```
import dash
import pandas as pd
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output
import plotly.express as px
import plotly.graph_objects as go

app = dash.Dash(__name__)

data = pd.read_csv('data.csv')

app.layout = html.Div([
    html.H1('Employee Attrition Analysis Dashboard'),

    # Dropdown for selecting a feature to visualize
    html.Label('Select Feature:'),
    dcc.Dropdown(
        id='feature-dropdown',
        options=[{'label': col, 'value': col} for col in data.columns],
        value='Age'
    ),

    # Histogram
    dcc.Graph(id='histogram'),

    # KDE Plot
    dcc.Graph(id='kde-plot'),

    # Bar plot for categorical features
    dcc.Graph(id='bar-plot'),

    # Attrition count plot
    dcc.Graph(
        id='attrition-count',
        figure=px.bar(data, x='Attrition_Yes', title='Attrition Count')
    ),

    # Additional visualizations
    dcc.Graph(id='department-distribution'),
    dcc.Graph(id='monthly-income-distribution'),
    dcc.Graph(id='monthly-income-jobrole-boxplot'),
    dcc.Graph(id='jobrole-distribution'),
    dcc.Graph(id='correlation-matrix'),
    dcc.Graph(id='age-income-scatter'),
    dcc.Graph(id='education-field-distribution')
])

# Callback for updating histogram, KDE plot, and bar plot based on dropdown selection
@app.callback(
    [Output('histogram', 'figure'),
     Output('kde-plot', 'figure'),
     Output('bar-plot', 'figure'),
     Output('department-distribution', 'figure'),
     Output('monthly-income-distribution', 'figure'),
     Output('monthly-income-jobrole-boxplot', 'figure'),
     Output('jobrole-distribution', 'figure'),
     Output('correlation-matrix', 'figure'),
     Output('age-income-scatter', 'figure'),
     Output('education-field-distribution', 'figure')],
    [Input('feature-dropdown', 'value')]
)
def update_graphs(selected_feature):
    if selected_feature in data.select_dtypes(include=['number']).columns:
        histogram = px.histogram(data, x=selected_feature, nbins=30, title=f'{selected_feature} Distribution')
        kde_plot = px.density_contour(data, x=selected_feature, title=f'{selected_feature} KDE')
        bar_plot = {}
    else:
        histogram = {}
        kde_plot = {}
        bar_plot = px.bar(data, x=selected_feature, title=f'{selected_feature} Distribution')

    # Department Distribution
    fig2 = px.histogram(data, x='Department', title='Department Distribution')

    # Monthly Income Distribution
    fig3 = px.histogram(data, x='MonthlyIncome', nbins=30, title='Monthly Income Distribution', marginal='box')

    # Monthly Income by Job Role
    fig4 = px.box(data, x='JobRole', y='MonthlyIncome', title='Monthly Income by Job Role')
    fig4.update_layout(xaxis={'categoryorder':'total descending'})
    # Job Role Distribution
    fig5 = px.histogram(data, y='JobRole', title='Job Role Distribution')

    # Correlation Matrix
    corr = data.corr()
    fig6 = go.Figure(data=go.Heatmap(
        z=corr.values,
        x=corr.columns,
        y=corr.index,
        colorscale='Viridis'
    ))
    fig6.update_layout(title='Correlation Matrix')

    # Age vs. Monthly Income Scatter Plot
    fig7 = px.scatter(data, x='Age', y='MonthlyIncome', color='Attrition_Yes', title='Age vs. Monthly Income by Attrition')

    # Education Field Distribution
    fig8 = px.histogram(data, y='EducationField', title='Education Field Distribution')

    return histogram, kde_plot, bar_plot, fig2, fig3, fig4, fig5, fig6, fig7, fig8

if __name__ == '__main__':
    app.run_server(debug=True)


"""
With this Dashboard app, we can see the visualization of all the data.
We just need to run this python script by the given below code
--- python3 Dashboard_for_Attrition.py ---
From here, we can find so many charts mainly with bar charts with dropdown
menu.
And this file should be in the same folder where the "data.csv" is located.
"""
```