

柱形图

```
In [1]:
from pyecharts.charts import Bar
from pyecharts import options as opts
import pandas as pd
data = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均可支配收入.xls')
# 城镇居民人均可支配收入(元)
data1 = [int(i) for i in list(data.loc[0][1:][::-1])]
year = [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='600px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white'
        )
        .add_xaxis(year)
        .add_yaxis('城镇居民',
                   data1,
                   # 设置柱形条颜色
                   itemstyle_opts=opts.ItemStyleOpts(color='#58B768'))
        .set_global_opts(
            title_opts=opts.TitleOpts(title='2014-2023 年城镇居民人均可支配收入',
                                       subtitle='单位：元',
                                       pos_left='left',
                                       pos_top='top',
                                       # 主标题样式

            title_textstyle_opts=opts.TextStyleOpts(
                color='red',
                font_size=15),
            # 副标题样式

            subtitle_textstyle_opts=opts.TextStyleOpts(
                color='red',
                font_size=15)

        ),
        legend_opts=opts.LegendOpts(# 图例是否显示
                                     is_show=True,
                                     # 图例位置
```

```

pos_left=' 70%',
pos_bottom=' 93%',
# 图例形状
legend_icon=' rect',
# 图例布局
orient=' vertical',
# 图例文本样式

textstyle_opts=opts.TextStyleOpts(

    color=' red',
    font_size=15)

))bar.render_notebook()

```

Out[1]:

堆叠柱形图

In [2]:

```

from pyecharts import options as optsfrom pyecharts.charts import
Barimport pandas as pd data1 =
pd.read_excel(' E:\Python\Anaconda\data\PeopleLife\城镇居民人均可支配收入情况.xls')# 城镇居民人均可支配收入(元)data1 = [int(i) for i in
list(data1.loc[0][1:][::-1])]data2 =
pd.read_excel(' E:\Python\Anaconda\data\PeopleLife\农村居民人均可支配收入情况.xls')# 农村居民人均可支配收入(元)data2 = [int(i) for i in
list(data2.loc[0][1:][::-1])]year = [2014, 2015, 2016, 2017, 2018,
2019, 2020, 2021, 2022, 2023]bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width=' 700px',
            # 设置画布高度
            height=' 400px',
            # 设置主题
            theme=' white')
    )
    .add_xaxis(year)
    .add_yaxis(' 城镇居民',
        data1,
        stack=' stack1',
        color=' lightskyblue'
    )
    .add_yaxis(' 农村居民',
        data2,

```

[illegible]

Out[2]:

条形图

In [3]:

```
from pyecharts.charts import Bar
from pyecharts import options as opts
import pandas as pd
data1 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均收入情况.xls')
# 城镇居民人均可支配收入(元)
data1 = [int(i) for i in list(data1.loc[0][1:])[::-1]]
data2 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均收入情况.xls')
# 农村居民人均可支配收入(元)
data2 = [int(i) for i in list(data2.loc[0][1:])[::-1]]
year = [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='600px',
            # 设置画布高度
            height='500px',
            # 设置主题
            theme='white'
        )
        .add_xaxis(year)
        .add_yaxis('城镇居民', data1,
            # 设置颜色
            itemstyle_opts=opts.ItemStyleOpts(color='#58B768'))
        .add_yaxis('农村居民', data2,
            # 设置颜色
            itemstyle_opts=opts.ItemStyleOpts(color='#E31424'))
        # 翻转XY轴数据
        .reversal_axis()
        .set_global_opts(
            title_opts=opts.TitleOpts(title='2014-2023年城镇和农村居民人均可支配收入',
                subtitle='单位：元',
                pos_left='left',
                pos_top='top',
                # 主标题样式

            title_textstyle_opts=opts.TextStyleOpts(
                color='red',
                font_size=15),
            # 副标题样式
```

```

subtitle_textstyle_opts=opts.TextStyleOpts(
    color='red',
    font_size=15)
),
legend_opts=opts.LegendOpts(# 图例是否显示
    is_show=True,
    # 图例位置
    pos_left='60%',
    pos_bottom='95%',
    # 图例形状
    legend_icon='rect',
    # 图例布局
    orient='horizontal',
    # 图例间隔
    item_gap=10,
    # 图例文本样式

textstyle_opts=opts.TextStyleOpts(
    color='red',
    font_size=15)
)
).set_series_opts(
    # 设置标签的显示位置
    label_opts=opts.LabelOpts(position='right')
))bar.render_notebook()
Out[3]:
In [ ]:
# 绘制柱形图的基本程序结构bar = (Bar(init_opts=opts.InitOpts())      #
设置初始化配置项
    .add_xaxis()                # 设置 x 轴数据
    .add_yaxis()                # 设置 y 轴数据
    .reversal_axis()            # 翻转 xy 轴数据
    .set_global_opts()          # 设置全局配置项
    .set_series_opts()          # 设置系列配置
项)bar.render_notebook()

```

折线图

In [1]:

```
from pyecharts.charts import Line
from pyecharts import options as opts
import pandas as pd
data1 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均可支配收入.xls') # 城镇居民人均可支配收入(元)
data1 = [int(i) for i in list(data1.loc[0][1:])[::-1]]
data2 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均可支配收入.xls') # 农村居民人均可支配收入(元)
data2 = [int(i) for i in list(data2.loc[0][1:])[::-1]]
data3 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\全国居民人均可支配收入.xls') # 全国居民人均可支配收入(元)
data3 = [int(i) for i in list(data3.loc[0][1:])[::-1]]
year = [str(i) for i in [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]]
line = (
    Line(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='600px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white'
        )
    ).add_xaxis(year)
    .add_yaxis('城镇居民', data1,
        # 设置线条颜色
        itemstyle_op
```

散点图

```
In [1]:
from pyecharts.charts import Scatter from pyecharts import options as
optsimport pandas as pd data1 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均收入情
况.xls')# 城镇居民人均可支配收入 data1 = [str(i) for i in
list(data1.loc[0][1:])[::-1]]data2 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均支出情
况.xls')# 城镇居民人均服务性消费支出 data2 = [i for i in
list(data2.loc[2][1:])[::-1]]scatter = (
    Scatter(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='800px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white')
    )
    .add_xaxis(data1)
    .add_yaxis('城镇居民', data2,
        # 设置标记颜色
        itemstyle_opts=opts.ItemStyleOpts(color='blue'),
        # 设置标记图形为圆形
        symbol='circle',
        # 设置标记的大小
        symbol_size=8
    )
    .set_global_opts(
        xaxis_opts=opts.AxisOpts(name='城镇居民人均可支配收入',
            name_textstyle_opts=opts.LabelOpts(font_size=14),
            name_location='middle',
            name_gap=25,
            axislabel_opts=opts.LabelOpts(font_size=12)),
        yaxis_opts=opts.AxisOpts(name='城镇居民人均服务性消费支出',
            name_textstyle_opts=opts.LabelOpts(font_size=14),
            name_location='middle',
            name_gap=50,
            axislabel_opts=opts.LabelOpts(font_s
```

```

In [1]:
from pyecharts.charts import Boxplot
from pyecharts import options as opts
import pandas as pd
data1 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均可支配收入情况.xls')
# 城镇居民人均可支配收入(元)
data1 = [int(i) for i in list(data1.loc[0][1:])[::-1]]
data2 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均可支配收入情况.xls')
# 农村居民人均可支配收入(元)
data2 = [int(i) for i in list(data2.loc[0][1:])[::-1]]
data3 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\全国居民人均可支配收入情况.xls')
# 全国居民人均可支配收入(元)
data3 = [int(i) for i in list(data3.loc[0][1:])[::-1]]
data4 = [data1, data2, data3]
boxplot = (

```

```

In [1]:
from pyecharts import options as opts
from pyecharts.charts import import
TreeMap
import pandas as pd
data1 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均支出情
况.xls')# 2023 年城镇居民各项指标
data1 = data1['2023 年']
data1_total
= float(data1[4] + data1[6] + data1[8] + data1[12] + data1[14] +
data1[16])
data2 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\
农村居民人均支出情况.xls')# 2023 年农村居民各项指标
data2 =
data2['2023 年']
data2_total = float(data2[4] + data2[6] + data2[8] +
data2[12] + data2[14] + data2[16])
data = [
    {
        'value': data1,
        'name': '城镇居民',
        'children': [
            {'va

```

```

In [1]:
from pyecharts.charts import Funnel
from pyecharts import options as opts
import pandas as pd
data = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\全国居民按收入五等份分组的收入情况.xls')
# 2023 年全国居民按五等份分类的类别
data1 = [i[:-15] for i in list(data['指标'])]
# 2023 年全国居民按五等份分类的收入
data2 = list(data['2023 年'])
# 漏斗图用的数据格式是 [(key1, value1)], 使用 zip 函数数据打包成元组
data_pair = [list(z) for z in zip(data1, data2)]
funnel = (
    Funnel(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='600px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme

```

```

In [1]:
from pyecharts import options as opts
from pyecharts.charts import Bar,
Line
import pandas as pd
data =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均收入情
况.xls')# 农村居民人均可支配收入(元)
data1 = [int(i) for i in
list(data.loc[0][1:][::-1])]# 农村居民人均可支配收入比上年增长比
(%)
data2 = [float(i) for i in list(data.loc[1][1:][::-1])]
year =
[str(i) for i in [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021,
2022, 2023]]
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='850px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white')
    )
    .add_xaxis(year)
    .add_yaxi

```

```

In [1]:
from pyecharts import options as opts
from pyecharts.charts import Bar
from pyecharts.charts import Grid
from pyecharts.charts import Pie
import pandas as pd
data = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均收入情况.xls')
# 城镇居民收入类别
data1 = [i[9:-3] for i in list(data['指标'])]
data2 = [data1[4], data1[6], data1[8], data1[10]]
# 2023 年城镇居民人均可支配收入
data3 = list(data['2023 年'])
data4 = [data3[4], data3[6], data3[8], data3[10]]
# 饼图用的数据格式是[(key1, value1)], 使用 zip 函数数据打包成元组
data_pair = [list(z) for z in zip(data2, data4)]
bar = (
    Bar()
    .add_xaxis(data2)
    .add_yaxis('2023 年城镇居民收入',
               data4
            )
    .set_global_opts(xaxis_opts=opts.AxisOpts(
        axislabel_opts=opts.LabelOpts(
            rotate=-20,
            font_size=15
        )
    ),
    title_opts=opts.TitleOpts(title='2023 年城镇居民人均收入柱形图',
                               subtitle='单位：元',
                               pos_left='5%',
                               pos_top='top',
                               # 主

```

```

In [1]:
from pyecharts import options as optsfrom pyecharts.charts import
Barfrom pyecharts.charts import Gridfrom pyecharts.charts import
Linefrom pyecharts.charts import Pageimport pandas as pd data1 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均收入情况.xls')# 城镇居民人均可支配收入(元)data1 = [int(i) for i in
list(data1.loc[0][1:])[::-1]]data2 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均收入情况.xls')# 农村居民人均可支配收入(元)data2 = [int(i) for i in
list(data2.loc[0][1:])[::-1]]data3 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均支出情况.xls')# 城镇居民人均消费支出(元)data3 = [int(i) for i in
list(data3.loc[0][1:])[::-1]]data4 =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均支出情况.xls')# 农村居民人均消费支出(元)data4 = [int(i) for i in
list(data4.loc[0][1:])[::-1]]year_1 = [2014, 2015, 2016, 2017, 2018,
2019, 2020, 2021, 2022, 2023]year_2 = [str(i) for i in [2014, 2015,
2016, 2017, 2018, 2019, 2020, 2021, 2022,

```

```

In [1]:
from pyecharts import options as opts
from pyecharts.charts import Bar
from pyecharts.charts import Line
from pyecharts.charts import Tab
import pandas as pd
data1 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民人均收入情况.xls')
# 城镇居民人均可支配收入(元)
data2 = [int(i) for i in list(data1.loc[0][1:])[::-1]]
# 城镇居民人均可支配收入比上年增长(%)
data3 = [float(i) for i in list(data1.loc[1][1:])[::-1]]
data4 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民人均收入情况.xls')
# 农村居民人均可支配收入(元)
data5 = [int(i) for i in list(data4.loc[0][1:])[::-1]]
# 农村居民人均可支配收入比上年增长(%)
data6 = [float(i) for i in list(data4.loc[1][1:])[::-1]]
year_1 = [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]
year_2 = [str(i) for i in [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023]]
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='800px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white')
    )

```

```

In [1]:
from pyecharts import options as opts
from pyecharts.charts import Bar
from pyecharts.charts import Timeline
import pandas as pd
data1 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\城镇居民按收入五等份分组的收入情况.xls')
# 按收入五等份分组
data2 = [i[:-17] for i in list(data1['指标'])]
# 城镇居民按收入五等份分组收入情况
data3 = [
    [int(i) for i in list(data1['2019 年'])],
    [int(i) for i in list(data1['2020 年'])],
    [int(i) for i in list(data1['2021 年'])],
    [int(i) for i in list(data1['2022 年'])],
    [int(i) for i in list(data1['2023 年'])]
]
data4 = pd.read_excel('E:\Python\Anaconda\data\PeopleLife\农村居民按

```


font_size=16),

```

In [1]:
import pyecharts.options as opts from pyecharts.charts import
Radar import pandas as pd data =
pd.read_excel('E:\Python\Anaconda\data\PeopleLife\全国居民主要食品消
费量.xls') data1 = data['2020 年'] v1 = [[data1[2], data1[3], data1[6],
data1[7], data1[8]]] data2 = data['2021 年'] v2 = [[data2[2], data2[3],
data2[6], data2[7], data2[8]]] data3 = data['2022 年'] v3 = [[data3[2],
data3[3], data3[6], data3[7], data3[8]]] radar=(
    Radar(# 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='600px',
            # 设置画布高度
            height='400px',
            # 设置主题
            theme='white'
        ))
    .add_schema(
        schema=[
            opts.RadarIndicatorItem(name='蔬菜及食用菌', max_=120),
            opts.RadarIndicatorItem(name='肉类', max_=40),
            opts.RadarIndicatorItem(name='蛋类', max_=20),
            opts.RadarIndicatorItem(name='奶类', max_=20),
            opts.RadarIndicatorItem(name='干鲜瓜果类', max_=65)
        ],
        splitarea_opt=opts.SplitAreaOpts(
            is_show=True,
            areastyle_op

```

In [1]:

```
from pyecharts.charts import Bar
from pyecharts.faker import Faker
bar = Bar()
bar.add_xaxis(["衬衫", "羊毛衫", "雪纺衫", "裤子", "高跟鞋", "袜子"])
bar.add_yaxis("商家A", [5, 20, 36, 10, 75, 90])
bar.add_yaxis("商家B", [5, 20, 36, 10, 75, 90])
# render 会生成本地 HTML 文件, 默认会在当前目录生成 render.html 文件# 也可以传入路径参数, 如 bar.render("mycharts.html")
bar.render()
```

Out[1]:

```
'E:\\Python\\Anaconda\\数据可视化-2024-4-18\\代码\\技术准备\\render.html'
```

In [2]:

```
from pyecharts import options as opts
# 导入 Bar 类, 绘制柱形图
from pyecharts.charts import Bar
# 导入 faker 模块, 生成随机数据
from pyecharts.faker import Fake
```

初始配置项

In [1]:

```
from pyecharts.charts import Bar
from pyecharts import options as opts
from pyecharts.faker import Faker
# from pyecharts.globals import ThemeType
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='500px',
            # 设置画布高度
            height='300px',
            # 设置主题
            theme='white',
            # 设置背景色
            bg_color='rgba(145, 214, 243, 0.4)'
        )
        .add_xaxis(Faker.clothes)
        .add_yaxis("商家A", Faker.values())
        .add_yaxis("商家B", Faker.values())
    )
    bar.render_notebook()
```

Out[1]:

全局配置项

In [2]:

```
from pyecharts.charts import Bar
from pyecharts import options as opts
from pyecharts.faker import Faker
bar = (
    Bar(
        # 初始化配置项
        init_opts=opts.InitOpts(
            # 设置画布宽度
            width='500px',
            # 设置画布高度
            height='300px',
            # 设置主题
            theme='white'
        )
        .add_xaxis(Faker.clothes)
        .add_yaxis("商家A", Faker.va
```