Name		Group		Cycle
齐钟昱		**Max_Dev**		**Scrum 02**

Monthly Personal OKR

Objective 目标1：提升编程技能，不欠技术债（**完成**）

KR1 继续看书《Scala编程》，每周5章（不够具体）

KR2 看书Gating压力测试工具，学会看测试报告（没有时限）

KR3 系统学习MongoDB，并搭建分片集群（没有时限）

KR4 撰写读书笔记和思想感悟（完成）

Objective 目标2：替换Max中生成panel的算法，配合MAX上线（**其上有个更大的Objective=>完成Max的商业化上线**）**整体而言，不想是个O，更像是个KRs**

KR1：提高Scala对处理非标准Excel的健壮性（没有时限）

KR2：在算法和数据源正确的情况下，保证生成的panel文件完全正确并且可用（没有时限）

KR3:　完成与Mαx的对接工作,以jar方式引入并根据设定的参数可正确生成panel文件（没有时限）

KR4：根据算法实现panel文件的时间选择功能（没有时限）

KR5：重构panel文件生成代码（没有时限）

KR6：修改panel算法，支持单月份单市场输出（没有时限）

KR7：提升panel算法的写入速度（没有时限）

Daily Brief

20180320 => 星期二					
学习收获	spark的数据倾斜	概念	Spark作业的性能会比期望差很多，即job时间很不稳定。		
		重点1：	数据倾斜的调优，就是使用各种技术方案解决不同类型的数据倾斜问题，以保证Spark作业的性能。数据倾斜是最能体现一个spark大数据工程师水平的性能调优问题。		
		危害	1.数据倾斜直接会导致一种情况：OOM。 2.运行速度慢,特别慢，非常慢，极端的慢，不可接受的慢。		
		起因	某个worker上处理的数据远远大于其他worker上的数据量。好像无聊的二八定律。		
		重点2：	一般情况下，OOM的原因都是数据倾斜。某个task任务数据量太大，GC的压力就很大。这比不了Kafka,因为kafka的内存是不经过JVM的。是基于Linux内核的Page。		

附录：

附1： 算法 => NhwaPanel 使用DataFrame实现

主要分为四部分，处理数据，补充数据，匹配数据，输出数据

1. **处理数据**，将提供的excel文件全部转为csv文件，并进行一些格式的预处理
    a. 对于cpa文件，进行步骤c。
    b. 对于gycx文件，对列进行重命名，将中文的转为英文。进行步骤c。
    c. 对CPA和GYCX进行如下处理。对列进行默认值处理，当列'PRODUCT_NAME'为空时，赋值为列'MOLE_NAME'的值；当列'VALUE'和'STANDARD_UNIT'为空时，赋值为零。新增列min1，为已有的列'PRODUCT_NAME'、'APP2_COD'、'PACK_DES'、'PACK_NUMBER'、'CORP_NAME'依次拼接的结果。新增列”YM”，为已有的列“year”和“month”以此拼接的结果，注意对1-9月需要在前面补0。
    d. 对于cpa的sheet2，1-xx月未到医院名单，转为csv，不需处理。
    e. 对于“2017年未出版医院名单.xlsx”，转为csv，不需处理。
    f. 对于补充医院，转为csv，新增列”YM”，为已有的列“year”和“month”以此拼接的结果，注意对1-9月需要在前面补0。
    g. 对于匹配表，仅保留其中'min1','min1_标准','药品名称'三列，并改“医药名称”列名为“通用名”。
    h. 对于universe表，仅保留其中列'If Panel_All'的值等于1的记录。并修改列名"样本医院编码" -> "ID","PHA医院名称" -> "HOSP_NAME","PHA ID" -> "HOSP_ID","市场" -> "DOI"。新建列’DOIE’使其值等于列’DOI’。只保留"ID" ，"HOSP_NAME"， "HOSP_ID"， "DOI" ，"DOIE"五列。
    i. 对于通用名市场定义表，读入指定市场的sheet，转为csv，不需处理。
2. **补充数据**，有时用户上传的CPA源数据中可能会缺少一些医院，所以需要先补充完整
    a. 读入“1-xx月未到医院名单”表，筛选出包含用户选择的月份的数据，只保留医院列表列，并改列名为“ID”，存储为nah
    b. 读入2017年未出版的20家医院名单，并改列名为“ID”，存储为nph
    c. 将nah和nph进行union并distinct，存储为nh，此名单为所有需要补充数据的医院名单。
    d. 读入处理后的CPA数据，保留YM等于用户所选年份的数据，存为c00
    e. 让c00和nh进行左连接，c0("HOSPITAL_CODE") === nh("ID")，并过滤掉ID is null的记录，删除“ID”列，存为c01，该表为删除需要补充的医院的CPA源数据。
    f. 加载补充医院名单，月份限定在用户选择的月份，并删掉列“x”，存为fhd0
    g. 让fhd0和nh进行内连接，fhd0("HOSPITAL_CODE") === nh("ID")，删除“ID”列，存为fhd1，该表为需要补充的医院数据。
    h. 按列名对fhd1和c01进行union，得到的数据就是补充后的CPA全部数据
3. **匹配数据**，实际匹配，生成panel的过程
    a. 此时已有上面补充后，返回的CPA数据集，存为c0。
    b. 读入处理后的GYCX数据，存为g0。
    c. 读入处理后的匹配表数据，去重，存为m1。m1为产品匹配表。

d. 读入处理后的universe表，保留列“DOI”为当前市场的数据，存为hos0
e. 读入通用名市场定义表，存为b0。b0为市场匹配表。
f. 将表b0与前面读入的表m1做了两次inner join, 分别by 'CPA反馈通用名' 与 'GYCX反馈通用名'，（在表m1中by的对象均为列'通用名'），获得此市场下的2个最小产品单位列表（m1_c与m1_g）。
g. 两个数据源的表分别与前面步骤中生成的最小产品单位列表进行inner join，by列'min1'，再限定其医院编码在hos0中。

4. **输出数据**，去掉多余列，对生成的panel格式化，补充数据，并将CPA和GYCX的结果合并
   a. 剩下的panel只保留如下11列：

原列名	新列名	备注
HOSPITAL_CODE	ID	
	Hosp_name	
YM	Date	
min1_标准	Prod_Name	
	Prod_CNAME	实质是复制Prod_Name
	HOSP_ID	
	Strength	实质是复制Prod_Name
	DOI	
	DOIE	

   b. 将得到的CPA和GYCX的结果合并
   c. 将列'ID'转为数值型。
   d. 剔除列' Sales '为空值的记录。
   e. 其他空值均填以空字符串“”
   f. by前9列计算Units& Sales的和。

附2： 算法 => AstellasPanel 使用RDD实现

主要分为三部分，清洗（处理）数据，匹配数据，输出数据

1. **清洗数据**，因为客户提供的数据可能有细微错误或者格式问题，需处理
    a. 读入CPA源数据
    b. 如果CPA源数据中“商品名备注”不为空，将”商品名”改为不为空的“商品名备注”
    c. 医院编码为“230231”的数据改为“230233”
    d. 医院编码为“110561”的数据改为“110563”
    e. CPA“数据来源”填入“CPA”
    f. 读入GYCX源数据
    g. GYCX中省份为“新疆维吾尔自治区”改为“新疆维吾尔族自治区”
    h. GYCX“数据来源”填入“GYC”
    i. 两个源数据中“商品名”为空的数据填入“药品名称”
    j. 两个源数据中“Value”为0的“Unit”改为0
    k. 两个源数据中“Unit”为0的“Value”改为0
    l. 两个源数据根据“商品名”+“剂型”+“药品规格”+“包装数量”+“生产厂商”依次拼接生成min1
    m. 读入产品匹配表product_match
    n. 对于“包装数量2”为空的数据填入“包装数量1”的值
    o. “标准药品名称”为“抗人胸腺细胞兔免疫球蛋白”的改为“抗人胸腺细胞免疫球蛋白”
    p. “标准商品名”为“米芙”的数据，“标准药品名称”改为“麦考芬酸钠”
    q. “标准商品名”为“哈乐”，“标准剂型”为“片剂”，“包装数量2”为14的，“标准商品名”改为“新哈乐”
    r. “标准商品名”为“新哈乐”，“标准剂型”为“片剂”，“包装数量2”为10的，“标准商品名”改为“哈乐”
2. **匹配数据**，包含一部分清洗
    2.1 找出重复医院：
    a. 读入“医院名称编码等级三源互匹20180314.xlsx”表
    b. 过滤掉“CPA重复码”为空，但“CPA编码”不为空的数据存为hosp_c
    c. 过滤掉“GYC重复码”为空，但“GYC编码”不为空的数据存为hosp_g
    d. “CPA源数据”和“hosp_c”进行left join，加入“判重标识” = “标准编码” + “_” + “年月”,并只保留“判重标识”一列存为hc
    e. “GYCX源数据”和“hosp_g”进行left join，加入“判重标识” = “标准编码” + “_” + “年月”,并只保留“判重标识”一列存为hg
    f. hc 和 hg 取交集，存所有的“判重标识”为表double_hosp_code
    2.2 匹配市场，清洗一些市场数据
    g. GYCX源数据按照double_hosp_code中的“判重标识”，过滤数据
    h. 然后根据“药品名称”进行匹配，为CYCX匹配市场
    i. 两个源数据和产品匹配表根据“min1”进行匹配，暂时定义匹配后的表为total
    j. total表中“标准药品名称” == “他克莫司” && “标准剂型” == “软膏剂”，竞争市场改为“普特彼市场”
    k. total表中“标准药品名称” == “他克莫司” && “标准剂型” != “软膏剂”，竞争市场改为“普乐可复市场”
    l. 删除“佩尔市场”中不是“粉针剂”和”注射剂”的数据
    m. 删除“阿洛刻市场”中是“粉针剂”, “注射剂”, “滴眼剂”的数据
    n. 删除“米开民市场”中是“颗粒剂”, “胶囊剂”, “滴眼剂”, “口服溶剂”, “片剂”的数据
    o. 删除“普乐可复市场”中是“滴眼剂”的数据
    p. 删除“标准商品名” == “保法止”的数据

q. 删除“min2” == "先立晓|片剂|1MG|10|浙江仙琚制药股份有限公司" 的数据
r. 删除“标准药品名称” == "倍他司汀" 的数据
s. 删除“标准药品名称” == "阿魏酰γ-丁二胺/植物生长素" 的数据
t. 删除“标准药品名称” == "丙磺舒" 的数据
u. 删除“标准药品名称” == "复方别嘌醇" 的数据

3. 输出数据

a. 根据（医院编码，日期，min2，市场）四列groupby，对同组的Value和Unit求和，只保留HOSPITAL_CODE -> (YM, min2, mkt, values, units)
b. 此时两表结构一样，正式将两个表union，存为panel0
c. 取出Universe表中“Panel_ID”不为空的数据，只保留“Panel_ID”和”PHA_ID”两列，存为出universe1
d. panle0和universe1进行left_join，根据('ID' = 'Panel.ID')，过滤掉PHA_ID为空的数据，存为panel

附3： 算法 => MAX DataFrame最新算法

单纯算法实现，未进行内存优化，时间复杂度优化的算法

1. 读入panel数据到DF中，命名为panelDF
2. 读入universe数据到DF中，命名为universeDF
3. 对panelDF，根据YM+min1+Hosp_ID分组，对Units和Sales求和，用于下面不中的预测数据，命名为panelSummed
4. 对panelDF只选择YM+min1，去重，并和universeDF进行笛卡尔积，存为joinDF
5. 将panelSummed中求和的值填入joinDF，即进行left join，joinDF根据Hosp_ID,YM,min1. panelSummed根据Hosp_ID,YM,min1, 未匹配到的数据设为0.0，生产有值的计算数据，命名为calcDF
6. 过滤calcDF，只保留（IF_panel_to_use = 1）的数据，根据Segment+min1+YM进行分组，对sumSales，sumUnits，和西药收入求和，分别存为s_sumSales，s_sumUnits，s_westMedicineIncome。
7. 然s_sumSales / s_westMedicineIncome 得 avgSales, ，s_sumUnits / s_westMedicineIncome 得 avg_Units， 结果集命名为segmentDF。
8. joinDF 和 segmentDF 进行join，根据Segment+min1+YM, 新增列f_salse如果（IF_panel_All = 1），f_salse = sumSalse, 如果avgSales或avg_Units小于0，f_salse = 0， 否则f_salse = $"Factor" * $"avgSales" * $"，s_westMedicineIncome"
9. 新增列f_salse如果（IF_panel_All = 1），f_units = sumUnits, 如果avgSales或avg_Units小于0，f_units = 0， 否则f_units = $"Factor" * $"avgUnits" * $"，s_westMedicineIncome"
10. 删除所有f_salse 和 f_units 都为 0 的数据，该结果集即为Max结果