File S1—Appendix: Core part of the program code.

```
success: (res) => {
                    var hour_week0 = res.data.data.week0;
                    that.week0Data[0] = { categories: text_x, series: [{ name: "Monday",
data: hour_week0[0].filter((v, k) => k % 2 === 0) }] };
                    that.week0Data[1] = { categories: text_x, series: [{ name: "Tuesday",
data: hour_week0[1].filter((v, k) => k % 2 === 0) }] };
                    that.week0Data[2] = { categories: text_x, series: [{ name:
"Wednesday", data: hour_week0[2].filter((v, k) => k % 2 === 0) }] };
                   that.week0Data[3] = { categories: text_x, series: [{ name: "Thursday",
data: hour_week0[3].filter((v, k) => k % 2 === 0) }] };
                    that.week0Data[4] = { categories: text_x, series: [{ name: "Friday",
data: hour_week0[4].filter((v, k) => k % 2 === 0) }] };
                    that.week0Data[5] = { categories: text_x, series: [{ name: "Saturday",
data: hour_week0[5].filter((v, k) => k % 2 === 0) }] };
                    that.week0Data[6] = { categories: text_x, series: [{ name: "Sunday",
data: hour_week0[6].filter((v, k) => k % 2 === 0) }] };

                    var hour_week1 = res.data.data.week1;
                    that.week1Data[0] = { categories: text_x, series: [{ name: "Monday",
data: hour_week0[0].filter((v, k) => k % 2 === 0) }, { name: "Next Monday", data:
hour_week1[0].filter((v, k) => k % 2 === 0) }] };
                    that.week1Data[1] = { categories: text_x, series: [{ name: "Tuesday",
data: hour_week0[1].filter((v, k) => k % 2 === 0) }, { name: "Next Tuesday", data:
hour_week1[1].filter((v, k) => k % 2 === 0) }] };
                    that.week1Data[2] = { categories: text_x, series: [{ name:
"Wednesday", data: hour_week0[2].filter((v, k) => k % 2 === 0) }, { name: "Next Wednesday",
data: hour_week1[2].filter((v, k) => k % 2 === 0) }] };
                   that.week1Data[3] = { categories: text_x, series: [{ name: "Thursday",
```

```
data: hour_week0[3].filter((v, k) => k % 2 === 0) }, { name: "Next Thursday", data: hour_week1[3].filter((v, k) => k % 2 === 0) }] };
                    that.week1Data[4] = { categories: text_x, series: [{ name: "Friday", data: hour_week0[4].filter((v, k) => k % 2 === 0) }, { name: "Next Friday", data: hour_week1[4].filter((v, k) => k % 2 === 0) }] };
                    that.week1Data[5] = { categories: text_x, series: [{ name: "Saturday", data: hour_week0[5].filter((v, k) => k % 2 === 0) }, { name: "Next Saturday", data: hour_week1[5].filter((v, k) => k % 2 === 0) }] };
                    that.week1Data[6] = { categories: text_x, series: [{ name: "Sunday", data: hour_week0[6].filter((v, k) => k % 2 === 0) }, { name: "Next Sunday", data: hour_week1[6].filter((v, k) => k % 2 === 0) }] };

                    // FUCK ucharts
                    setTimeout(() => {
                        console.log(that.week0Data)
                        console.log(that.week1Data)

                        that.week3 = { categories: text_x, series: [{ name: "Tuesday", data: hour_week0[1] }] };
                        console.log(that.week3)


                        var w = new Date().getDay() - 1;
                        if (w == -1) {
                            w = 6;
                        }
                        var h = parseInt(new Date().getHours());
                        console.log(hour_week0, w, h, hour_week0[w][h])
                        that.now_price = hour_week0[w][h - h % 2];
                        that.pre_price = hour_week0[w][(h == 0 ? 24: h) - 2];
```

```
var today_data = that.week0Data[w].series[0].data;
var min_hour = 0;
var min_price = 0;
var max_hour = 0;
var max_price = 0;
var sum_num = 0;
var sum_all = 0;
var sum_price = 0;
console.log(that.week0Data[w].series[0]);
today_data.forEach((v, k) => {
    if (min_price == 0) {
        min_hour = k * 2;
        min_price = v;
    }
    if (max_price == 0) {
        max_hour = k * 2;
        max_price = v;
    }
    if (v < min_price) {
        min_hour = k * 2;
        min_price = v;
    }
    if (v > max_price) {
        max_hour = k * 2;
        max_price = v;
    }
    sum_num += 1;
    sum_all += parseFloat(v);
    console.log(k, v, min_hour, min_price, max_hour,
```

```
max_price, sum_num, sum_all)
                                        });
                                        sum_price = (sum_all /sum_num).toFixed(2);
                                        that.today_min_price = min_price;
                                        that.today_max_price = max_price;
                                        that.today_min_hour = min_hour;
                                        that.today_sum_price = sum_price;
                                    }, 1000)


    <qiun-data-charts class="a1" type="line" :opts="opts" :chartData="v1" />
```