app.layout = html.Div(
    children=[
        html.H1(children="Willkommen zu Dash!"),
        html.Label("Art"),
        dcc.Dropdown(["Adelie", "Chinstrap", "Gentoo"], id="dd-art),
        html.Br(),
        dcc.Graph(id="Grafik", figure=fig),
    ]
)

app.run_server(debug=True)
```

Interaktivität wird über Dekoratoren definiert, mit denen die Inputs und Outputs definiert werden. Diese Verbindungen werden über die Ids der Layout-Komponenten hergestellt.

```
@app.callback(
                Output("Grafik", "figure"),
                Input("dd-art", "value")
)
def grafik(art):
    return px.scatter(
        penguins[penguins["species"] == art],
                x="flipper_length_mm",
                y="body_mass_g"
    )
```



Es gibt viele verschiedene Komponenten (HTML, DCC, DataTable, Bio, DAQ, Canvas, Slicer, …)

HTML

Fast alle HTML-Tags gibt es in Dash

- html.Div
- html.P
- html.Br
- html.H1
- …

Diese können mit dem Parameter style per CSS angepasst werden

```
html.Div('Example Div',
        style={'color': 'blue',        'fontSize': 14})
```

DCC (Dash Core Components)

Fast alle HTML-Tags gibt es in Dash

- dcc.Graph
- dcc.Dropdown
- dcc.Slider
- dcc.RadioItems
- dcc.ConfirmDialog
- dcc.Tab
- dcc.DatePickerSingle
- …

Plotly Dash – mehrere Seiten

DataCraft

Normalerweise besteht eine Dash-App aus einer Seite. Es können aber mehrere Seiten definiert werden

- Erzeuge für jede Seite eine .py-Datei in einem Unterordner pages
- In jeder der Seiten wird diese per `dash.register_page(__name__)` registriert
- In der Haupt-App `app=Dash(__name__, use_pages=True)` verwenden
- Mit `dash.page_container` werden die Inhalte der gewälten Seite gezeigt