```
In [241]:
import requests
from bs4 import BeautifulSoup
import matplotlib.pyplot as plt
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import pandas as pd
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait
import time
from pyecharts import options as opts
from pyecharts.charts import Bar
from pyecharts.charts import Scatter
from pyecharts.charts import Page
from pyecharts.charts import WordCloud
import jieba
```

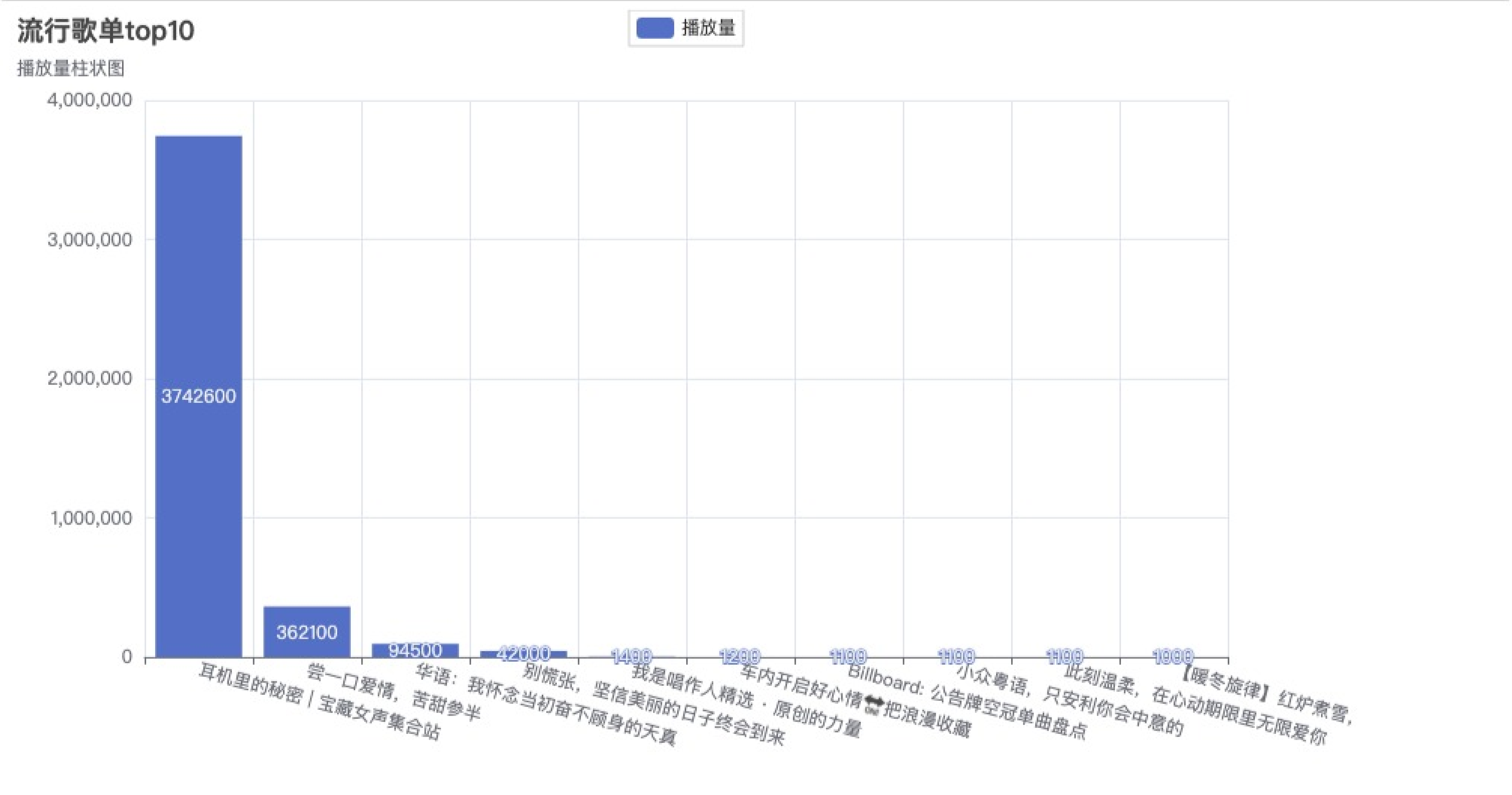
1.1 获取流行歌单播放量top10

```
In [242]:
# 进入分类歌单，获取流派中的流行歌单播放量top10(柱状图)
resp = requests.get("https://y.qq.com/n/ryqq/category?t1=3152")
soup = BeautifulSoup(resp.text, "html")
data_dict = {}
# 解析数据，拿到前10个歌单
for i in soup.find_all(name="li", class_="playlist__item")[:10]:
    key = i.select(".playlist__title_txt")[0].text
    val = float(i.select(".playlist__other")[0].text.replace("播放量：", "").replace("万", ""))*1000
    data_dict[key] = val

# 准备绘图的数据
data_dict = sorted(data_dict.items(),key=lambda d:d[1],reverse=True)
labels = [i[0] for i in data_dict]
data = [i[1] for i in data_dict]
# pyecharts绘制柱状图
c = (
    Bar()
    .add_xaxis(
        labels
    )
    .add_yaxis("播放量", data)
    .set_global_opts(
        xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-15)),
        title_opts=opts.TitleOpts(title="流行歌单top10", subtitle="播放量柱状图"),
    )

)
c.render_notebook()
```

Out[242]:



1.2 获取单个歌单的名称、标签、收藏量、播放量、歌单收录的歌曲数目、评论数

```
In [179]:
from selenium.webdriver.chrome.service import Service

# 以下为selenium配置
chrome_driver = './chromedriver.exe'  # chromedriver的文件位置 https://chromedriver.chromium.org/downloads 此处下载
opts = Options()
opts.add_argument('--headless')
opts.add_argument('--disable-gpu')
opts.add_argument('--no-sandbox')  # root用户不加这条会无法运行
opts.headless = False
prefs = {
    'profile.default_content_setting_values': {
        'images': 2
    }
}
opts.add_experimental_option('prefs', prefs)
s = Service("./chromedriver")

web = webdriver.Chrome(options=opts, service=s)

df = pd.DataFrame(index=None)  # 存放歌单的名称、标签、收藏量、播放量、歌单收录的歌曲数目、评论数
df_relevant = df.copy()  # 作为1.3的分析数据，增加一列歌曲时间
for i in soup.find_all(name="li", class_="playlist__item"):
    web.get(f'https://y.qq.com{i.select(".playlist__cover a")[0]["href"]}')
    web.implicitly_wait(30)
    data = {}
    # 歌单名称
    data["name"] = web.find_element(By.XPATH, '//*[@id="app"]/div/div[2]/div[1]/div/div[1]/h1').text
    # 标签
    data["tag"] = web.find_element(By.XPATH, '//*[@id="app"]/div/div[2]/div[1]/div/ul/li[1]/div/span/a[1]').text
    # 播放量，为字符串"x.x万"，转化为int类型
    data["play"] = float(
        web.find_element(By.XPATH, '//*[@id="app"]/div/div[2]/div[1]/div/ul/li[2]').text.replace("播放量：","").replace(
    try:
        # 收藏量，为字符串"x.x万"，转化为int类型
        time.sleep(1)
        star = web.find_elements(By.CLASS_NAME, 'data_info__item')[2].text.replace("收藏量：", "")
        if "万" in star:
            star = float(star.replace("万", "")) * 10000
        data["star"] = int(star)
    except Exception as e:
        data["star"] = 0
    data["num"] = len(web.find_elements(By.XPATH, '//*[@id="app"]/div/div[2]/div[2]/div[1]/div[1]/ul[2]/li'))
    try:
        comment = int(web.find_element(By.XPATH, '//*[@id="comment_box"]/div[1]/h2/span').text.replace("共", "").replac
        data["comment"] = comment
    except Exception as e:
        data["comment"] = 0
    # 此处d用于数据分析
    d = pd.DataFrame.from_dict([data])
    # 时间 "x:xx"，转化为秒
    times = web.find_element(By.XPATH,
                             '//*[@id="app"]/div/div[2]/div[2]/div[1]/div[1]/ul[2]/li[1]/div/div[4]').text.replace(
        "0", '').split(':')
    time_ = int(times[0]) * 60 + int(times[1])
    data["time"] = time_
    # 增加一列时间
    dr = pd.DataFrame.from_dict([data])
    df = pd.concat([df, d])
    df_relevant = pd.concat([df_relevant, dr])
web.close()
df.replace("", 0, inplace=True)  # 将空值替换为0
print(df)

# 保存到excel
df.to_excel("./data.xlsx", index=False)
```

```
                  name    tag        play   star  num  comment
0      耳机里的秘密 | 宝藏女声集合站  网络歌曲  37425000.0  53000   10      145
0       别慌张，坚信美丽的日子终会到来     国语    420000.0    916   10        5
0            尝一口爱情，苦甜参半     国语   3621000.0   9628   10       17
0       华语：我怀念当初奋不顾身的天真     伤感    945000.0   1601   10        8
0  Billboard: 公告牌空冠单曲盘点     英语     11000.0      7   10        2
0      【暖冬旋律】红炉煮雪，岁寒烤暖     流行     10000.0      0   10        0
0        小众粤语，只安利你会中意的     流行     11000.0      4   10        0
0        车内开启好心情□把浪漫收藏     流行     12000.0      2   10        0
0       我是唱作人精选 · 原创的力量   现场音乐     12000.0      5   10        0
0       此刻温柔，在心动期限里无限爱你     国语     11000.0      1   10        0
0        车载伤感，去听没放完的那首歌     开车     11000.0      0   10        0
0       浪漫物语：心动是四季流转的陪伴     流行     13000.0      2   10        0
0       梗王驾到！标新立异的沙雕进化史   背景音乐     11000.0      1   10        0
0       联系停止了，可思念又怎么停止呢     流行     17000.0     20   10        0
0       2022韩国金唱片获奖作品收录     韩语     88000.0    483   10       43
0       国产宝藏唱作人—暧昧灯光的闪耀     国语     14000.0     15   10        0
0      2022年我最喜欢的华语流行作品     国语     16000.0     14   10        0
0  三石一声 2022 Top25 华语单曲     流行     13000.0     12   10        1
0        抖音热歌：全网最火超好听   背景音乐    188000.0    149   10        0
0        别具一格•当华语歌曲遇上日系     流行     11000.0      7   10        0
```

1.3 分析流行歌单的各歌曲时长的相关性

由于没有歌曲点赞量，此处分析时长、播放量、评论数和收藏量的相关性

```
In [243]:
# 归一化
df_norm=df_relevant.iloc[4:].loc[:, ['play', 'star', 'time', 'comment']].apply(lambda x: round((x - np.min(x)) / (np.ma
# 绘图
c = (
    Scatter()
    .add_xaxis(list(df_relevant.iloc[4:]["name"]))
    .add_yaxis("播放量", list(df_norm.iloc[:]["play"]))
    .add_yaxis("收藏量", list(df_norm.iloc[:]["star"]))
    .add_yaxis("评论量", list(df_norm.iloc[:]["comment"]))
    .add_yaxis("时间", list(df_norm.iloc[:]["time"]))
    .set_global_opts(
        title_opts=opts.TitleOpts(title="相关性散点图"),
        visualmap_opts=opts.VisualMapOpts(type_="size", max_=150, min_=20),
        xaxis_opts=opts.AxisOpts(axislabel_opts=opts.LabelOpts(rotate=-15))
    )

)
c.render_notebook()
```

Out[243]:

2. 分别做三个流派分类下的歌单名的词云图

```
In [244]:
urls = {"流行": "https://y.qq.com/n/ryqq/category?t1=3152",
"r&b": "https://y.qq.com/n/ryqq/category?t1=43",
"轻音乐": "https://y.qq.com/n/ryqq/category?t1=49"}

page = Page()
# 请求三个分类页面
for key in urls.keys():
    resp = requests.get(urls[key])
    soup = BeautifulSoup(resp.text, "html")
    all_text = ""
    word_count = {}
    # 找到页面下所有歌单
    for i in soup.find_all(name="li", class_="playlist__item")[:10]:
        # 拿到歌单名称
        text = i.select(".playlist__title_txt")[0].text
        # jieba库进行分词
        count = jieba.lcut(text)
        for word in count:
            # 处理特殊字符
            if word in ['&']:
                word_count["R&B"] = word_count.get("R&B", 0) + 1
            if word in ["R", '&', "B", ":", ",", "，", "：", "·"]:
                continue
            word_count[word] = word_count.get(word, 0) + 1
    # 排序
    word_items = list(word_count.items())
    word_items.sort(key=lambda x: x[1], reverse=True)
    # 绘图
    c = (
        WordCloud(init_opts=opts.InitOpts(width="800px", height="300px"))
            .add(
            "",
            word_items[:40],
            word_size_range=[20, 100],
            textstyle_opts=opts.TextStyleOpts(font_family="cursive"),
        )
            .set_global_opts(title_opts=opts.TitleOpts(title=f"{key}歌单词云"))
    )
    page.add(c)
page.render_notebook()
```

Out[244]:

流行歌单词云

r&b歌单词云

轻音乐歌单词云

In [ ]: