Geekbrains

**Анализ данных и внедрение автоматизированного аналитического инструмента для повышения продаж артикулов с территорий в продуктовом оптовом секторе**

Программа: Разработчик - Аналитика

Специализация: Data Engineer

Сидоренко Сергей Алексеевич

Москва

2024

**Содержание**

Введение

Глава 1. Подготовка данных и инструментов разработки

1.1 Предпосылки создания инструмента, гипотезы

1.2 Техническое задание

1.3 Объяснения причин выбора, используемого стэка на основе имеющегося датасета

1.4 Анализ имеющегося датасета

1.5 Создание итоговой таблицы для будущей автоматизации

1.6 Настройки макроса и внешний вид диалогового окна

1.7 Описание необходимых модулей для разработки

Глава 2. Разработка инструмента при помощи VBA

2.1 Функция CheckInputFiles()

2.2 Модуль ClearAllData()

2.3 Модуль UpdateInputData()

2.4 Модуль FindUniqueInData()

2.5 Модуль UpdateBranchFilters()

2.6 Модуль FirstStepWithData()

2.7 Модуль MakeTable()

2.8 Модуль SecondStepWithData()

2.9 Тестирование технической части

2.10 Улучшение быстродействия

Глава 3. Взаимодействие с рабочей группой для выпуска инструмента

3.1 Определение KPI2 и сроков тестирования

3.2 Инструкция по обучению

3.3 Подведение итогов

Заключение

Список используемой литературы

Глоссарий

Приложения

**Введение**

**Тема проекта:** Анализ данных и внедрение автоматизированного аналитического инструмента для повышения продаж артикулов с территорий в продуктовом оптовом секторе.

**Цель:** Разработать методику, создать и протестировать инструмент для повышения продаж артикулов с территорий в оптовом секторе, описать обоснование на внедрение в случае успешного результата

**Какую проблему решает:** В текущей компании отсутствуют инструменты, которые точечно и на артикульном уровне позволяют быстро оценить объем продаж и доходность товаров с территорий, а также определять руководящему составу задачи для менеджеров.

Кроме того, в связи с закрытым контуром безопасности компании вносить изменения в существующие системы не представляется возможным без должного на то основания. Таким образом для согласования будущего внедрения инструмента в рамках всей компании единственно возможный инструмент для разработки – **EXCEL + VBA** на основании экспортируемых датасетов. И в случае успешного результата составление технического задания для внедрения инструмента на используемых компанией программных продуктах.

В дополнение к указанным выше причинам, для использования в рамках текущего проекта датасеты решено было привести к общему виду, все чувствительные данные обезличены и изменены. Все совершенные изменения позволяют оценить работу инструмента в полной мере.

**Задачи:**

1. Изучить возможные варианты выгрузки данных и возможные коннекторы к внутренним системам для автоматизации процесса
2. Собрать рабочую группу для определения технических требований к будущему инструменту и фиксации тестируемых территорий
3. Создать MVP1 для технического тестирования инструмента
4. Ввести корректировки в инструмент на основе обратной связи технического тестирования
5. Выпустить инструмент на рабочую группу территорий, провести обучение сотрудников
6. Зафиксировать две группы территорий для отслеживания результатов использования новой методики и KPI для оценки
7. Подготовить финальное обоснование необходимости внедрения

**Инструменты:** Excel + VBA, Drawio

**Глава 1. Подготовка данных и инструментов разработки**

* 1. Предпосылки создания инструмента, гипотезы

От операционных функций поступил запрос на создание отчетности по продажам в разрезе менеджеров для контроля со стороны руководителей отделов и последующего увеличения продаж с территорий.

Проектная группа провела “мозговой шторм” и по итогам приоритезации самый высокий приоритет получила следующая гипотеза:

**Создание автоматизированного инструмента на артикульном уровне и последующий анализ менеджеров в сравнении с другими территориями для руководителей отделов позволит увеличить продажи на 10%**

Дополнительная гипотеза:

**Создание инструмента, последующий анализ менеджеров по артикулам с 0 продажами, и выдача листов с заданиями на реализацию этих артикулов позволят увеличить общие продажи на 5%**

Ввиду того, что в компании установлены правила, что все внедрения должны быть финансово обоснованы, было получено разрешение на разработку только за пределами внутреннего контура ИТ систем. И в случае успешного результата подача всех необходимых обоснований внедрения.

Предполагается, что инструмент будут использовать руководители отделов на местах для индивидуальной работы с менеджерами, которые находятся у них в подчинении.

* 1. Техническое задание

Схема будущего процесса выгрузки и анализа данных выглядит следующим образом:

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Автоматически созданное описание

Рис. 1. Схема бизнес-процесса

Общие требования:

* Инструмент должен использовать EXCEL, так как он установлен на все рабочие компьютеры, использование других инструментов не предполагается
* Инструмент не должен подразумевать сложных действий с файлами внутри, так как уровень обученности руководителей находится на разном уровне
* Инструмент должен иметь инструкцию по работе с ним
* Инструмент должен позволят выгружать разное количество данных по территориям с разным количеством менеджеров
* Сделать проверку на наличие файлов с данными, предусмотреть сразу возможность проверки на несколько файлов, в случае будущего добавления полей
* Шаблон финальной таблицы, необходимо утвердить с бизнесом

В общем видео инструмент должен отражать продажи в разбивке на уникальные артикулы в разрезе каждого менеджера и филиала и иметь:

* дополнительно поле - Ранг, которое позволяет увидеть какое количество менеджеров продают тот или иной товар на своей территории
* возможность визуально увидеть артикулы с 0 продажами
* фильтрацию по полям
  1. Объяснения причин выбора, используемого стэка на основе имеющегося датасета
* Внутреннее in-house приложение для отслеживания продаж и аналогичных показателей позволяет делать экспорт отчетности в EXCEL формате
* Подключение любых коннекторов запрещено со стороны безопасности и на этапах разработки, тестирования и отладки
* Было принято решение создать автоматизацию на основе макроса при помощи VBA, который будет преобразовывать экспортируемый датасет и приводить к требуемому виду
* Анализ датасет при помощи других инструментов, например Python, в глобальном смысле не применимы, так как инструмент будет использоваться руководителями подразделений, у которых нет иного ПО, кроме Excel по умолчанию. Кроме того, использования Excel позволит избежать дополнительного обучения сотрудников
* Также для данной работы была необходимость обезличить датасет
  1. Анализ имеющегося датасета

Изображение выглядит как текст, число, Шрифт, снимок экрана

Автоматически созданное описание

Рис. 2. Пример экспортируемого датасета

Экспортируемый датасет имеет 12 полей (столбцов). Количество строк различается в зависимости от количества филиалов и территорий для экспорта. Так, например, для 8 филиалов и 38 территорий выгрузка за месяц содержит более 60 тыс. строк.

* Номер территории согласно географическому распределению
* Номер филиала, который отвечает за ту или иную территорию
* ФИО менеджера, который отвечает за определенную территорию
* № артикула в учетной системе компании
* Наименование артикула в учетной системе компании
* Наименование сегмента бизнеса
* Групповое название отрасли
* Групповое название отрасли (повторное поле, так как выгрузка происходит с глубиной до подсегмента, где дублируется это поле)
* Номер подсегмента в учетной системе компании
* Наименование подсегмента в учетной системе компании
* Продажи в рублях без учета НДС в учетной системе компании
* Маржа в рублях в учетной системе компании
  1. Создание итоговой таблицы для будущей автоматизации

На основании имеющегося датасета были проведены встречи с представителями бизнеса, которые в потенциале будут использовать инструмент. На основе их обратной связи получены требования и согласно им вручную была построена таблица, вида:

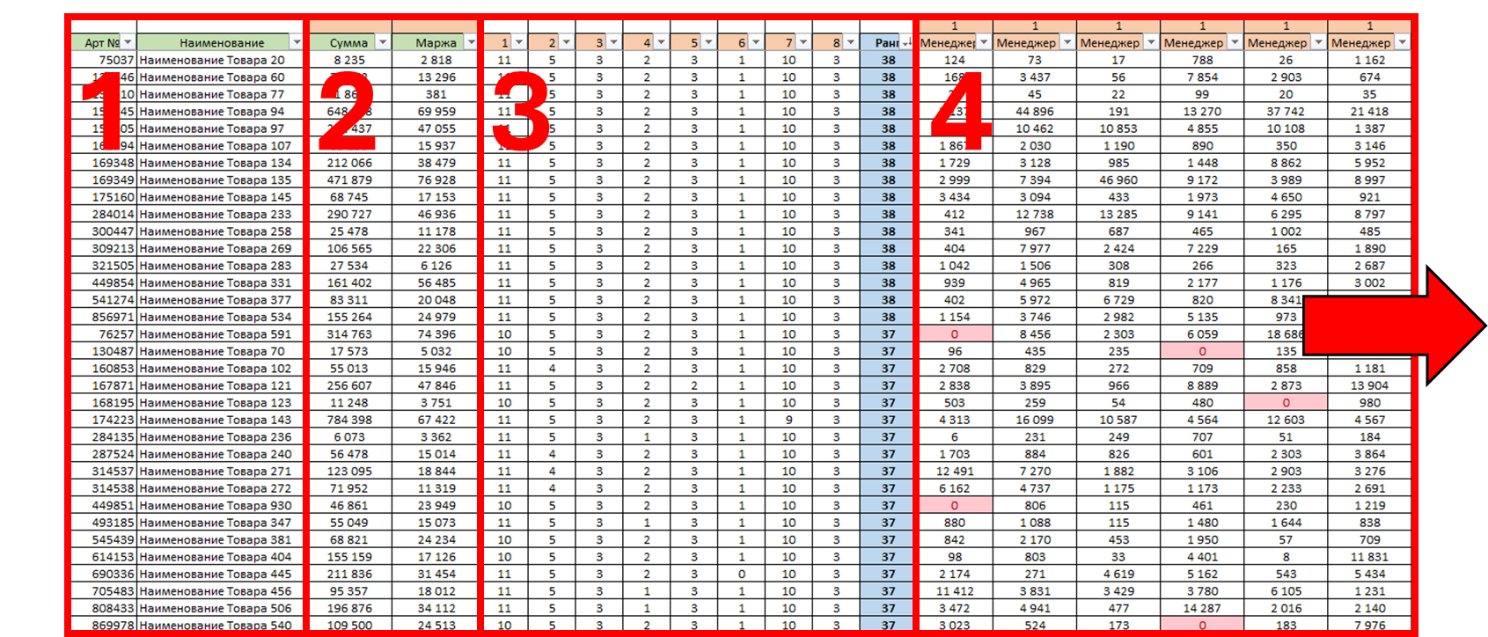


Рис 3. Шаблон итоговой таблицы

Где должны быть 4 блока:

1. Номер артикула и его наименование – не повторяющиеся данные для анализа на артикульном уровне. Основа для сбора всех остальных данных.

2. Итоговые продажи и сумма маржи по каждому артикулу.

3. Количество продающих тот или иной артикул менеджеров для выявления наименований, которые не продаются какой-то территории/филиале. Кроме того, добавление нового поля Ранг, который суммирует предыдущие значения. Фильтрация по убыванию должна производиться именно по этому показателю.

4. Сумма продаж в разрезе каждого филиала и менеджера (территории). Блок должен масштабироваться вправо в зависимости от того сколько будет экспортировано в датасет филиалов и территорий. Необходимо реализовать автоматическую подсветку сумм продаж равных 0, для более удобного визуального поиска.

Дополнительные требования:

* Фильтрацию необходимо по умолчанию установить на 2 строке
* Закрепление произвести по горизонтали и вертикали после столбца «Ранг»
* Таблица должна строиться в момент исполнения программы и одинаково работать как с одним филиалом, так и с экспортом данных по всей компании
  1. Настройки макроса и внешний вид диалогового окна



Рис. 4. Внешний вид листа Settings с настройками и этапами запуска макроса

Настройка макроса должна давать возможность дополнительно настраивать поля, которые будут преобразованы, так как есть гипотеза, что внутренние системы могут по-разному выгружать датасет в редких случаях.

Кроме того, после первого этапа загрузки данных было принято решение сразу дать возможность фильтровать будущую таблицу по Сегментам и Подсегментам.

Окно настройки содержит только две кнопки, ровно на два этапа формирования будущей таблицы.

* 1. Описание необходимых модулей для разработки

Для последующего удобства внесения корректировок принято решение разбить исполнение на два основных блока:

* Этап загрузки данных
* Этап формирования результирующей таблицы

Каждый из этапов декомпозируем до модулей, которые имеют различное предназначение и могут быть заменены или модернизированы в зависимости от обратной связи от бизнеса. Необходимость разбивки на два этапа также обусловлена тем, что после первого этапа, необходимо, чтобы была возможность фильтровать будущие данные пользователю в моменте выгрузки на основе имеющихся в его конкретном датасете Сегментов и Подсегментов.

Далее подробно о каждом модуле.

**Глава 2. Разработка инструмента при помощи VBA**

2.1 Функция CheckInputFiles()

Основная задача функции проверять наличие всех необходимых файлов для загрузки данных в макрос. По ТЗ от бизнеса согласовано добавить несколько вариантов файлов для будущего масштабирования, если потребуются дополнительные файлы со входными данными.

Function CheckInputFiles() As Boolean

'

' Проверка на наличие всех необходимых файлов для работы

'

'----------------------------------------------------------

' По тех. заданию заложить как минимум +1 файл для проверки

' Дублируем первый файл как заглушку для будущего добавления

'----------------------------------------------------------

'

Dim File1Path As String

Dim File2Path As String

File1Path = ThisWorkbook.Path & "/data.xlsx"

File2Path = ThisWorkbook.Path & "/data.xlsx"

If Dir(File1Path, vbDirectory) = vbNullString Or \_

    Dir(File2Path, vbDirectory) = vbNullString Then

    MsgBox "Необходимые файлы для запуска отсуствуют" & vbNewLine & \_

"Проверьте наличие файлов" & vbNewLine & \_

        "data.xlsx" & vbNewLine & \_

        "data.xlsx"

    CheckInputFiles = False

    Exit Function

End If

    CheckInputFiles = True

End Function

2.2 Модуль ClearAllData()

Простой модуль для очистки всех листов с данными для последующей загрузки новых входных параметров

Sub ClearAllData()

'

' Очистка всех листов с данными перед запуском обновления

'

Sheets("data").Select

Cells.Select

Selection.Clear

Cells(1, 1).Select

For i = 1 To 5

    Sheets(CStr(i)).Select

    Cells.Select

    Selection.Clear

    Cells(1, 1).Select

Next

End Sub

2.3 Модуль UpdateInputData()

Ввиду того, что на начальном этапе было достаточно использовать выгрузку одного датасета из внутренних систем компании, для удобства работы и ускорения расчетов загружаем данные непосредственно в файл с макросом. Так как выгружаемый датасет имеет стандартизованный вид, принято решение убирать лишние строки при загрузке. Далее по итогам тестирования в реальных условиях будет производит анализ будущего применения – либо унификация формы с проверкой на соответствие шаблону, либо, в случае встраивания в BI системы компании, создание технического задания на получение необходимых данных.

Sub UpdateInputData()

'

' Загрузка и обновление данных по продажам и марже

'

    ' Загрузка данных из файла data

    ' файл располагается в той же папке, что и сам макрос

    ' --------------------------------------------------------------------------

    'открываем файл

    Workbooks.Open Filename:=ThisWorkbook.Path & "\data.xlsx"

    Cells.Select

    Selection.Copy

    'возвращаемся в исходный

    Windows("ProjectX.xlsm").Activate

    Sheets("data").Select

    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

    'очистка буфера обмена

    Application.CutCopyMode = False

    'особенность входных данных - первые две ячейки пустые

    'проверяем, что все так и есть, и удаляем

    If Cells(1, 2).Value = "" And Cells(2, 2).Value = "" Then

        Rows("1:2").Select

        Selection.Delete Shift:=xlUp

    End If

    'особенность входных данных - вторая ячейка с данными суммирующая

    'проверяем, что все так и есть, и удаляем

    If Cells(2, 1).Value = "Итого" Then

        Rows("2:2").Select

        Selection.Delete Shift:=xlUp

    End If

    Windows("data.xlsx").Activate

    Windows("data.xlsx").Close

    ' --------------------------------------------------------------------------

    Windows("ProjectX.xlsm").Activate

    Sheets("Settings").Select

End Sub

2.4 Модуль FindUniqueInData()

Датасет с данными содержит полные данные про продажам в разрезе всех необходимых параметров и для последующего сбора сводной результирующей таблицы необходимо создать списки уникальных значений по которым будут считаться параметры значений продаж. Создаем полный список уникальных артикулов, филиалов, менеджеров для подсчета продаж. А так список уникальных сегментов и подсегментов для возможной фильтрации и детализации отчетности. Важность модуля в том, что пользователь может изначально выбрать при выгрузке из внутренней системы компании разное количество территорий для выгрузки и все параметры будут браться из текущего датасета, а не из списка всех возможных вариаций по данным компании. Таким образом достигается более высокая универсальность решения.

Sub FindUniqueInData()

'

' Создание списков уникальных значений для формирования результирующей таблицы

'

Dim Col\_number\_of\_stores As Integer

Dim Col\_ts\_fio As Integer

Dim Col\_articles As Integer

Dim Col\_sales As Integer

' Использование глобальных переменных отменено по ТЗ, реализована возможность настройки

' пользователем номеров столбцов во входном файле, так как у части учетных записей

' данные могут выгружаться в другом порядке, загружаем с листа Settings

Col\_number\_of\_stores = Sheets("Settings").Cells(2, 6).Value

Col\_ts\_fio = Sheets("Settings").Cells(3, 6).Value

Col\_articles = Sheets("Settings").Cells(4, 6).Value

Col\_sales = Sheets("Settings").Cells(5, 6).Value

Col\_branch = Sheets("Settings").Cells(6, 6).Value

Col\_sub\_branch = Sheets("Settings").Cells(7, 6).Value

Sheets("data").Select

'находим последнюю непустую строку на листе

LastRow = Cells(Rows.Count, 2).End(xlUp).Row

'----------------------------------------------------------------------------------

'Формируем список филиалов

'выделяем диапазон всех ячеек с дублирующейся информацией

Range(Cells(2, Col\_number\_of\_stores), Cells(LastRow, Col\_number\_of\_stores)).Select

Selection.Copy

Sheets("1").Select

Cells(1, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

LastRow\_temp = Cells(Rows.Count, 1).End(xlUp).Row

'оставляем только уникальные значения

ActiveSheet.Range(Cells(1, 1), Cells(LastRow\_temp, 1)).RemoveDuplicates Columns:=1, Header:=xlNo

Cells(1, 1).Select

'--------------------------------

'Сортировка списка по возрастанию

    Cells.Select

    ActiveWorkbook.Worksheets("1").Sort.SortFields.Clear

    ActiveWorkbook.Worksheets("1").Sort.SortFields.Add2 Key:=Range("A1:A65889"), \_

        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

    With ActiveWorkbook.Worksheets("1").Sort

    'сортируем весь столбец

        .SetRange Range("A1:B65889")

        .Header = xlGuess

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

    Range("A1").Select

'--------------------------------

Sheets("data").Select

'----------------------------------------------------------------------------------

'Формируем список фамилий менеджеров

'выделяем диапазон всех ячеек с дублирующейся информацией

Range(Cells(2, Col\_number\_of\_stores), Cells(LastRow, Col\_ts\_fio)).Select

Selection.Copy

Sheets("2").Select

Cells(1, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

LastRow\_temp = Cells(Rows.Count, 1).End(xlUp).Row

'оставляем только уникальные значения

ActiveSheet.Range(Cells(1, 1), Cells(LastRow\_temp, 2)).RemoveDuplicates Columns:=2, Header:=xlNo

Cells(1, 1).Select

'--------------------------------

'Сортировка списка по возрастанию

    Cells.Select

    ActiveWorkbook.Worksheets("2").Sort.SortFields.Clear

    ActiveWorkbook.Worksheets("2").Sort.SortFields.Add2 Key:=Range("A1:A65889"), \_

        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

    ActiveWorkbook.Worksheets("2").Sort.SortFields.Add2 Key:=Range("B1:B65889"), \_

        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

    With ActiveWorkbook.Worksheets("2").Sort

        .SetRange Range("A1:B65889")

        .Header = xlGuess

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

    Range("A1").Select

'--------------------------------

Sheets("data").Select

'----------------------------------------------------------------------------------

'Формируем список артикулов

'выделяем диапазон всех ячеек с дублирующейся информацией

Range(Cells(2, Col\_articles), Cells(LastRow, Col\_articles + 1)).Select

Selection.Copy

Sheets("3").Select

Cells(1, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

LastRow\_temp = Cells(Rows.Count, 1).End(xlUp).Row

'оставляем только уникальные значения

ActiveSheet.Range(Cells(1, 1), Cells(LastRow\_temp, 2)).RemoveDuplicates Columns:=1, Header:=xlNo

Cells(1, 1).Select

'--------------------------------

'Сортировка списка по возрастанию

    Cells.Select

    ActiveWorkbook.Worksheets("3").Sort.SortFields.Clear

    ActiveWorkbook.Worksheets("3").Sort.SortFields.Add2 Key:=Range("A1:A65889"), \_

        SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

    With ActiveWorkbook.Worksheets("3").Sort

        .SetRange Range("A1:B65889")

        .Header = xlGuess

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

    Range("A1").Select

'--------------------------------

Sheets("data").Select

'----------------------------------------------------------------------------------

'Формируем список сегментов

Range(Cells(2, Col\_branch), Cells(LastRow, Col\_branch)).Select

Selection.Copy

Sheets("4").Select

Cells(1, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

LastRow\_temp = Cells(Rows.Count, 1).End(xlUp).Row

ActiveSheet.Range(Cells(1, 1), Cells(LastRow\_temp, 2)).RemoveDuplicates Columns:=1, Header:=xlNo

Cells(1, 1).Select

Sheets("data").Select

'----------------------------------------------------------------------------------

'Формируем список подсегментов

Range(Cells(2, Col\_sub\_branch), Cells(LastRow, Col\_sub\_branch)).Select

Selection.Copy

Sheets("5").Select

Cells(1, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

LastRow\_temp = Cells(Rows.Count, 1).End(xlUp).Row

ActiveSheet.Range(Cells(1, 1), Cells(LastRow\_temp, 2)).RemoveDuplicates Columns:=1, Header:=xlNo

Cells(1, 1).Select

Sheets("data").Select

'----------------------------------------------------------------------------------

Sheets("Settings").Select

End Sub

2.5 Модуль UpdateBranchFilters()

Модуль обновляет список сегментов и подсегментов исходя из текущего датасета. Данный подход позволяет избежать проблемы, в случае переименования названий сегментов и подсегментов со стороны разных функций, формируется список возможных вариантов в моменте выполнения макроса.

Sub UpdateBranchFilters()

'

' Проверка значений для сегментов и подсегментов

' создание выпадающих списков для фильтрации

'

    'сегменты

    Sheets("4").Select

    LastRow\_brunch = Cells(Rows.Count, 1).End(xlUp).Row

    Sheets("Settings").Select

    Range("J2:J4").Select

    Application.CutCopyMode = False

    With Selection.Validation

        .Delete

        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= \_

        xlBetween, Formula1:="='4'!$A$1:$A$" & LastRow\_brunch

        .IgnoreBlank = True

        .InCellDropdown = True

        .InputTitle = ""

        .ErrorTitle = ""

        .InputMessage = ""

        .ErrorMessage = ""

        .ShowInput = True

        .ShowError = True

    End With

    'подсегменты

    Sheets("5").Select

    LastRow\_subbrunch = Cells(Rows.Count, 1).End(xlUp).Row

    Sheets("Settings").Select

    Range("J5:J7").Select

    Application.CutCopyMode = False

    With Selection.Validation

        .Delete

        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= \_

        xlBetween, Formula1:="='5'!$A$1:$A$" & LastRow\_subbrunch

        .IgnoreBlank = True

        .InCellDropdown = True

        .InputTitle = ""

        .ErrorTitle = ""

        .InputMessage = ""

        .ErrorMessage = ""

        .ShowInput = True

        .ShowError = True

    End With

End Sub

2.6 FirstStepWithData()

Модуль объединяет части программы, которые относятся к шагу подготовки и загрузки новых данных для последующей работы с ними внутри макроса.

Sub FirstStepWithData()

'

' Запуск всех модулей, связаннных с обновлением входных данных

'

    Dim checkFiles As Boolean

    'проверяем наличие файла с исходными данными в папке с макросом

    checkFiles = CheckInputFiles()

    Application.ScreenUpdating = False

    Application.CutCopyMode = False

    Sheets("1").Visible = True

    Sheets("2").Visible = True

    Sheets("3").Visible = True

    Sheets("4").Visible = True

    Sheets("5").Visible = True

    Sheets("data").Visible = True

    Sheets("Tasks").Visible = True

    'обновляем только в случае наличия входных данных

    If checkFiles Then

        Dim t As Single

        t = Timer

        ClearAllData ' Очистка всех листов для загрузки данных

        UpdateInputData ' Обновляем файлы с продажами и маржой

        FindUniqueInData ' Отбор уникальных элементов для итоговой таблицы

        UpdateBranchFilters ' Обновление фильтрации

        Sheets("Settings").Select

        t = Timer - t

        UpdateMsg = MsgBox("Обновление прошло успешно за " & Round(t, 0) & \_

" секунд", vbOKOnly, "Обновление")

    End If

    Sheets("1").Visible = False

    Sheets("2").Visible = False

    Sheets("3").Visible = False

    Sheets("4").Visible = False

    Sheets("5").Visible = False

    Sheets("data").Visible = False

    Sheets("Tasks").Visible = False

    Application.CutCopyMode = False

    Application.ScreenUpdating = True

End Sub

2.7 Модуль MakeTable()

В данном модуле создаем результирующую таблицу для последующей работы пользователей. Ширина будущей таблицы неизвестна заранее по ТЗ и формируется в моменте исходя из поданного в нее датасета, что позволяет гибко создавать инструмент для работы с малым или полным списком территорий в зависимости от уровня и детализации анализа.

Финальный список упорядочивается по полу Ранг, который отображает какое количество менеджеров продает на территории тот или иной товар.

Первая версия модуля содержит блок, который за счет использования цикла на больших выгрузках может считаться довольно продолжительное время (Помечен \*\*\* в коде). По итогам технического тестирования будет принято решение о необходимости ускорения модуля на основании обратной связи от бизнеса. Кроме того, могут потребоваться дополнительные поля для отображения.

Sub Make\_table()

'

' Формирование результирующей таблицы

'

'объявляем счетчик

Dim Counter As Integer

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Очистка листов Work и Tasks перед загрузкой новых данных

Sheets("Work").Select

Cells.Select

Selection.Clear

'убираем закрепление границ

ActiveWindow.FreezePanes = False

Cells.Select

Cells.FormatConditions.Delete

Cells(1, 1).Select

Sheets("Tasks").Select

Columns("C:C").Select

Range(Selection, Selection.End(xlToRight)).Select

Selection.Delete Shift:=xlToLeft

Rows("4:4").Select

Range(Selection, Selection.End(xlDown)).Select

Selection.Delete Shift:=xlUp

Cells.Select

'убираем условное форматирование со всего листа

Cells.FormatConditions.Delete

ActiveWindow.FreezePanes = False

Cells(1, 1).Select

'----------------------------------------------------------------------------------

'Заполнение списка артикулов

Sheets("3").Select

LastRow = Cells(Rows.Count, 1).End(xlUp).Row

Range(Cells(1, 1), Cells(LastRow, 2)).Select

Selection.Copy

Sheets("Tasks").Select

Cells(4, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Заполнение списка всех менеджеров слева направо

Sheets("2").Select

Number\_of\_managers = Cells(Rows.Count, 1).End(xlUp).Row

'используем цикл для заполнения любого возможного количества менеджеров

Counter = 1

Do While Counter <= Number\_of\_managers

    Sheets("2").Select

    Cells(Counter, 2).Select

    Selection.Copy

    Sheets("Tasks").Select

    LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

    Cells(3, LastCol).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Counter = Counter + 1

Loop

'----------------------------------------------------------------------------------

'Копируем номер филиала в привязке к менеджеру

LastCol\_managers = Cells(3, Columns.Count).End(xlToLeft).Column

Cells(2, 2).Select

Selection.Copy

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Cells(2, 2).Select

Selection.Copy

Range(Cells(2, 3), Cells(3, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.Copy

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Заполняем суммы продаж по каждому менеджеру, учитывая фильтрацию по сегментам и подсегментам

'

LastRow = Cells(Rows.Count, 1).End(xlUp).Row

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'TO-DO отдать в тестирование, нормально ли по времени исполнения

Sheets("data").Select

LastRow\_data = Cells(Rows.Count, 1).End(xlUp).Row

Sheets("Tasks").Select

Sum\_column = Sheets("Settings").Cells(5, 6).Value

Article\_column = Sheets("Settings").Cells(4, 6).Value

Manager\_column = Sheets("Settings").Cells(3, 6).Value

Brunch\_column = Sheets("Settings").Cells(6, 6).Value

SubBrunch\_column = Sheets("Settings").Cells(7, 6).Value

Dim rng\_1 As Range

Dim rng\_2 As Range

Dim rng\_3 As Range

Dim rng\_4 As Range

Dim rng\_5 As Range

Sheets("data").Select

'присваиваем необходимые диапазоны для удобства использования в циклах

Set rng\_1 = Sheets("data").Range(Cells(1, Sum\_column), Cells(LastRow\_data, Sum\_column))

Set rng\_2 = Sheets("data").Range(Cells(1, Article\_column), Cells(LastRow\_data, Article\_column))

Set rng\_3 = Sheets("data").Range(Cells(1, Manager\_column), Cells(LastRow\_data, Manager\_column))

Set rng\_4 = Sheets("data").Range(Cells(1, Brunch\_column), Cells(LastRow\_data, Brunch\_column))

Set rng\_5 = Sheets("data").Range(Cells(1, SubBrunch\_column), Cells(LastRow\_data, SubBrunch\_column))

Branch\_Value = Sheets("Settings").Range("J2").Value

SubBranch\_Value = Sheets("Settings").Range("J5").Value

Sheets("Tasks").Select

'---------------------

'Циклы заполнения в зависимости от условий выбора сегмента и подсегмента

Counter\_sum = 1

If Branch\_Value = "" And SubBranch\_Value = "" Then

    For i = 3 To LastCol\_managers

        For j = 4 To LastRow

            Cells(j, i).Value = WorksheetFunction.SumIfs(rng\_1, rng\_2, Cells(j, 1).Value, \_

                rng\_3, Cells(3, i).Value)

        Next

        Counter\_sum = Counter\_sum + 1

    Next

ElseIf Branch\_Value = "" And SubBranch\_Value <> "" Then

    For i = 3 To LastCol\_managers

        For j = 4 To LastRow

            Cells(j, i).Value = WorksheetFunction.SumIfs(rng\_1, rng\_2, Cells(j, 1).Value, \_

                rng\_3, Cells(3, i).Value, \_

                rng\_5, SubBranch\_Value)

        Next

        Counter\_sum = Counter\_sum + 1

    Next

ElseIf SubBranch\_Value = "" And Branch\_Value <> "" Then

    For i = 3 To LastCol\_managers

        For j = 4 To LastRow

            Cells(j, i).Value = WorksheetFunction.SumIfs(rng\_1, rng\_2, Cells(j, 1).Value, \_

                rng\_3, Cells(3, i).Value, \_

                rng\_4, Branch\_Value)

        Next

        Counter\_sum = Counter\_sum + 1

    Next

Else

    For i = 3 To LastCol\_managers

        For j = 4 To LastRow

            Cells(j, i).Value = WorksheetFunction.SumIfs(rng\_1, rng\_2, Cells(j, 1).Value, \_

                rng\_3, Cells(3, i).Value, \_

                rng\_4, Branch\_Value, \_

                rng\_5, SubBranch\_Value)

        Next

        Counter\_sum = Counter\_sum + 1

    Next

End If

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'копируем формулу, вставляем формат ячейки и оставляем только значения без формул

Cells(1, 2).Select

Selection.Copy

Range(Cells(4, 3), Cells(LastRow, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Selection.Copy

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Подсвечиваем 0 условных форматированием на всем диапазоне

Range(Cells(4, 3), Cells(LastRow, LastCol\_managers)).Select

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, \_

    Formula1:="=0"

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

With Selection.FormatConditions(1).Font

    .Color = -16383844

    .TintAndShade = 0

End With

With Selection.FormatConditions(1).Interior

    .PatternColorIndex = xlAutomatic

    .Color = 13551615

    .TintAndShade = 0

End With

Selection.FormatConditions(1).StopIfTrue = False

'----------------------------------------------------------------------------------

'Добавляем счет продаж по каждому филиалу

Sheets("1").Select

Number\_of\_stores = Cells(Rows.Count, 1).End(xlUp).Row

Counter = 1

Do While Counter <= Number\_of\_stores

    Sheets("1").Select

    Cells(Counter, 1).Select

    Selection.Copy

    Sheets("Tasks").Select

    LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

    Cells(3, LastCol).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Counter = Counter + 1

Loop

'----------------------------------------------------------------------------------

'Считаем количество продающих менеджеров

LastCol = Cells(4, Columns.Count).End(xlToLeft).Column

Cells(4, LastCol).Select

'-----------------------------------------------------------------

'Переменная для определения границ сумм столбца Ранг

LastCol\_for\_sum = Cells(4, Columns.Count).End(xlToLeft).Column + 1

'-----------------------------------------------------------------

'реализация через цикл, так как не большой объем вычислений

Counter = 1

For i = 1 To Number\_of\_stores

    For j = 4 To LastRow

        Cells(j, i + LastCol).Value = WorksheetFunction.CountIfs(Range(Cells(j, 3), Cells(j, LastCol)), ">0", Range(Cells(2, 3), Cells(2, LastCol)), Cells(3, LastCol + Counter).Value)

    Next

    Counter = Counter + 1

Next

'----------------------------------------------------------------------------------

'Добавляем столбец Ранг

LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

Cells(3, LastCol).Value = "Ранг"

For i = 4 To LastRow

    Cells(i, LastCol).Value = WorksheetFunction.Sum(Range(Cells(i, LastCol\_for\_sum), Cells(i, LastCol)))

Next

'----------------------------------------------------------------------------------

'Изменяем форматирование вновь созданных столбцов

Cells(2, 2).Select

Selection.Copy

Range(Cells(3, LastCol\_for\_sum), Cells(3, LastCol)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Cells(1, 2).Select

Selection.Copy

Range(Cells(4, LastCol\_for\_sum), Cells(LastRow, LastCol)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Range(Cells(3, LastCol), Cells(LastRow, LastCol)).Select

Selection.Font.Bold = True

With Selection.Interior

    .Pattern = xlSolid

    .PatternColorIndex = xlAutomatic

    .ThemeColor = xlThemeColorAccent5

    .TintAndShade = 0.599993896298105

    .PatternTintAndShade = 0

End With

'----------------------------------------------------------------------------------

'Копирование всей страницы на новый лист для удобства работы

    Range(Cells(2, 1), Cells(LastRow, LastCol)).Select

    Selection.Copy

    Sheets("Work").Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Selection.Borders(xlDiagonalDown).LineStyle = xlNone

    Selection.Borders(xlDiagonalUp).LineStyle = xlNone

    With Selection.Borders(xlEdgeLeft)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeTop)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeBottom)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeRight)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlInsideVertical)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlInsideHorizontal)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    'автоматическая настройка ширины и внешнего вида отображения

    Range(Cells(2, LastCol\_for\_sum), Cells(LastRow, LastCol)).EntireColumn.Select

    Selection.Cut

    Columns("C:C").Select

    Selection.Insert Shift:=xlToRight

    Application.CutCopyMode = False

    Range(Cells(2, 1), Cells(LastRow, LastCol)).EntireColumn.AutoFit

    ActiveWindow.Zoom = 70

    ColFilter = 0

    '----------------------------------------

    'Ищем столбец Ранг для будущей сортировки

    For i = 1 To LastCol

        If Cells(2, i).Value = "Ранг" Then

            ColFilter = i

            Exit For

        End If

    Next

    '----------------------------------------

    'добавляем фильтры и сортируем по столбцу Ранг по убыванию

    Rows("2:2").Select

    Selection.AutoFilter

    ActiveWorkbook.Worksheets("Work").AutoFilter.Sort.SortFields.Clear

    ActiveWorkbook.Worksheets("Work").AutoFilter.Sort.SortFields.Add2 Key:=Range( \_

        Cells(2, ColFilter), Cells(2, ColFilter)), SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:= \_

        xlSortNormal

    With ActiveWorkbook.Worksheets("Work").AutoFilter.Sort

        .Header = xlYes

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

    Cells(3, ColFilter + 1).Select

    ActiveWindow.FreezePanes = True

    '--------------------------------------------------------------------------

    Cells(1, 1).Select

    Selection.Copy

    Cells(1, 2).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

End Sub

2.8 Модуль SecondStepWithData()

Модуль, который служит для объединений второй части модулей по формированию результирующей таблицы. Требуется для возможного добавления новых модулей во второй этап работы макроса.

Sub SecondStepWithData()

'

' Модуль запуска для старта формирования результирующей таблицы и дополнительных расчетов в потенциале

'

'выключаем обновление экрана

Application.ScreenUpdating = False

Application.CutCopyMode = False

Sheets("1").Visible = True

Sheets("2").Visible = True

Sheets("3").Visible = True

Sheets("4").Visible = True

Sheets("5").Visible = True

Sheets("data").Visible = True

Sheets("Tasks").Visible = True

Dim t As Single

t = Timer

    MakeTable ' Заполнение результирующей таблицы

    Sheets("Settings").Select

    t = Timer - t

    UpdateMsg = MsgBox("Обновление прошло успешно за " & Round(t, 0) & \_

" секунд", vbOKOnly, "Обновление")

Sheets("1").Visible = False

Sheets("2").Visible = False

Sheets("3").Visible = False

Sheets("4").Visible = False

Sheets("5").Visible = False

Sheets("data").Visible = False

Sheets("Tasks").Visible = False

Application.ScreenUpdating = True

End Sub

2.9 Тестирование технической части

По итогам технического тестирования от бизнеса получены следующие вводные:

* Необходимо ускорить выполнение второй части макроса по формированию результирующей таблицы. На данный момент на этот процесс уходит в среднем 12 минут, что не позволяет быстро менять входные параметры, особенно в случае, когда пользователь выгрузил датасет с меньшим количеством параметров, от того, что планировал или просто ошибся в параметрах
* Необходимо добавить поле Сумма с результирующим значением всех продаж по артикулу
* Необходимо добавить поле Маржа с результирующим значением всей заработанной маржи по каждому артикулу

Данные комментарии взяты в работу и для выполнения этих требований требуется внести изменения в модуль MakeTable().

2.10 Улучшение быстродействия

Итогом работы над модулем стало изменение подхода для результирующего подсчета значений продаж – отказ от использования цикла и использование вставки готовой формулы **СУММЕСЛИМН**, и дальнейшим масштабированием на весь диапазон необходимых ячеек. Использование внутренних функций EXCEL позволило сократить время выполнение всего модуля до 10-15 секунд или в **50-60 раз** в сравнении с изначальным временем исполнения.

Кроме того, при финальном формировании списка добавлены необходимые поля Сумма и Маржа.

Весь результирующий список копируется на новый лист для последующей работы пользователя.

Полный код модуля:

Sub MakeTable()

'

' Формирование результирующей таблицы

'

'объявляем счетчик

Dim Counter As Integer

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Очистка листов Work и Tasks перед загрузкой новых данных

Sheets("Work").Select

Cells.Select

Selection.Clear

'убираем закрепление границ

ActiveWindow.FreezePanes = False

Cells.Select

Cells.FormatConditions.Delete

Cells(1, 1).Select

Sheets("Tasks").Select

Columns("C:C").Select

Range(Selection, Selection.End(xlToRight)).Select

Selection.Delete Shift:=xlToLeft

Rows("4:4").Select

Range(Selection, Selection.End(xlDown)).Select

Selection.Delete Shift:=xlUp

Cells.Select

'убираем условное форматирование со всего листа

Cells.FormatConditions.Delete

ActiveWindow.FreezePanes = False

Cells(1, 1).Select

'----------------------------------------------------------------------------------

'Заполнение списка артикулов

Sheets("3").Select

LastRow = Cells(Rows.Count, 1).End(xlUp).Row

Range(Cells(1, 1), Cells(LastRow, 2)).Select

Selection.Copy

Sheets("Tasks").Select

Cells(4, 1).Select

ActiveSheet.Paste

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Заполнение списка всех менеджеров слева направо

Sheets("2").Select

Number\_of\_managers = Cells(Rows.Count, 1).End(xlUp).Row

'используем цикл для заполнения любого возможного количества менеджеров

Counter = 1

Do While Counter <= Number\_of\_managers

    Sheets("2").Select

    Cells(Counter, 2).Select

    Selection.Copy

    Sheets("Tasks").Select

    LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

    Cells(3, LastCol).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Counter = Counter + 1

Loop

'----------------------------------------------------------------------------------

'Копируем номер филиала в привязке к менеджеру

LastCol\_managers = Cells(3, Columns.Count).End(xlToLeft).Column

Cells(2, 2).Select

Selection.Copy

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Cells(2, 2).Select

Selection.Copy

Range(Cells(2, 3), Cells(3, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.Copy

Range(Cells(2, 3), Cells(2, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Заполняем суммы продаж по каждому менеджеру, учитывая фильтрацию по сегментам и подсегментам

'

'!новая реалиазация через вставку готовой формулы

'!время выполнения уменьшено в 50 раз!

'!текущий блок сокращен на 70 строк

LastRow = Cells(Rows.Count, 1).End(xlUp).Row

Sheets("Tasks").Select

' Использование глобальных переменных отменено по ТЗ, реализована возможность настройки

' пользователем номеров столбцов во входном файле, так как у части учетных записей

' данные могут выгружаться в другом порядке, загружаем с листа Settings

Sum\_column = Sheets("Settings").Cells(5, 6).Value

Article\_column = Sheets("Settings").Cells(4, 6).Value

Manager\_column = Sheets("Settings").Cells(3, 6).Value

Brunch\_column = Sheets("Settings").Cells(6, 6).Value

SubBrunch\_column = Sheets("Settings").Cells(7, 6).Value

'копируем значения сегментов и подсегментов для фильтрации данных

Branch\_Value = Sheets("Settings").Range("J2").Value

SubBranch\_Value = Sheets("Settings").Range("J5").Value

'для удобства внесения формулы меняем нотацию

Application.ReferenceStyle = xlR1C1

If Branch\_Value = "" And SubBranch\_Value = "" Then

    Range("B1").FormulaR1C1Local = "=СУММЕСЛИМН(data!C" & Sum\_column & \_

    ";data!C" & Article\_column & ";Tasks!RC1;data!C" & Manager\_column & ";Tasks!R3C)"

ElseIf Branch\_Value = "" And SubBranch\_Value <> "" Then

    Range("B1").FormulaR1C1Local = "=СУММЕСЛИМН(data!C" & Sum\_column & \_

    ";data!C" & Article\_column & ";Tasks!RC1;data!C" & Manager\_column & \_

    ";Tasks!R3C;data!C" & SubBrunch\_column & ";Settings!R5C10)"

ElseIf SubBranch\_Value = "" And Branch\_Value <> "" Then

    Range("B1").FormulaR1C1Local = "=СУММЕСЛИМН(data!C" & Sum\_column & \_

    ";data!C" & Article\_column & ";Tasks!RC1;data!C" & Manager\_column & \_

    ";Tasks!R3C;data!C" & Brunch\_column & ";Settings!R2C10)"

Else

    Range("B1").FormulaR1C1Local = "=СУММЕСЛИМН(data!C" & Sum\_column & \_

    ";data!C" & Article\_column & ";Tasks!RC1;data!C" & Manager\_column & \_

    ";Tasks!R3C;data!C" & Brunch\_column & ";Settings!R2C10;data!C" & SubBrunch\_column & ";Settings!R5C10)"

End If

'возращаем нотацию по умолчанию

Application.ReferenceStyle = xlA1

'копируем формулу, вставляем формат ячейки и оставляем только значения без формул

Cells(1, 2).Select

Selection.Copy

Range(Cells(4, 3), Cells(LastRow, LastCol\_managers)).Select

Selection.PasteSpecial Paste:=xlPasteFormulas, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Selection.Copy

Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

Application.CutCopyMode = False

'----------------------------------------------------------------------------------

'Подсвечиваем 0 условных форматированием на всем диапазоне

Range(Cells(4, 3), Cells(LastRow, LastCol\_managers)).Select

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, \_

    Formula1:="=0"

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

With Selection.FormatConditions(1).Font

    .Color = -16383844

    .TintAndShade = 0

End With

With Selection.FormatConditions(1).Interior

    .PatternColorIndex = xlAutomatic

    .Color = 13551615

    .TintAndShade = 0

End With

Selection.FormatConditions(1).StopIfTrue = False

'----------------------------------------------------------------------------------

'Добавляем счет продаж по каждому филиалу

Sheets("1").Select

Number\_of\_stores = Cells(Rows.Count, 1).End(xlUp).Row

Counter = 1

Do While Counter <= Number\_of\_stores

    Sheets("1").Select

    Cells(Counter, 1).Select

    Selection.Copy

    Sheets("Tasks").Select

    LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

    Cells(3, LastCol).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Counter = Counter + 1

Loop

'----------------------------------------------------------------------------------

'Считаем количество продающих менеджеров

LastCol = Cells(4, Columns.Count).End(xlToLeft).Column

Cells(4, LastCol).Select

'-----------------------------------------------------------------

'Переменная для определения границ сумм столбца Ранг

LastCol\_for\_sum = Cells(4, Columns.Count).End(xlToLeft).Column + 1

'-----------------------------------------------------------------

'реализация через цикл, так как не большой объем вычислений

Counter = 1

For i = 1 To Number\_of\_stores

    For j = 4 To LastRow

        Cells(j, i + LastCol).Value = WorksheetFunction.CountIfs(Range(Cells(j, 3), Cells(j, LastCol)), ">0", Range(Cells(2, 3), Cells(2, LastCol)), Cells(3, LastCol + Counter).Value)

    Next

    Counter = Counter + 1

Next

'----------------------------------------------------------------------------------

'Добавляем столбец Ранг

LastCol = Cells(3, Columns.Count).End(xlToLeft).Column + 1

Cells(3, LastCol).Value = "Ранг"

For i = 4 To LastRow

    Cells(i, LastCol).Value = WorksheetFunction.Sum(Range(Cells(i, LastCol\_for\_sum), Cells(i, LastCol)))

Next

'----------------------------------------------------------------------------------

'Изменяем форматирование вновь созданных столбцов

Cells(2, 2).Select

Selection.Copy

Range(Cells(3, LastCol\_for\_sum), Cells(3, LastCol)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Cells(1, 2).Select

Selection.Copy

Range(Cells(4, LastCol\_for\_sum), Cells(LastRow, LastCol)).Select

Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

Application.CutCopyMode = False

Range(Cells(3, LastCol), Cells(LastRow, LastCol)).Select

Selection.Font.Bold = True

With Selection.Interior

    .Pattern = xlSolid

    .PatternColorIndex = xlAutomatic

    .ThemeColor = xlThemeColorAccent5

    .TintAndShade = 0.599993896298105

    .PatternTintAndShade = 0

End With

'----------------------------------------------------------------------------------

'Копирование всей страницы на новый лист для удобства работы

    Range(Cells(2, 1), Cells(LastRow, LastCol)).Select

    Selection.Copy

    Sheets("Work").Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

    Selection.Borders(xlDiagonalDown).LineStyle = xlNone

    Selection.Borders(xlDiagonalUp).LineStyle = xlNone

    With Selection.Borders(xlEdgeLeft)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeTop)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeBottom)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlEdgeRight)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlInsideVertical)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    With Selection.Borders(xlInsideHorizontal)

        .LineStyle = xlContinuous

        .ColorIndex = xlAutomatic

        .TintAndShade = 0

        .Weight = xlThin

    End With

    'автоматическая настройка ширины и внешнего вида отображения

    Range(Cells(2, LastCol\_for\_sum), Cells(LastRow, LastCol)).EntireColumn.Select

    Selection.Cut

    Columns("C:C").Select

    Selection.Insert Shift:=xlToRight

    Application.CutCopyMode = False

    Range(Cells(2, 1), Cells(LastRow, LastCol)).EntireColumn.AutoFit

    ActiveWindow.Zoom = 70

    ColFilter = 0

    '----------------------------------------

    'Ищем столбец Ранг для будущей сортировки

    For i = 1 To LastCol

        If Cells(2, i).Value = "Ранг" Then

            ColFilter = i

            Exit For

        End If

    Next

    '----------------------------------------

    'добавляем фильтры и сортируем по столбцу Ранг по убыванию

    Rows("2:2").Select

    Selection.AutoFilter

    ActiveWorkbook.Worksheets("Work").AutoFilter.Sort.SortFields.Clear

    ActiveWorkbook.Worksheets("Work").AutoFilter.Sort.SortFields.Add2 Key:=Range( \_

        Cells(2, ColFilter), Cells(2, ColFilter)), SortOn:=xlSortOnValues, Order:=xlDescending, DataOption:= \_

        xlSortNormal

    With ActiveWorkbook.Worksheets("Work").AutoFilter.Sort

        .Header = xlYes

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

    Cells(3, ColFilter + 1).Select

    ActiveWindow.FreezePanes = True

    '------------------------------------------------------------------------------

    'Добавление столбца Сумма

    Columns("C:C").Select

    Selection.Insert Shift:=xlToRight

    Cells(2, 3).Value = "Сумма"

    Cells(2, 2).Select

    Selection.Copy

    Cells(2, 3).Select

    Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

    Application.CutCopyMode = False

    LastCol\_Total = Cells(3, Columns.Count).End(xlToLeft).Column

    LastRow\_Total = Cells(Rows.Count, 1).End(xlUp).Row

    Counter = 1

    For i = 3 To LastRow\_Total

        Cells(i, 3).Value = WorksheetFunction.Sum(Range(Cells(i, ColFilter + 2), Cells(i, LastCol\_Total)))

        Counter = Counter + 1

    Next

    Columns("C:C").Select

    Selection.NumberFormat = "#,##0"

    With Selection

        .HorizontalAlignment = xlCenter

        .VerticalAlignment = xlBottom

        .WrapText = False

        .Orientation = 0

        .AddIndent = False

        .IndentLevel = 0

        .ShrinkToFit = False

        .ReadingOrder = xlContext

        .MergeCells = False

    End With

    Columns("C:C").EntireColumn.AutoFit

    '------------------------------------------------------------------------------

    'Добавление столбца Маржа

    'определяем номер столбца для нотации R1C1

    '(номер столбца на листе data) - (номер колонки, в которую записываем значения)

    Gross\_column = (Sum\_column + 1) - 4 '4 остается неизменным

    Columns("D:D").Select

    Selection.Insert Shift:=xlToRight

    Cells(2, 4).Value = "Маржа"

    Cells(2, 2).Select

    Selection.Copy

    Cells(2, 4).Select

    Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, \_

        SkipBlanks:=False, Transpose:=False

    Application.CutCopyMode = False

    'Считаем маржу по каждому артикулу

    Application.ReferenceStyle = xlR1C1

    Counter = 1

    For i = 3 To LastRow\_Total

        Cells(i, 4).FormulaR1C1Local = "=СУММЕСЛИМН(data!C[" & Gross\_column & "];data!C;Work!RC[-3])"

        Counter = Counter + 1

    Next

    Application.ReferenceStyle = xlA1

    'форматируем данные к требуемому виду

    Columns("D:D").Select

    Selection.NumberFormat = "#,##0"

    With Selection

        .HorizontalAlignment = xlCenter

        .VerticalAlignment = xlBottom

        .WrapText = False

        .Orientation = 0

        .AddIndent = False

        .IndentLevel = 0

        .ShrinkToFit = False

        .ReadingOrder = xlContext

        .MergeCells = False

    End With

    Columns("D:D").EntireColumn.AutoFit

    'снимаем формулы

    Range(Cells(3, 4), Cells(LastRow\_Total, 4)).Select

    Selection.Copy

    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks \_

        :=False, Transpose:=False

    Application.CutCopyMode = False

    '------------------------------------------------------------------------------

    Cells(1, 1).Select

    Selection.Copy

    Cells(1, 2).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False

End Sub

Глава 3. Взаимодействие с рабочей группой для выпуска инструмента

3.1 Определение KPI и сроков тестирования

Поскольку все требования бизнеса по ТЗ и итогам технического тестирования были учтены было принято решение о раскатке инструмента на часть филиалов и определение другой группы филиалов для тестирования. Поскольку основным KPI бизнеса является сумма выручки были решено разбить филиалы так, чтобы суммы по последнему месяцу в каждой выборке перед внедрением были примерно равны:





Рис 5. Итоговые выборки по филиалам

Ввиду этого, также был определен **основной KPI** – **прирост месяц к месяцу в рублях и %** в каждой из указанных выборок, так как по сезонным историческим данным компании стартовый месяц для анализа и контрольный месяц не отличаются существенным ростом продаж.

Дополнительным KPI было принято решение принять **сокращение количества 0 артикулов**, для проверки второй гипотезы.

Срок тестирования: 1 месяц

Срок подготовки и обучение сотрудников: 1 месяц

Итого первичная выгрузка формируется за Июнь, контрольная выгрузка должна содержать данные за Август. Июль промежуточный месяц для обучения сотрудников и на местах.

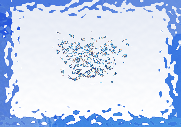
3.2 Инструкция по обучению

Ниже привожу инструкцию для запуска макроса. Инструкция по работе с результирующей таблицей была рассказана в ходе виде конференций на конкретных примерах.

**Инструкция по выгрузке входящего файла для макроса**

(ввиду использования внутренних систем, ко всем скриншотам применены эффекты размытия)

1. Заходим в систему отчетности и выбираем группу отчетов как на рисунке:



1. Далее в пункте 4 выбираем необходимые филиалы фактической отгрузки товара – **можно варьировать**:

Изображение выглядит как текст

Автоматически созданное описание

1. Далее в пункте 7 выбираем следующие метрики – **не изменять**:



1. Далее в пункте 15 выбираем следующие метрики для фильтрации требуемых филиалов продажи по менеджерам – **можно варьировать**:

Изображение выглядит как текст, дизайн

Автоматически созданное описание

1. Далее в пункте 17 выбираем следующие метрики для отображения – **не изменять**:

Изображение выглядит как текст, чек, снимок экрана

Автоматически созданное описание

1. Далее в пункте 21 выбираем следующие метрики для отображения продаж и маржи – **в текущей версии не изменять**:

Изображение выглядит как текст

Автоматически созданное описание

1. И в финальной выгрузке необходимо детализировать отчетность кликнув на указанное место и нажав указанную кнопку:

Изображение выглядит как снимок экрана

Автоматически созданное описание

1. Сделать экспорт в EXCEL и сохранить файл под названием **data.xlsx** и разместить его в ту же папку, где находится файл **ProjectX.xlsm**
2. Открываем файл **ProjectX.xlsm** и нажимаем кнопку **1. Загрузка данных:**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

1. Далее выставляем нужные **Сегмента или Подсегменты** для детализации и нажимаем кнопку **2. Создание таблицы (опционально –** если оставить пустыми, выгрузятся данные по всем сегментам и подсегментам**)**

3.4 Подведение итогов

По результатам тестирования инструмента в двух выборках были достигнуты следующие результаты, подготовленные на таблицах **Before** и **After** в разбивке выборок и менеджеров в отношении к первому месяцу анализа:



Рис 6. Итоги тестирования инструмента.

Если, не учитывать аномалию менеджера 1, который в Июне был в длительном отпуске, видно, что филиалы из первой выборки в целом показывают значительный прирост в выручке относительно первого месяца выгрузки отчета.

Сведем таблицу по филиалам:



Рис. 7. Сравнительная таблица по филиалам.

Филиалы из первой выборки демонстрируют прирост в выручке – 25% в среднем, в то время как филиалы, где не применялся новый подход с отчетностью показывают лишь 7% прирост продаж.

Что касается артикулов с 0 продажами также видна лучшая динамика в фиалах из первой выборки – сокращение в целом не продаваемых артикулов на 2075 штук, против сокращения 511 артикулов в выборке номер 2.

Итоговая таблица по двум выборкам:



Рис. 8. Итоговая таблица по выборкам

**Заключение**

По результатам тестирования руководству было продемонстрировано значительное изменение результатов у филиалов из первой выборки, против выборки номер два.

Использование инструмента позволяет сделать вывод, что магазины первой выборки с очищенным от стандартного прироста (+7%) заработали дополнительные 2,94 млн. рублей выручки и 700 тыс. рублей маржи.



Рис. 8. Расчет дополнительного дохода и маржи

Также использование инструменты позволяет в 4 раза лучше работать с непродаваемыми артикулами, за счет фокусировки менеджеров на списках 0 артикулов при работе с клиентами.

Все достигнутые результаты косвенно подтверждают обе гипотезы, но для точных выводов необходима четкая система аналитики, которая разделить приросты за счет увеличения продаж продаваемых артикулов и приросты за счет продаж непродаваемых артикулов.

Несмотря на то, что на данный момент, недостаточно данных, чтобы доподлинно подтвердить первую или вторую гипотезы, бизнесу очевидно, что использование инструмента позволяет получить прирост в выручке в среднем на 17%, что больше совокупных ожиданий на 2%. И при дальнейшем внедрении во внутренние системы важно зафиксировать разделение на причины прироста, чтобы делать правильные выводы.

На данный момент готовится техническое задание на согласование и дальнейшее внедрение инструмента во внутренний контур компании и раскатка его на следующие выборки магазинов.

**Предложения по совершенствованию самого инструмента**

**В случае продолжения тестирования на базе EXCEL, если на совете директоров будет дана задача дополнительно протестировать инструмент:**

* + - * Добавить модуль для открытия любого файла, а не определенного названия через отдельный модуль, чтобы исключить возможные ошибки с размещением не в той папке
      * Создание модуля для автоматического назначения заданий сразу всем менеджерам
      * Разработать методику разделения результатов по причинам прироста – продаваемые и непродаваемые ранее артикулы
      * Разработать модуль для автоматизации аналитики для отслеживания динамики

**В случае интеграции во внутренние системы:**

* + - * Разработать методику разделения результатов по причинам прироста – продаваемые и непродаваемые ранее артикулы
      * Сформировать техническое задание на создание Dashboard для отслеживания результатов ad-hoc и в динамике

**Список используемой литературы**

* *Соглашения о структуре программы и коде (Visual Basic) -* [*https://learn.microsoft.com/ru-ru/dotnet/visual-basic/programming-guide/program-structure/program-structure-and-code-conventions*](https://learn.microsoft.com/ru-ru/dotnet/visual-basic/programming-guide/program-structure/program-structure-and-code-conventions)
* *Книга: Павлов Николай. Excel – готовые решения. Бери и пользуйся. Город: Москва. Книга по Требованию, 2014.*
* *Книга: Павлов Николай. Microsoft Excel: Мастер Формул. Город: Москва. Книга по Требованию, 2017.*

**Глоссарий**

1MVP (Minimum viable product) - Минимально жизнеспособный продукт

2KPI (Key performance indicators,) - Ключевые показатели эффективности

**Приложения**

**Ссылка на проект:**

[**https://github.com/s-metrolog-s/Final-diploma-Data**](https://github.com/s-metrolog-s/Final-diploma-Data)

По ссылке на Github проекта выложены как артефакты следующие файлы:

* Результирующая таблица Before
* Результирующая таблица After
* Файл EXCEL с макросом
* Инструкция по работе с файлом
* Датасет Before
* Датасет After
* Схема бизнес-процесса
* Код VBA всех модулей