```
    }
    comp_dict = {
        'order_no': ['cnt'],
        'application_amount': ['sum', 'mean', 'max', 'min']
    }
    cols_dtypes_dict = {'has_overdue': int, 'application_term': float,
          'application_amount': float}

    # 根据业务逻辑生成客户历史订单特征
    features_auto = gen_order_feature_auto(raw_orders, 'create_time', '2020-
                    12-14', cols_dtypes_dict, type_cols, comp_cols, time_cut)
    print("特征维度: ", len(features_auto.keys()))
print(features_auto)
```

在上述示例代码中，函数 gen_order_feature_auto() 为特征批量生成的函数，经过特征切分组合之后，返回 80 维特征；函数 rfm_cut() 为数据切分计算特征的核心代码。生成的特征形式如图 3-13 所示。

```
{'f_30_has_overdue_0_order_no_cnt': 11,
 'f_30_has_overdue_1_order_no_cnt': 1,
 'f_30_is_weekend_0_order_no_cnt': 11,
 'f_30_is_weekend_1_order_no_cnt': 1,
 'f_30_has_overdue_0_application_amount_sum': 5170000.0,
 'f_30_has_overdue_0_application_amount_mean': 470000.0,
 'f_30_has_overdue_0_application_amount_max': 850000.0,
 'f_30_has_overdue_0_application_amount_min': 160000.0,
 'f_30_has_overdue_1_application_amount_sum': 850000.0,
 'f_30_has_overdue_1_application_amount_mean': 850000.0,
 'f_30_has_overdue_1_application_amount_max': 850000.0,
 'f_30_has_overdue_1_application_amount_min': 850000.0,
 'f_30_is_weekend_0_application_amount_sum': 5820000.0,
 'f_30_is_weekend_0_application_amount_mean': 529090.9090909091,
 'f_30_is_weekend_0_application_amount_max': 850000.0,
 'f_30_is_weekend_0_application_amount_min': 160000.0,
 'f_30_is_weekend_1_application_amount_sum': 200000.0,
 'f_30_is_weekend_1_application_amount_mean': 200000.0,
 'f_30_is_weekend_1_application_amount_max': 200000.0,
 'f_30_is_weekend_1_application_amount_min': 200000.0}
```

图3-13　RFM切分生成的部分特征

这只是基于 RFM 思路生成特征的简单示例。该示例生成了固定时间窗内的统计特征，在实际工程应用中，我们还可以进一步优化。对于任何结构化的原始数据，简单配置维度数据、度量数据和计算指标即可半自动地完成特征挖掘，批量生成上千维特征。我们可以从下列 4 个方面进行优化：提升数据切分的性能；扩展统计指标函数；区分特征在线计算和离线计算（在线上系统中，可以只计算部分特征以提高性能）；增加定制类别及组合的配置，时间间隔类变量、比值类变量和趋势类变量的计算，以及特征字典生成功能等。

除自行开发基于 RFM 思路的工具以外，还可以借助一些成熟的特征计算工具，如 Featuretools（自动化特征工程的开源 Python 框架）、tsfresh（处理时间序列的关系数据库的特征工程工具）。我们以 tsfresh 生成特征为例，tsfresh 可以方便地提取时间序列的基本特征，如峰值数量、平均值和最大值等，也可以进一步提取更复杂的时间序列特征。

接下来，针对客户历史订单数据，给出提取时间序列特征的示例。每个客户的历史订单申请金额、逾期天数可以看作两个时间序列，需要将文件 order_data.xlsx 中的订单数据做简单的预处理，生成用作输入的订单时间序列。

基于此挖掘时间序列特征的代码如下所示。