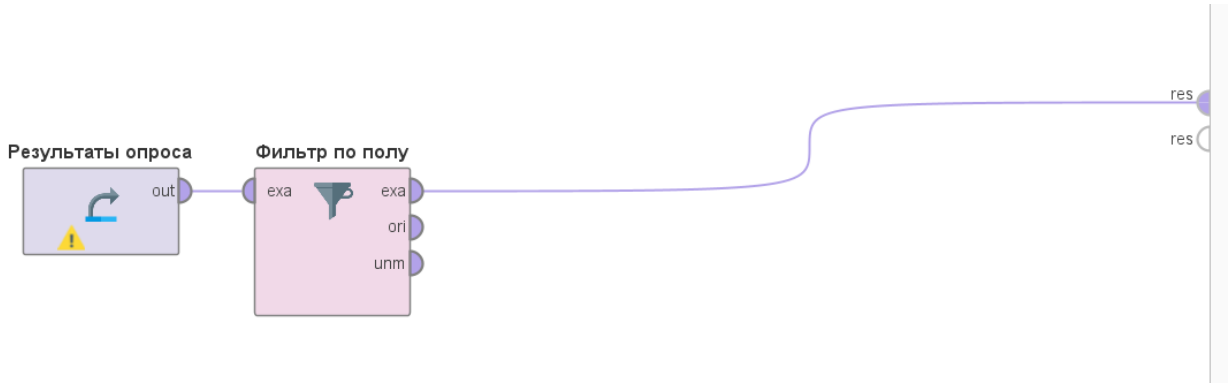
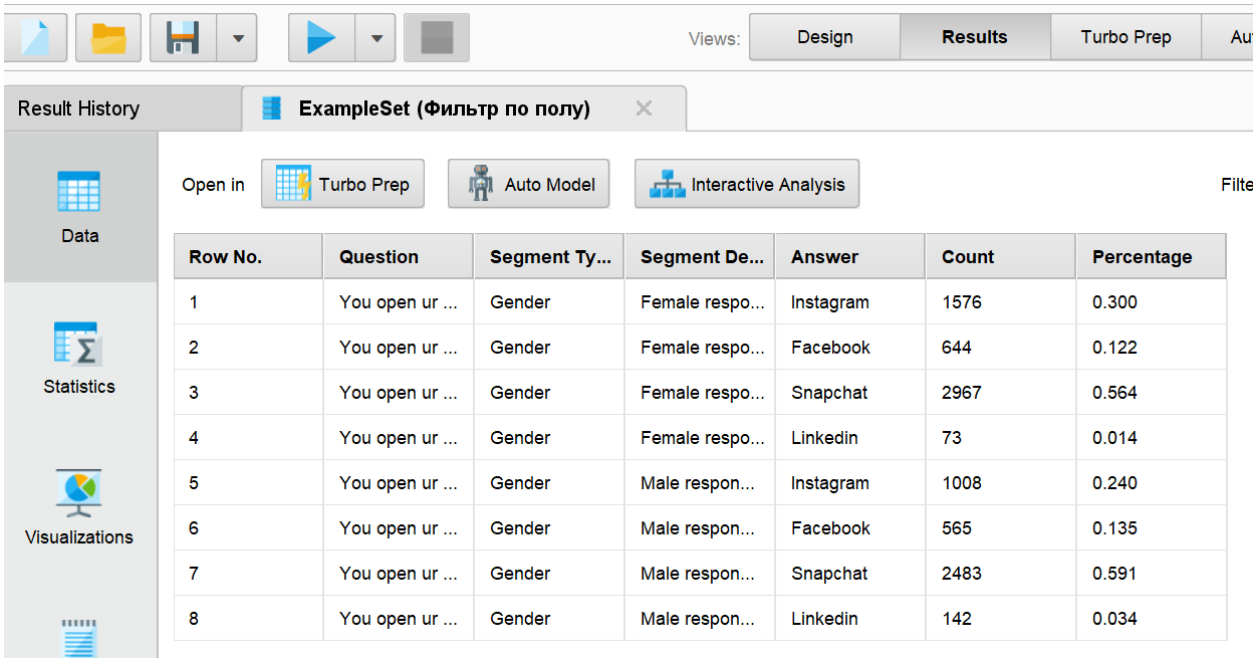
RapidMiner — это мощная и многопользовательская платформа, она служит для создания, передачи и обслуживания наукоемких данных. Эта платформа предлагает больше функций, чем любое другое визуальное решение, к тому же она открыта и расширяема для поддержки всех потребностей научных данных. RapidMiner ускоряет создание полных аналитических рабочих процессов — от подготовки данных до моделирования до развертывания бизнеса — в единой среде, значительно повышая эффективность и сокращая время, необходимое для проектов в области данных. Если сравнивать RapidMiner с другими программами, то у RM гораздо шире функциональные возможности по обработке, банально больше узлов.

RapidMiner — **инструмент, созданный для data-майнинга, с основной идеей, что аналитик не должен программировать при выполнении своей работы**. Программу снабдили достаточно хорошим набором операторов решающих большой спектр задач получения и обработки информации из разнообразных источников (базы данных, файлы и т.п.), и можно с уверенностью говорить, что это ещё и полноценный инструмент для ETL (Extract, Transform, Load) (Извлекать, преобразовывать, загружать).

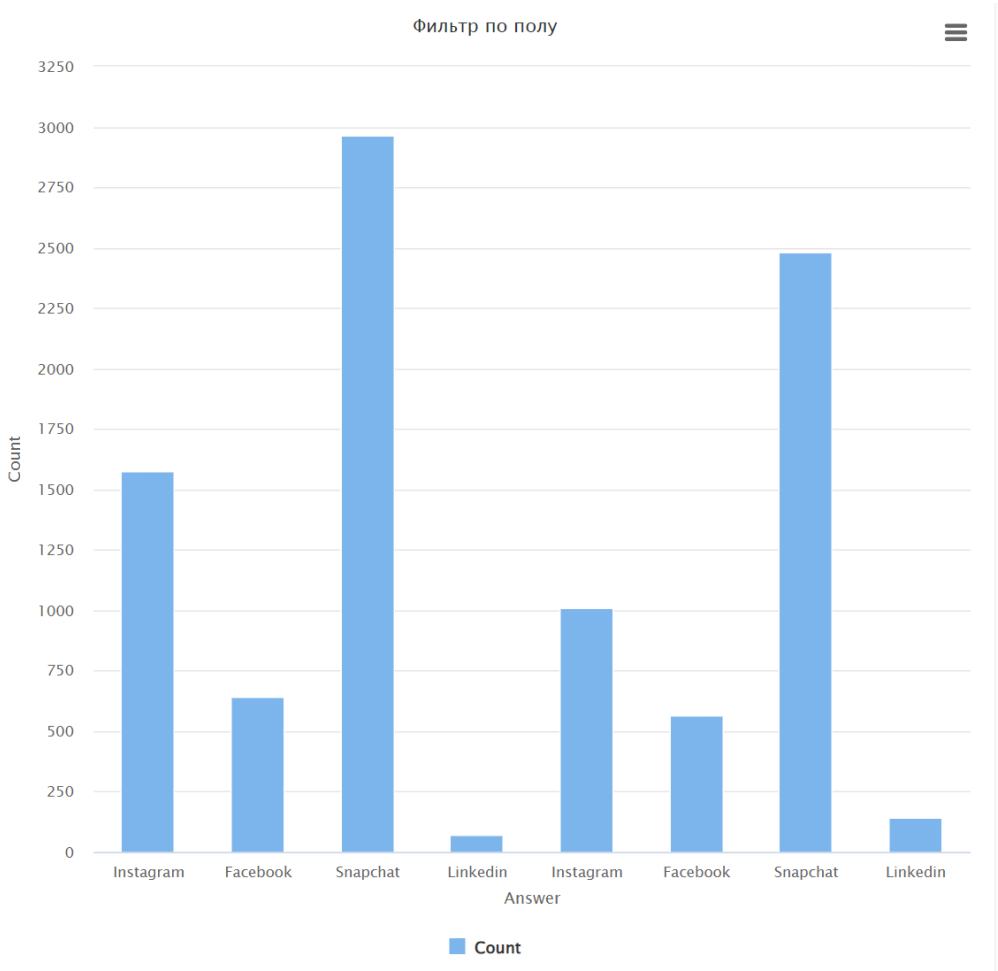
1. Гендерная статистика
   1.1. Допустим, мы хотим узнать результаты того, как ответили мужчины, и как ответили женщины. Для этого необходимо произвести несколько преобразований.
   1.2. Перейдите на вкладку Design и перетащите файл на вкладку Process, чтобы приступить к анализу. Назовите импортированные данные «Результаты опроса».
   1.3. Отфильтруем данные, оставив только Gender в качестве Segment Type.
      1.3.1. Для этого из вкладки Operators перетащим Filter Examples.
      1.3.2. Свяжем конец out блока «Результаты опроса» с входом exa фильтра.
      1.3.3. Дважды кликнув по фильтру или нажав на Add Filters во вкладке Parameters, настроим фильтр так, как мы обговорили в начале этого абзаца.
      1.3.4. Назовите фильтр «Фильтр по полу».
   1.4. Чтобы посмотреть результат фильтрации, соедините выход фильтра с точкой res на вкладке process.
   1.5. Запуск процесса производится нажатием кнопки «Start» (синей кнопки с треугольником вверху приложения). Вам откроется таблица с результатами фильтрации. Статистика все еще неинформативна. Однако с визуализацией можно поработать.





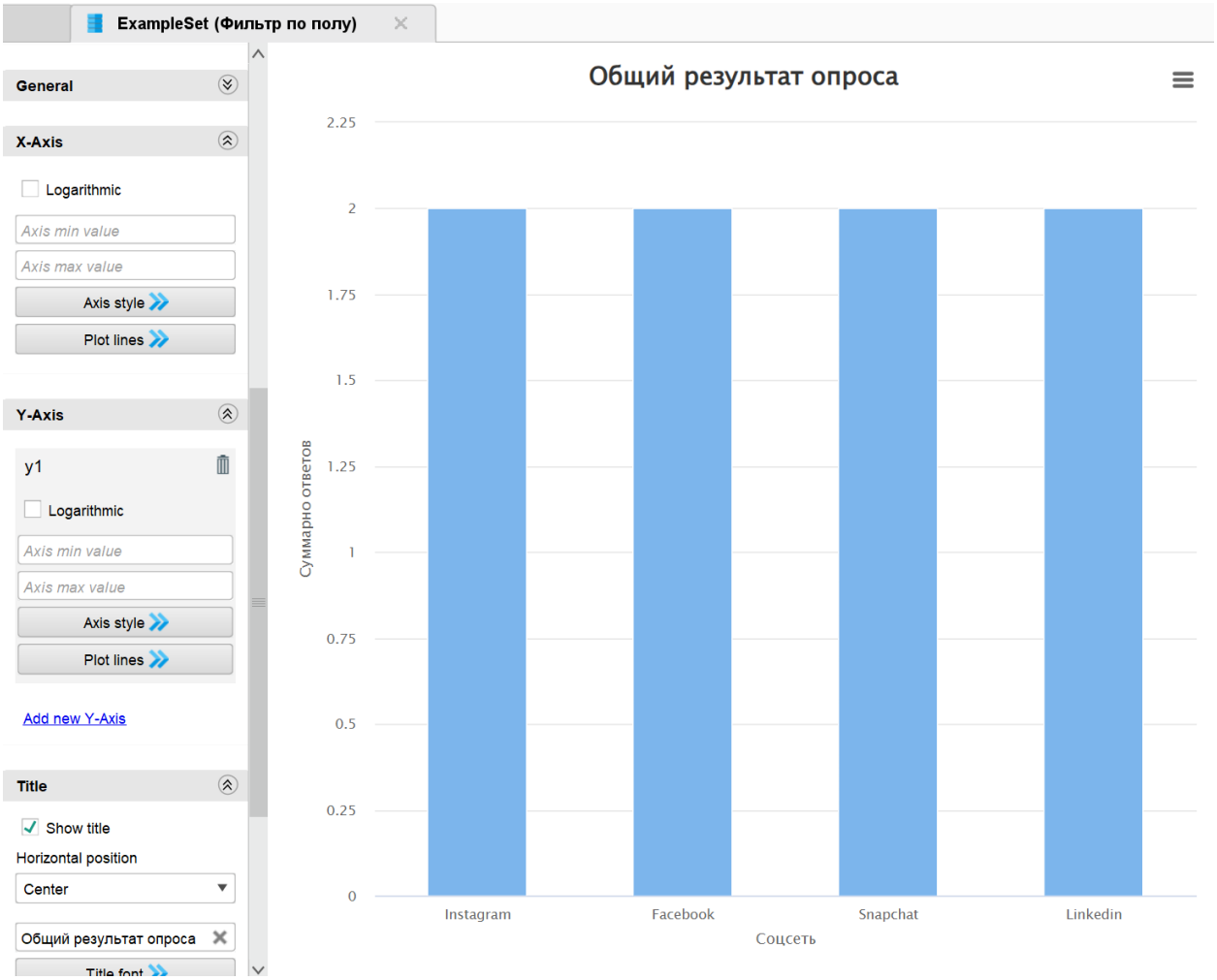
Row No.	Question	Segment Ty...	Segment De...	Answer	Count	Percentage
1	You open ur ...	Gender	Female respo...	Instagram	1576	0.300
2	You open ur ...	Gender	Female respo...	Facebook	644	0.122
3	You open ur ...	Gender	Female respo...	Snapchat	2967	0.564
4	You open ur ...	Gender	Female respo...	Linkedin	73	0.014
5	You open ur ...	Gender	Male respon...	Instagram	1008	0.240
6	You open ur ...	Gender	Male respon...	Facebook	565	0.135
7	You open ur ...	Gender	Male respon...	Snapchat	2483	0.591
8	You open ur ...	Gender	Male respon...	Linkedin	142	0.034

1.6. В качестве типа графика Plot type выберите столбчатую диаграмму Bar (Column). Пусть X-Axis Column будет Answer, Value columns – Count. Уже что-то получается. Но непонятно, где результаты мужчин, а где женщин. Да и хотелось бы, чтобы результаты были сгруппированы по соцсетям для наглядного сравнения результатов по каждой соцсети.



1.7. Для этого поставим галочку Aggregate data. Группируем по ответу (Answer), в качестве Aggregation Function поставьте сумму, ведь мы суммируем кол-во ответов. Таким образом, мы получили общую статистику, показывающую что студента заинтересовало бы в первую очередь. Назовём график «Общий результат опроса». Для этого необходимо зайти в свойства Title. Сделаем жирный шрифт размером 20 в свойствах Title font.

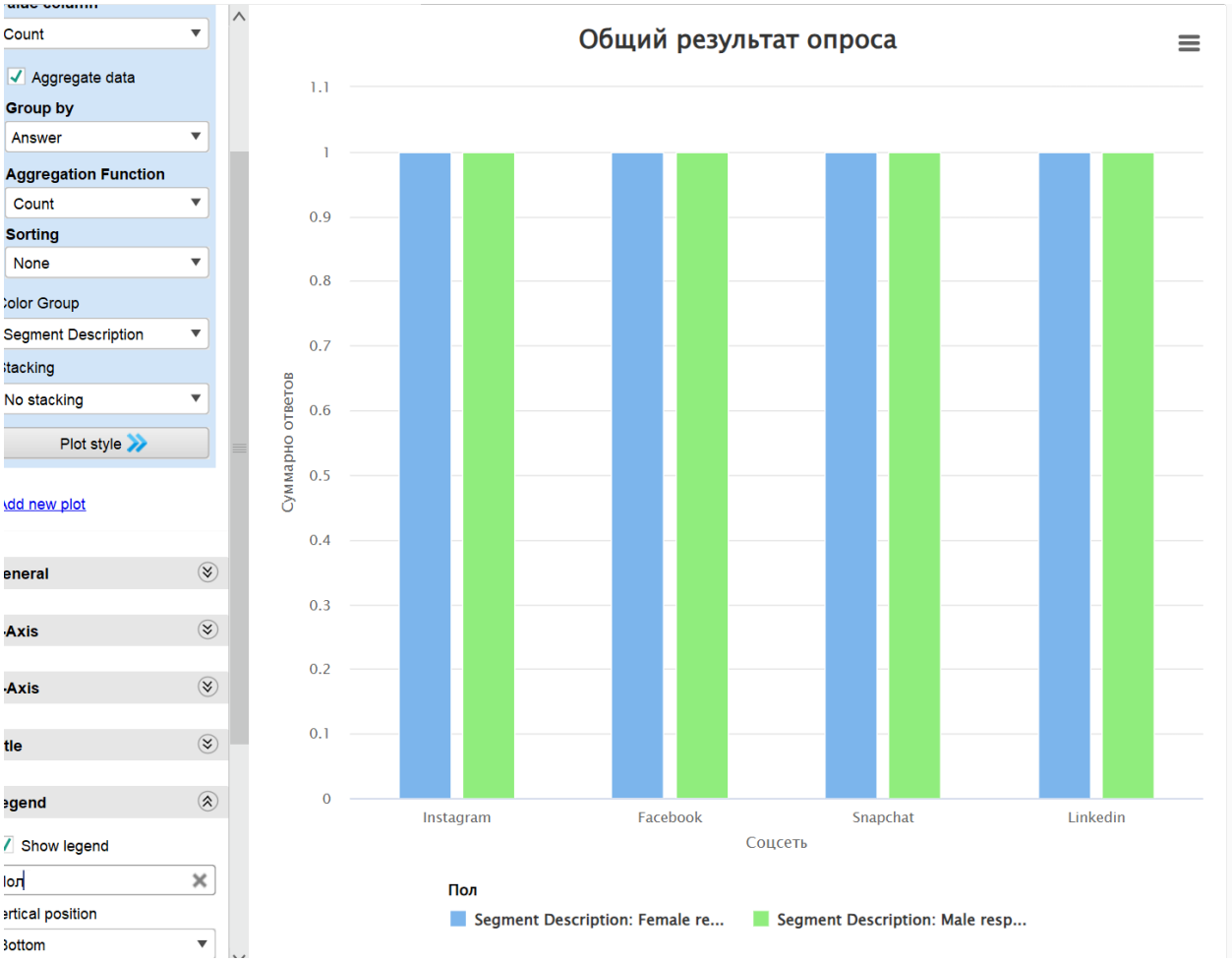
1.8. Переименуем оси в соответствующих пунктах свойств. Пусть ось X – «Соцсеть», ось Y – «Суммарно ответов»



1.9. Уберем легенду, убрав соответствующую галочку. Но ведь мы хотели узнать, как отвечали мужчины и женщины отдельно. Как это увидеть? Для этого выберите Segment Description в поле Color Group.

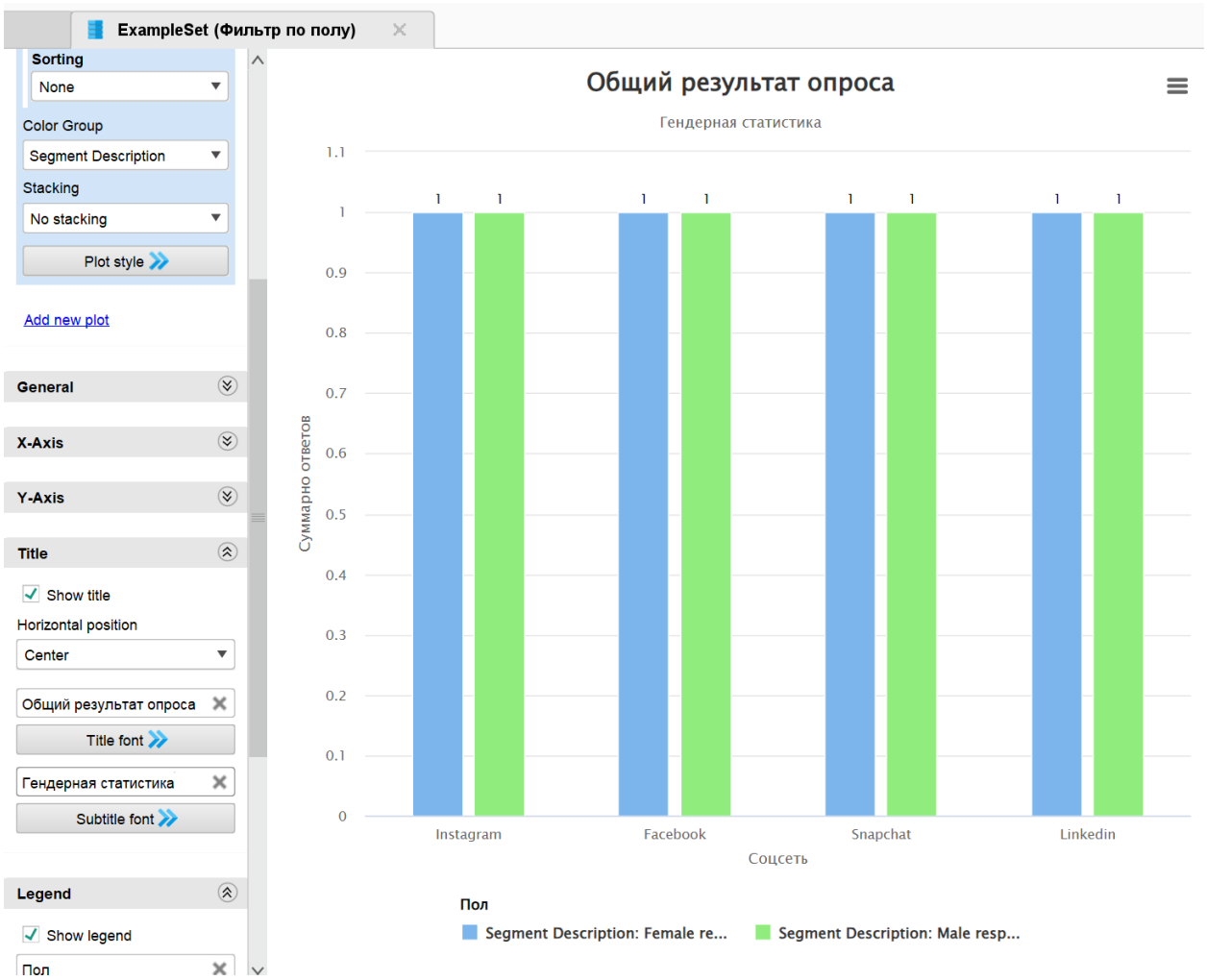
1.10. Все еще непонятно, где мужчины и где женщины. Исправим это, вернув обратно галочку Show legend.

1.11. Установим заголовок легенды «Пол».

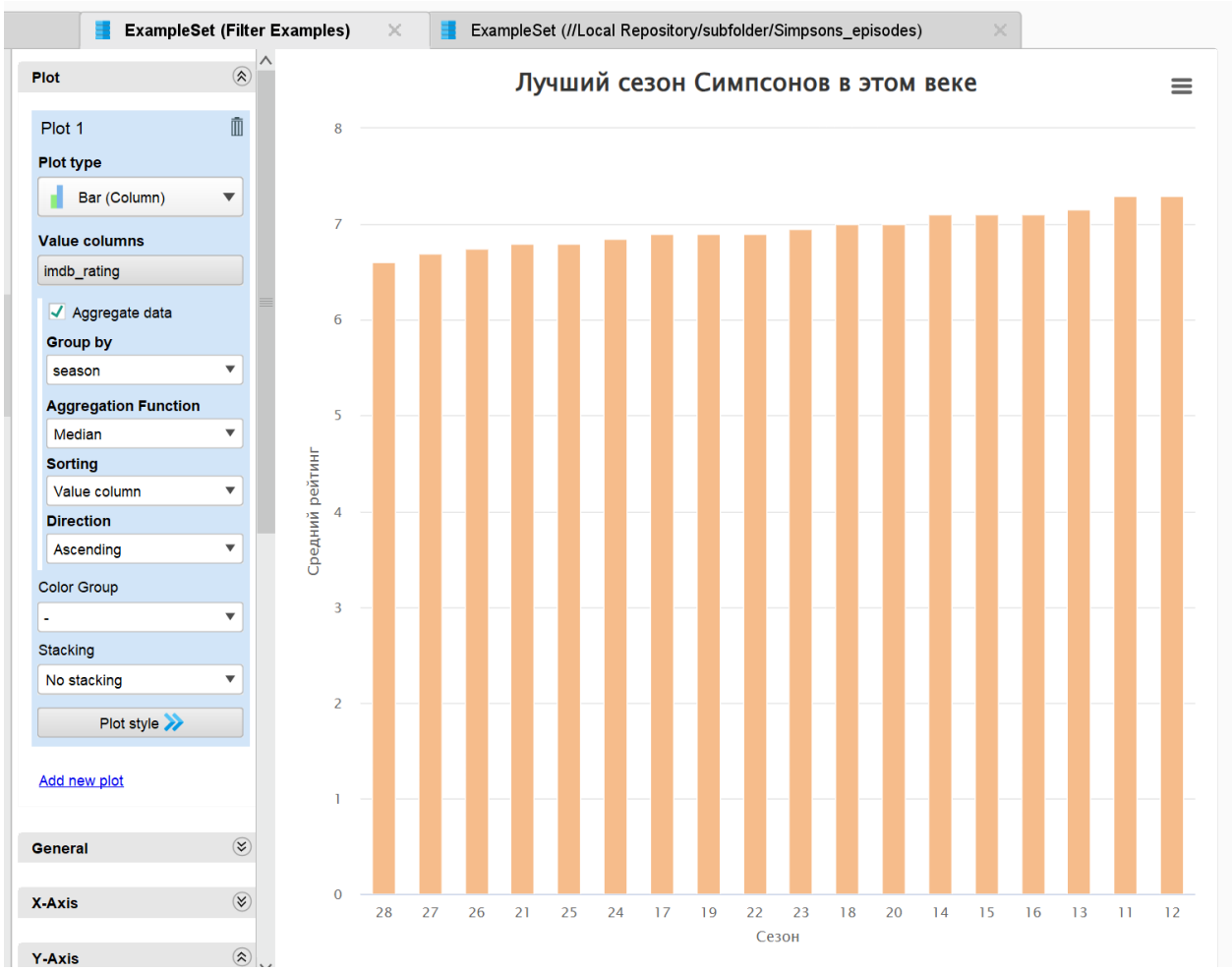


1.12. Хорошо бы сделать график более информативным, добавив точные значения для каждой колонки. Для этого установите галочку Show labels в Plot Style -> Labels Style.

1.13. Ну и в качестве завершающего штриха добавим подзаголовок Subtitle «Гендерная статистика».



2. Задание: Лучший сезон Симпсонов в этом веке
   2.1. Пользуясь изученными инструментами и предоставленными исходными данными в simpsons_episodes.csv, найдите лучший сезон Симпсонов по версии IMDB за последние 20 лет. Для этого визуализируйте информацию таким образом, чтобы было сразу понятно, какой сезон лучший.



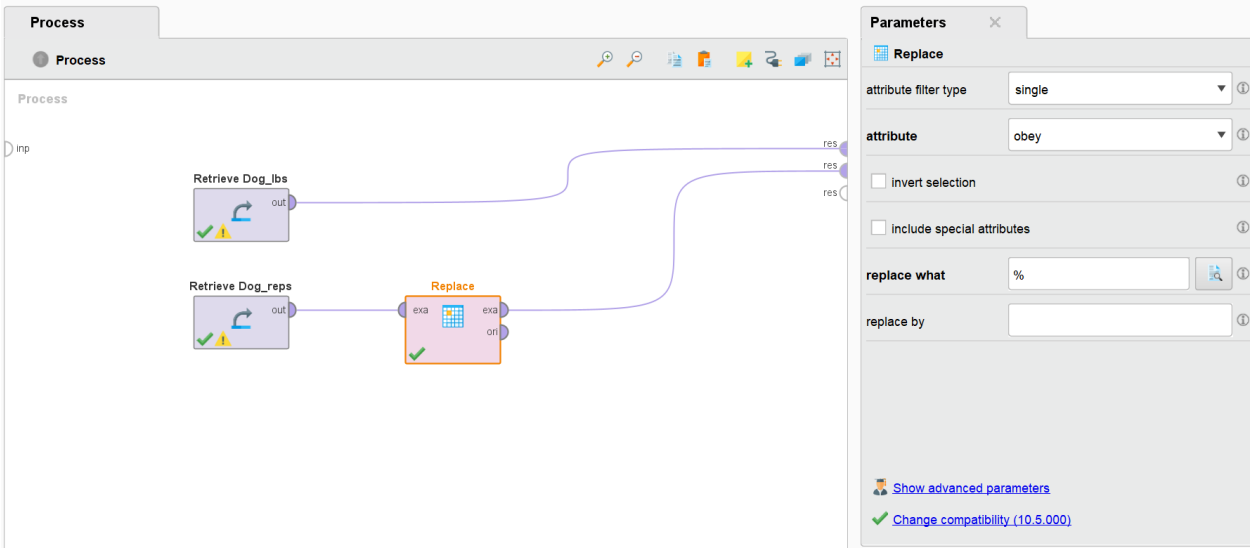
3. Коэффициент корреляции – показатель статистической связи двух атрибутов, изменяется в пределах от -1 до 1. Корреляцию считают:
   Функциональной при $|r|=1$
   Сильной, если коэффициент корреляции $|r|\geq 0{,}7$
   Средней при $|r|\in[0{,}5;0{,}7)$
   Умеренной при $|r|\in[0{,}3;0{,}5)$
   Слабой при $|r|\in[0{,}2;0{,}3)$
   Очень слабой при $|r|<0{,}19$
   Если $r>0$, то корреляционная связь между атрибутами прямая (увеличение значения одного атрибута ведет к увеличению значения другого), при $r<0$ – обратная.
   Корреляционная матрица – квадратная матрица P размера n*n, где n – количество атрибутов. aij – корреляционная связь между iм и jм атрибутом. Все диагональные элементы матрицы P равны 1.
   3.1. Пусть у нас есть два независимых исследования пород собак на предмет обучаемости и послушности собак. Допустим, мы хотим ответить на вопрос: как зависит обучаемость и послушность собаки от её размеров?
   3.2. Для начала нужно посмотреть на данные, с которыми мы будем работать, преобразовать их и объединить. Займёмся этим.
   3.3. Загрузите в окно Process оба набора данных (dog_lbs и dog_reps).
   3.4. Теперь соединим выходы обоих блоков Retrieve с точками res.
   3.5. Запустим процесс. Нам откроются две таблицы с импортированными данными.
   3.6. Становится очевидна одна проблема: в одном исследовании послушность obey собак ввели как десятичную дробь, а в другом преобразовали в проценты, да еще и

приписали сам значок процента. Послушность собаки должна быть одного формата. Самый лёгкий способ – создать новый атрибут obey_new, который будет содержать преобразованное значение послушности. Займёмся этим.

3.7. С помощью оператора Replace заменим в неудобном наборе значок «%» на пустоту « ».

3.8. Для начала соединим выход нашего неудобного набора с входом оператора.

3.9. Выберем атрибут obey и заполним соответствующие значения replace hat и replace by в окне параметров. Подсоединив Replace к точке res и запустив процесс, увидим результат. Все проценты исчезли.



3.10. Но проблемы на этом не заканчиваются. Тип данных у атрибута obey все еще текстовый.

3.11. Чтобы исправить это, подсоединим оператор Parse Numbers и выберем в нем атрибут, в котором нужно преобразовать текст в число.

3.12. Теперь остаётся создать новый атрибут obey_new, в который мы положим выражение, вычисляющее новое значение для старого неудобного атрибута obey.

3.13. Сделать это можно с помощью оператора Generate attributes. В его свойствах введем название нового атрибута и выражение для его вычисления.

3.14. Очевидно, что для того, чтобы данные из двух таблиц совпадали, нам нужно работать с процентом как с десятичной дробью. Для этого в качестве выражение поставим «obey/100».

3.15. Если посмотреть результат, то действительно мы добавили новое поле с соответствующими значениями.

3.16. Раз нам надо, чтобы значения послушности из двух таблиц слились воедино, то столбцы должны иметь одинаковые названия.

3.17. Решить этот костыль можно глупым способом, добавив к вроде бы удобной нам таблице новый атрибут obey_new с такими же значениями, как и в обычном атрибуте obey.

3.18. Или! Можно поступить умнее и просто переименовать сам атрибут. Сделаем это с помощью оператора Rename. Переименуем obey в obey_new, очевидно.

3.19. Теперь мы готовы объединять таблицы. Объединять мы будем уже знакомой из SQL операцией Join. В данном случае, это будет оператор, у которого входы займут наши две таблицы. Какой тип Join выбрать? Если не знаешь, какой тип выбрать, то выбирай Outer Join и справляйся с последствиями. Так сделаем и мы.

В качестве ключевого значения key attribute поставим породу Breed в обоих наборах.

3.20. В результате мы получили одну таблицу с данными из обеих таблиц. Раз у нас есть атрибут obey_new, то атрибут obey нам уже не нужен.

3.21. Выберем только нужные атрибуты оператором Select Attributes. В нашем случае нужные все кроме obey.

3.22. Некоторые значения нам неизвестны – они указаны знаком вопроса «?» в таблице. Перейдя на вкладку «Статистика» можно увидеть, сколько их в таблице в колонке Missing. Это как раз последствия outer join. Как бороться с подобным мусором? Аккуратно смести его под ковёр, пока никто не заметил.

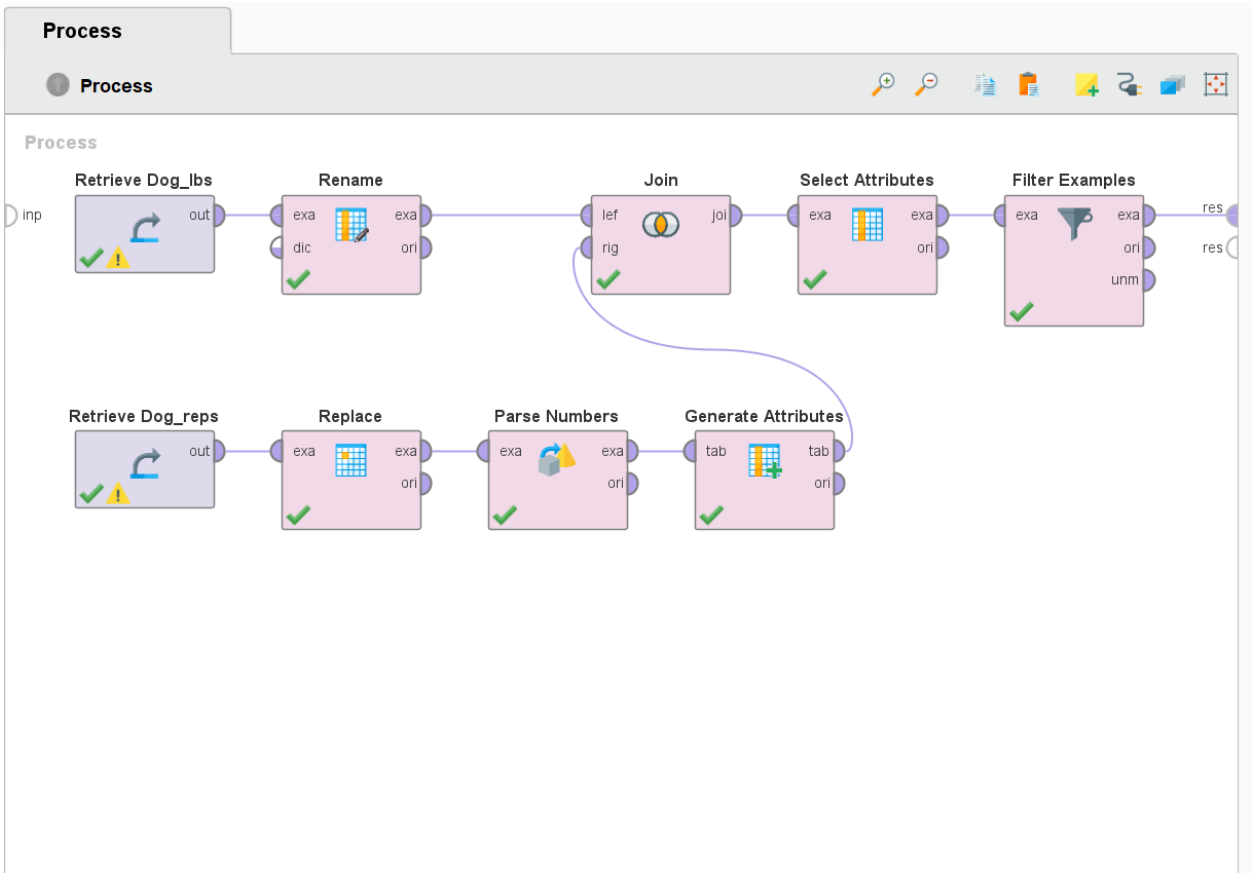
3.23. В нашем случае – отфильтруем набор, оставив только существующие значения.

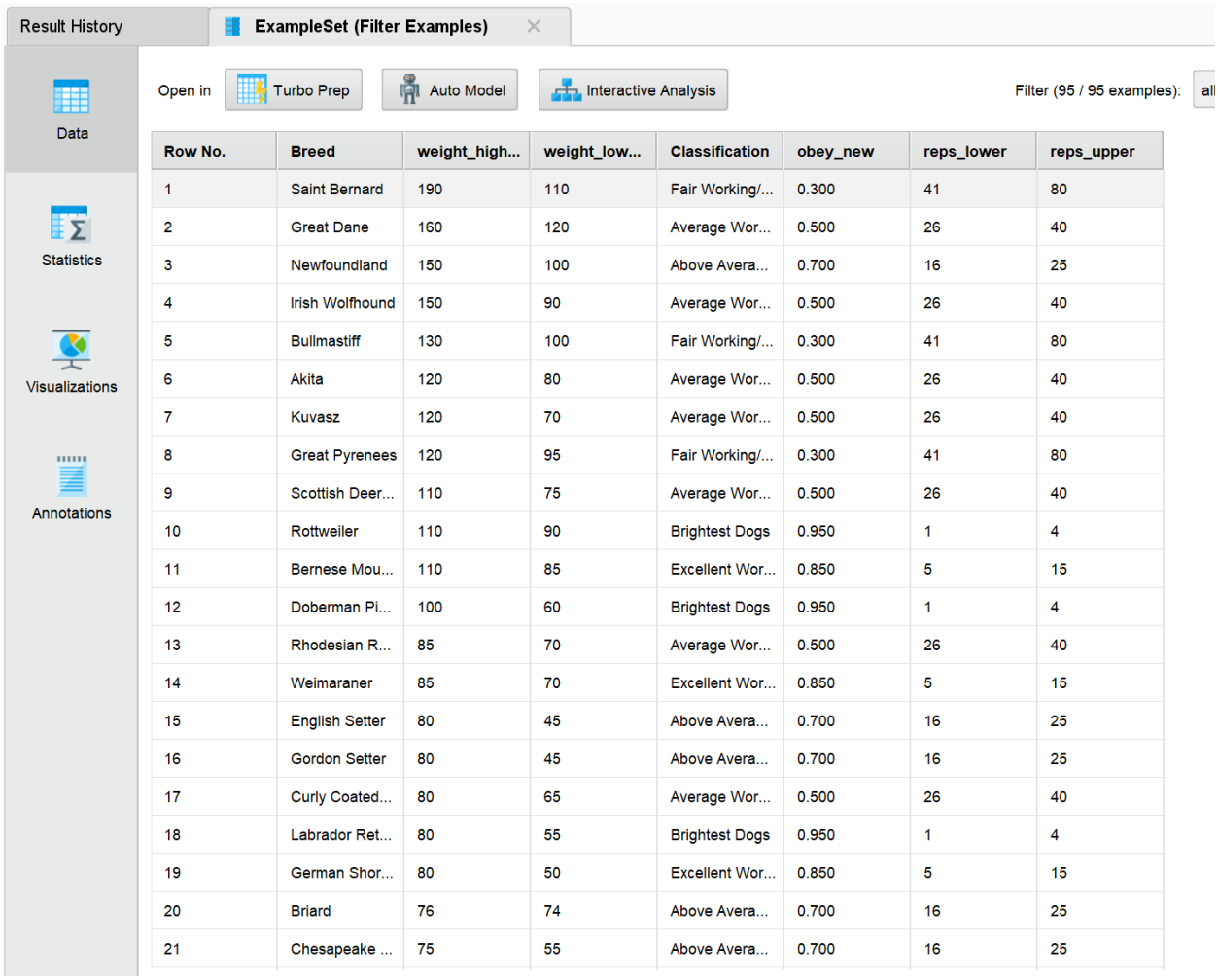
3.24. Для этого после подключения оператора-фильтра укажем в нем те поля, которые содержали «?».

3.25. Для добавления новых полей для фильтрации используйте кнопку Add Entry в окне фильтра.

3.26. В качестве параметра у каждого поля поставьте «is not missing».

3.27. Отлично! Теперь мы подготовили данные к обработке! Можете полюбоваться своей работой, подключив выход последнего фильтра к точке res и выполнив процесс.





Row No.	Breed	weight_high...	weight_low...	Classification	obey_new	reps_lower	reps_upper
1	Saint Bernard	190	110	Fair Working/...	0.300	41	80
2	Great Dane	160	120	Average Wor...	0.500	26	40
3	Newfoundland	150	100	Above Avera...	0.700	16	25
4	Irish Wolfhound	150	90	Average Wor...	0.500	26	40
5	Bullmastiff	130	100	Fair Working/...	0.300	41	80
6	Akita	120	80	Average Wor...	0.500	26	40
7	Kuvasz	120	70	Average Wor...	0.500	26	40
8	Great Pyrenees	120	95	Fair Working/...	0.300	41	80
9	Scottish Deer...	110	75	Average Wor...	0.500	26	40
10	Rottweiler	110	90	Brightest Dogs	0.950	1	4
11	Bernese Mou...	110	85	Excellent Wor...	0.850	5	15
12	Doberman Pi...	100	60	Brightest Dogs	0.950	1	4
13	Rhodesian R...	85	70	Average Wor...	0.500	26	40
14	Weimaraner	85	70	Excellent Wor...	0.850	5	15
15	English Setter	80	45	Above Avera...	0.700	16	25
16	Gordon Setter	80	45	Above Avera...	0.700	16	25
17	Curly Coated...	80	65	Average Wor...	0.500	26	40
18	Labrador Ret...	80	55	Brightest Dogs	0.950	1	4
19	German Shor...	80	50	Excellent Wor...	0.850	5	15
20	Briard	76	74	Above Avera...	0.700	16	25
21	Chesapeake ...	75	55	Above Avera...	0.700	16	25

3.28. А теперь давайте спрячем все, чтобы никто это больше никогда не видел.

3.29. Выделите все блоки и, кликнув правой кнопкой мыши, выберите Move into new subprocess. Вы создали подпроцесс.

3.30. Переименуйте подпроцесс в «Подготовка данных».

3.31. Приступим к постройке матрицы корреляций (наконец-то!). Хотя нет, нужно еще немного обработать наши данные перед построением матрицы.

3.32. В таблице присутствуют атрибуты weight_low_lbs и weight_high_lbs. Это нижняя и верхняя границы веса, который может иметь собака конкретной породы. Также есть атрибуты reps_upper и reps_lower. Это верхняя и нижняя граница количества повторений, необходимых для того, чтобы собака понимала новую команды. Зачем нам иметь верхнюю и нижнюю границу, если в принципе нам хватит и средних значений? Исправим это.

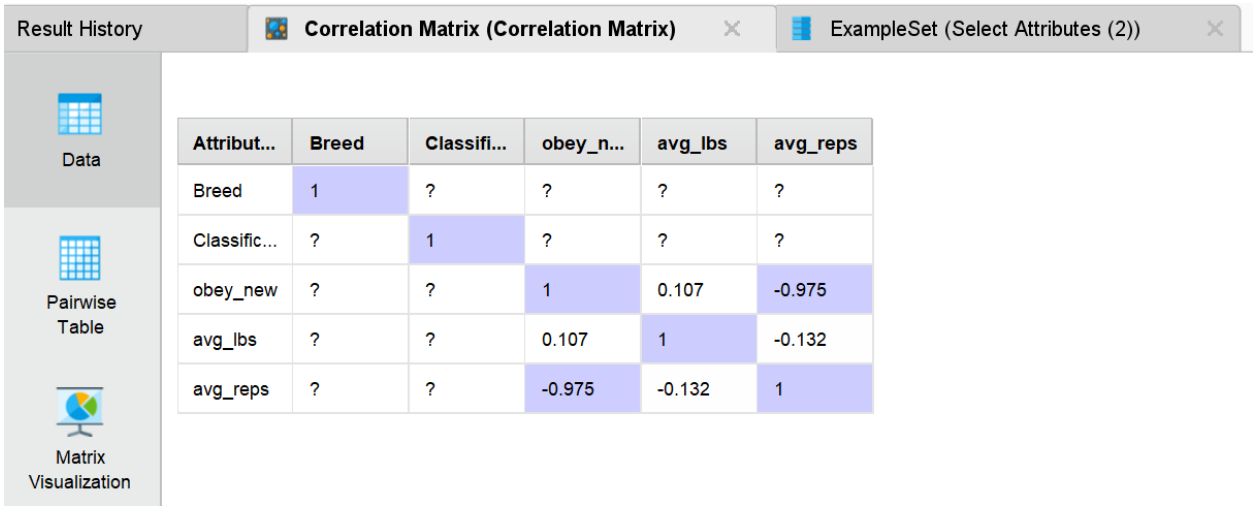
3.33. С помощью оператора Generate Attributes создадим два новых поля avg_lbs и avg_reps. В качестве выражения для вычисления новых полей будем использовать среднее из двух наших границ – avg(<граница1>,<граница2>). Сам оператор получает таблицу из выхода out нашего подпроцесса.

3.34. Ну и чтобы лишние столбцы не мозолили глаза, подключим далее оператор Select Attributes и оставим все атрибуты, кроме атрибутов наших граничных значений.

3.35. Посмотрите результат. Все получилось? Тогда идём дальше.

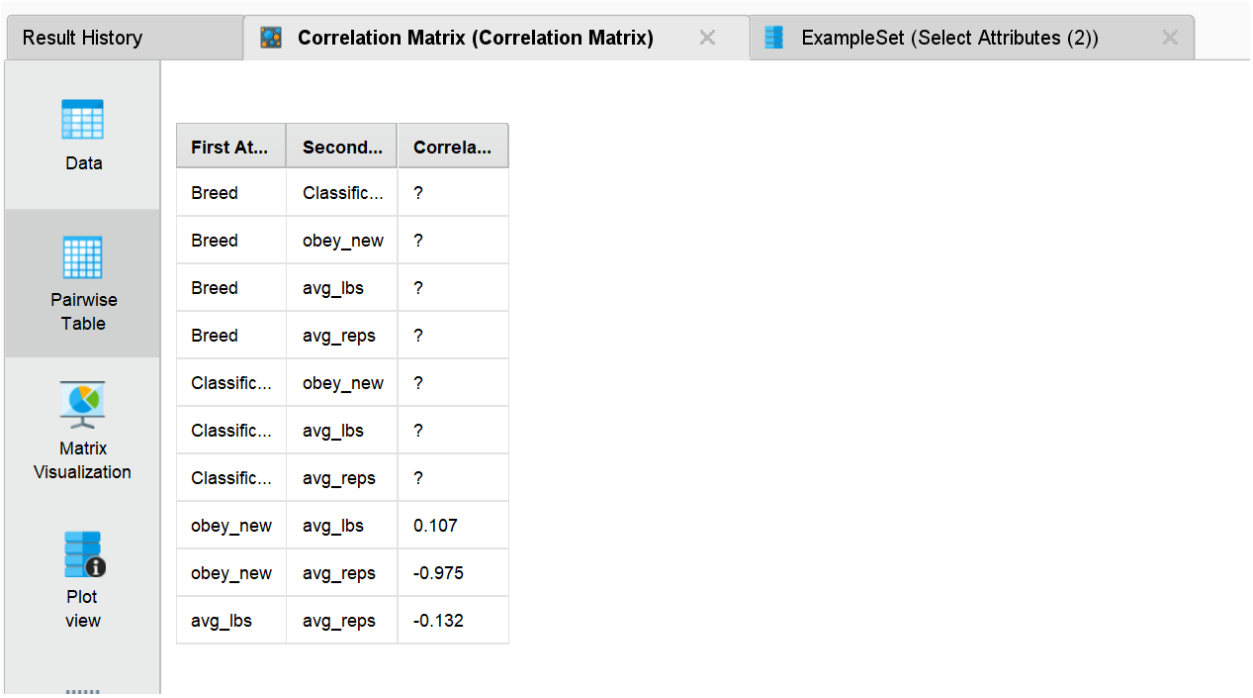
3.36. Подключите оператор Correlation Matrix. И соедините его выход mat и exa с точкой res.

3.37. Запустите процесс. Мы получили матрицу корреляций.



Attribut...	Breed	Classifi...	obey_n...	avg_lbs	avg_reps
Breed	1	?	?	?	?
Classific...	?	1	?	?	?
obey_new	?	?	1	0.107	-0.975
avg_lbs	?	?	0.107	1	-0.132
avg_reps	?	?	-0.975	-0.132	1

3.38. Для наглядности воспользуемся пунктом Pairwise Table.



First At...	Second...	Correla...
Breed	Classific...	?
Breed	obey_new	?
Breed	avg_lbs	?
Breed	avg_reps	?
Classific...	obey_new	?
Classific...	avg_lbs	?
Classific...	avg_reps	?
obey_new	avg_lbs	0.107
obey_new	avg_reps	-0.975
avg_lbs	avg_reps	-0.132

3.39. Между послушностью и весом положительная корреляция. Чем больше одно, тем больше другое. Между весом и кол-вом повторений – отрицательная.

3.40. И присутствует также очевидная отрицательная корреляция между количеством повторений и послушностью собаки. Чем меньше повторений требуется собаке для заучивания упражнения, тем она в общем является послушнее.

3.41. Взглянем на визуализацию нашего ExampleSet – набора, данных, который использовался для создания матрицы. Его можно получить как раз подключением выхода exa из блока матрицы корреляций.

3.42. Взглянем на график разброса Scatter. По оси X – средний вес. По оси Y – показатель послушности.

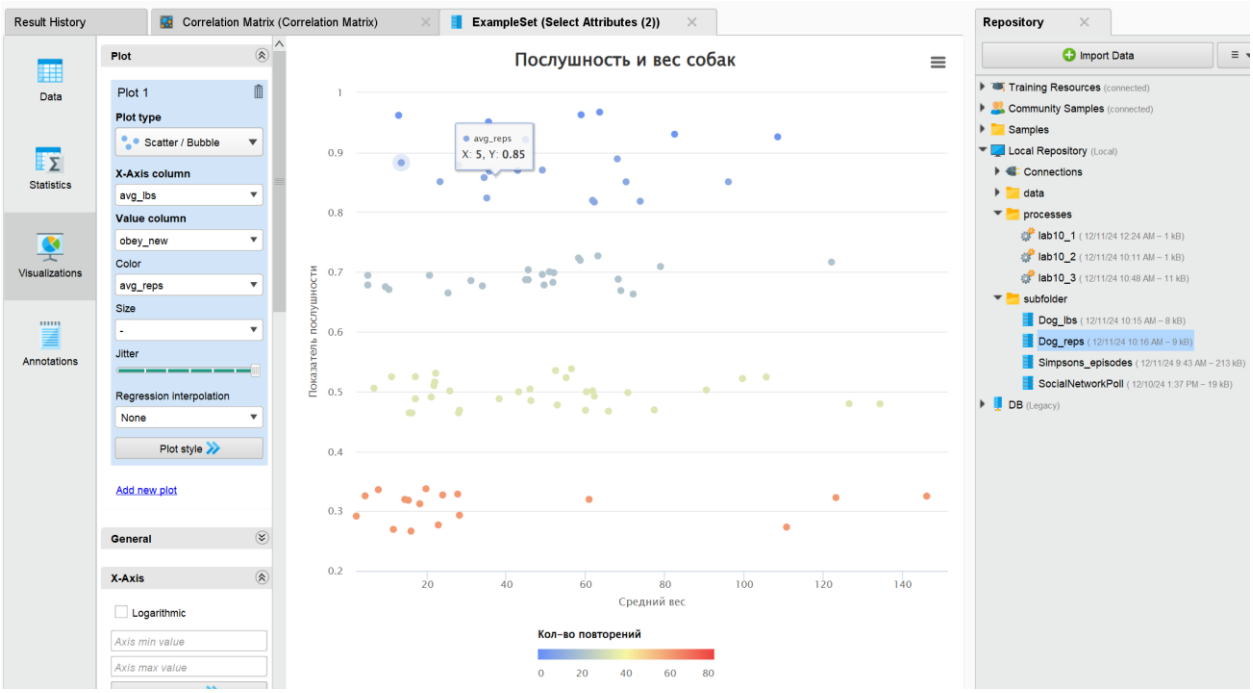
3.43. Увеличим параметр Jitter для повышения реалистичности данных.

3.44. Цветом color обозначим среднее количество повторений.

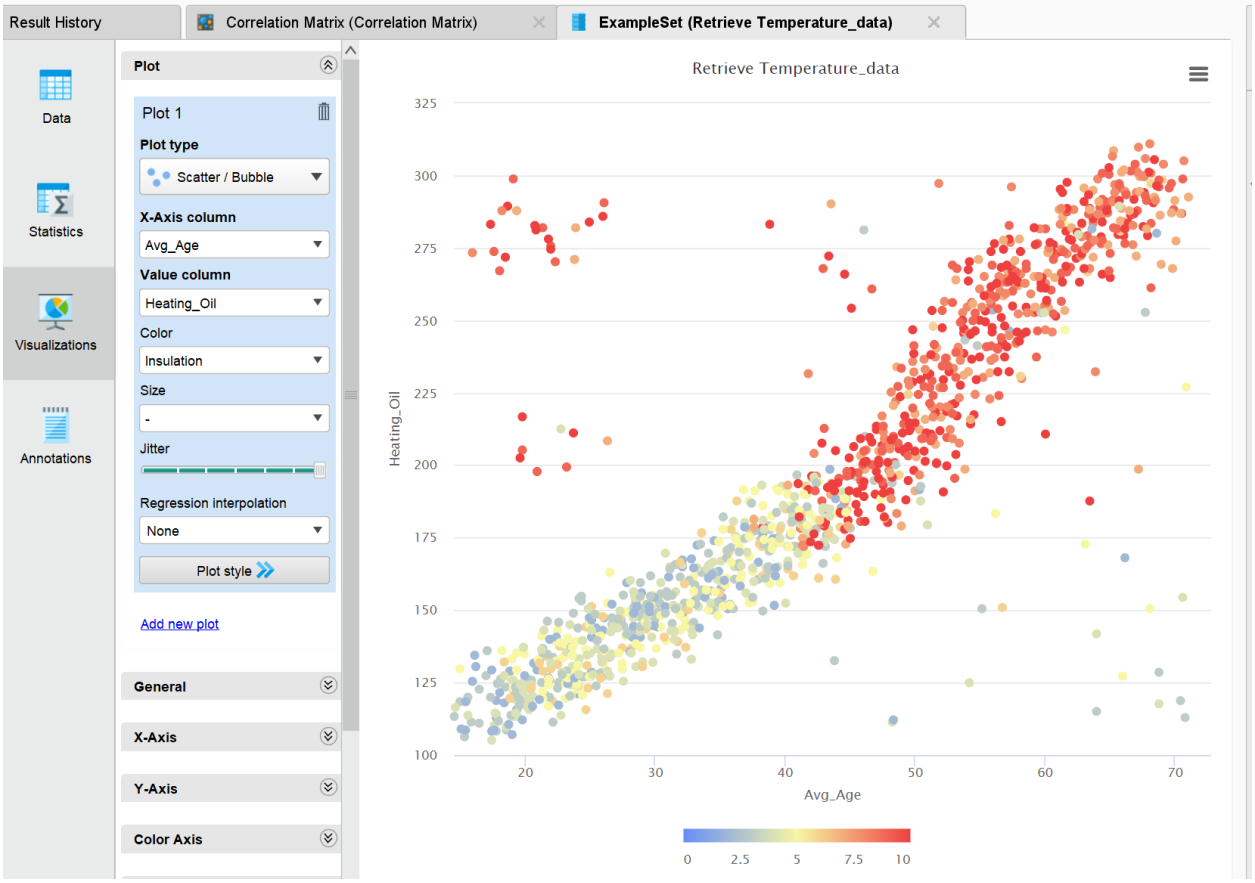
3.45. Добавим легенду для количества повторений.

3.46. На графике видна однозначная связь кол-ва повторений и послушности собак. Корреляция послушности и веса здесь не так заметна. Видно, что

присутствует некоторое «нормальное распределение». Размер имеет значение? Вообще - имеет. Но очень большая и очень маленькая собака будет слушаться одинаково плохо.

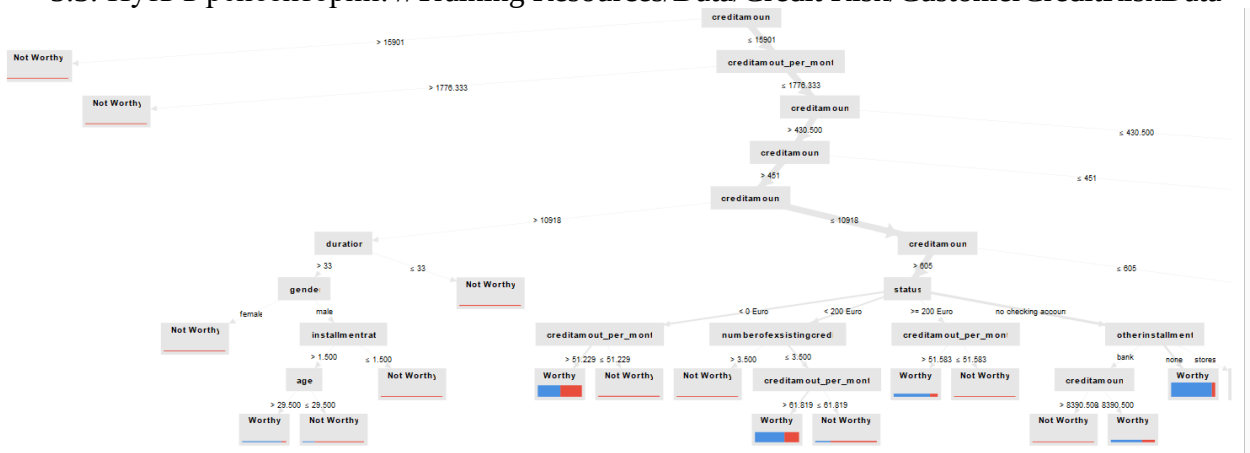


4. Задание: корреляция параметров домов.
   4.1. Постройте матрицу корреляций для набора данных temperature_data.csv.
   4.2. Сделайте выводы.
   4.3. Визуализируйте данные на диаграмме разброса.

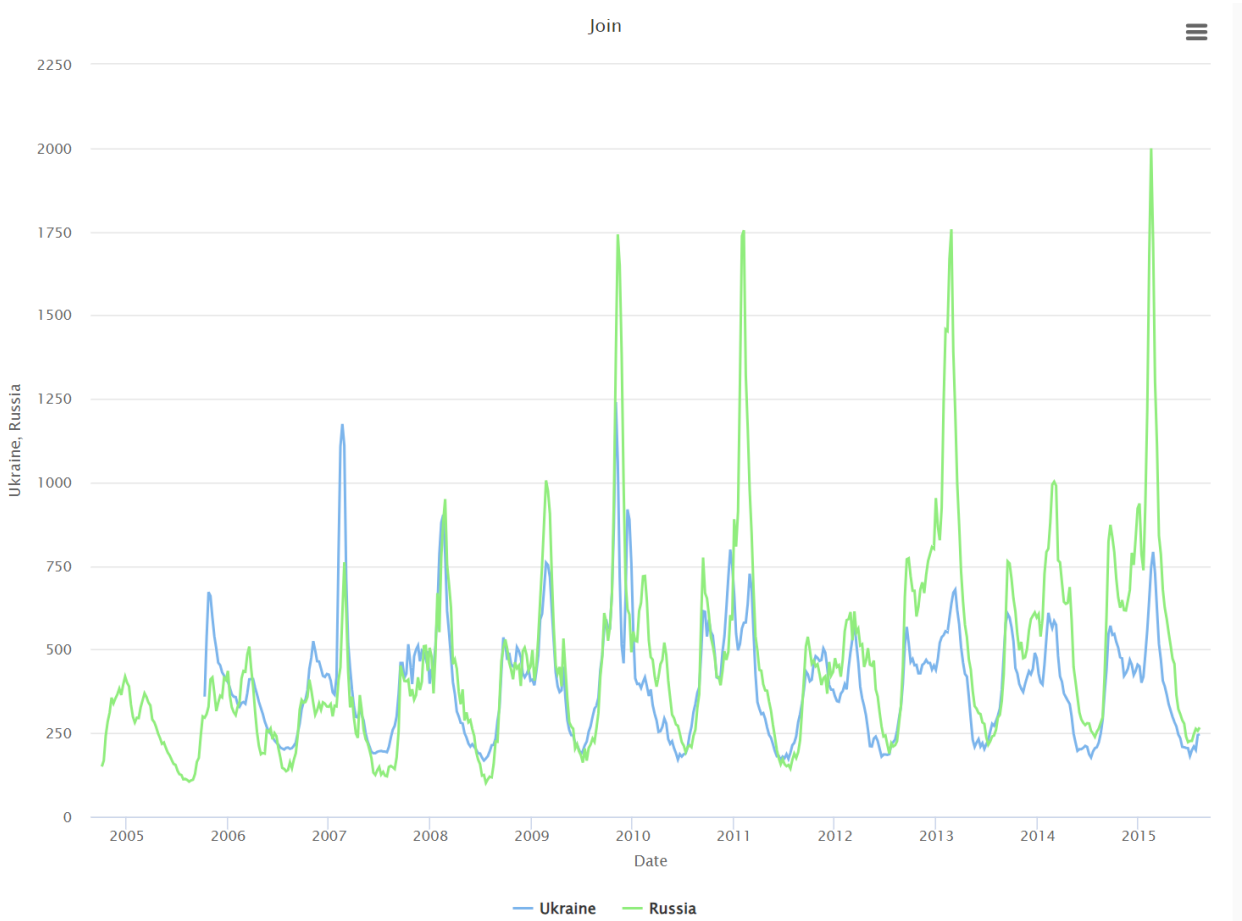


5. Дерево решений
   5.1. Используя встроенный набор данных по кредитам и навыки пользования программой, постройте дерево решений.

5.2. Сделайте выводы.
5.3. Путь в репозитории: //Training Resources/Data/Credit Risk/CustomerCreditRiskData



6. Машинное обучение в RapidMiner
   6.1. Для начала загрузим данные о количестве украинских поисковых запросов в Google, связанных с простудой (Ukraine-requests.txt)
   6.2. Данные представляют собой количество запросов на конец недели с 2005 по 2015 год. При импортировании данных необходимо задать формат даты для корректного построения временных графиков. Все остальные параметры оставляем по умолчанию.
   6.3. На вкладке Design перетаскиваем данные Ukraine-requests из Repository на рабочий стол RapidMiner. Соединим выход блока данных с точкой вывода результатов процесса (res).
   6.4. При нажатии кнопки «старт» программа покажет общую статистику.
   6.5. Используя вкладку Visualizations, построим график распределения данных. График отражает очевидную периодичность заболеваемости простудой: первая волна начинается осенью, а пик мы можем наблюдать к февралю.
   6.6. Теперь возьмём данные для России и посмотрим сохранится ли в них такая же периодичность, совпадают ли вспышки с теми периодами, которые мы выделили в Украине.
   6.7. Для этого загружаем новые данные и объединяем их с загруженными ранее; объединение производим по полю Date с помощью оператора “Join”.
   6.8. На графике мы можем видеть, что цикличность сохраняется и пики заболеваемости практически совпадают.



6.9. Перейдем к построению модели, которая будет предсказывать количество заболевших в Украине. Прогнозировать будем значение ряда на следующую неделю на основании значений четырех предыдущих недель (примерно одного месяца). Мы используем нейронную сеть прямого распространения для прогнозирования временного ряда. Выбор нейронных сетей обоснован простотой подбора параметров модели и их дальнейшего использования. В отличии от моделей авторегрессии и скользящего среднего нейронные сети не требуют проведения корреляционного анализа временного ряда.

6.10. Для корректной работы оператора нейронной сети необходимо преобразовать изначальный временной ряд в формат обучающей выборки. Для этого мы использовали оператор Windowing из пакета расширений Series Extension.

6.11. Далее с помощью оператора “Select Attributes” мы убрали из выборки лишние поля (даты для значений 1—4).

6.12. Обучение нейронной сети с учителем предполагает наличие обучающей и тестовой выборки, поэтому с помощью оператора “Split Data” мы разделили ВР в пропорции 80 на 20.

6.13. Согласно документации оператора “Neural Net”, необходимо, чтобы столбец прогнозируемых значений в обучающей выборке имел название/роль “Label”, для чего был использован оператор “Set Role”.

6.14. Поскольку столбец “Дата прогноза” не участвует в прогнозировании, ему необходимо присвоить роль “Id”.

6.15. Второй выход оператора “Split Data” и выход “mod” оператора “Neural Net” соединяем с соответствующими входами “Apply Model”.

6.16. Оператор “Apply Model” подает на вход натренированной модели контрольную выборку и сопоставляет прогнозируемое и реальное значения.

6.17. Завершающий этап нашего процесса — оператор “Performance”, необходимый для определения погрешности результатов.

6.18. Прогнозируемому значению, полученному от “Apply model” с помощью “Set Role(2)”, была присвоена роль “Prediction”.

6.19. Необходимо построить схему процесса, позволяющего прогнозировать значения временного ряда, так как показано на рисунке ниже:

6.20. Мы можем увидеть результат прогнозирования. Как видите, график с предсказанными данными очень близок к реальным данным.