Исследование: Тинькофф сториз

майкрософт ексель 2008 взлом скачать рутрекер

“Истории” в приложении Тинькофф - это короткий контент, показывающийся вверху главной страницы
приложения. Истории могут содержать персонализированную статистику, новости от банка,
интересные факты, ссылки на интересные ресурсы и многое другое. Анализ активности пользователей
в историях поможет банку устраивать публикации историй так, чтобы максимизировать вовлеченность
клиентов.

Исследовательский вопрос: *зависит ли активность пользователей в историях от времени суток?*

Гипотезы:
1. Люди, которые смотрят истории ночью (00:00-05:00), с большей вероятностью с ней провзаимодействуют
Механизм: если человек заходит в приложение ночью, то он заходит туда именно ради контента в приложении, а не для перевода денег или покупок.

2. Вечером (20:00-00:00) пользователи больше взаимодействуют с историями
Механизм: вечером у людей больше свободного времени, а значит, больше времени на активность в приложении

Первичная оценка данных

In[1]:=
```
data = Import["/home/flexagoon/Downloads/Тинькофф Сторис(1) (1)/data.csv",
   "Dataset", "HeaderLines" → 1];
```

Структура данных

Таблица содержит данные о просмотрах историй в приложении Тинькофф в период с 16 по 22 октября 2022 года.

Одна строка таблицы обозначает один просмотр, и содержит следующие данные:

- Айди пользователя
- Айди и название истории
- Дату, время и день недели просмотра в часовом поясе пользователя
- Данные об устройстве (тип, ОС, браузер
- Пол, возраст и регион пользователя
- Флаги для разных видов активности (лайк, пересылка, сохранение в избранное, нажатие на кнопку в истории)

Ошибки в базе данных

Во время первичной оценки было выявлено несколько ошибок и неточностей.

Дни недели

Одной и той же дате в базе данных может соответствовать несколько разных дней недели. Мы не смогли выяснить, с чем связана такая ошибка, но решили игнорировать переменную day_of_week, и обращать внимание только на дату просмотра.

In[412]:=
```
data[GroupBy["date_"], Counts, "day_of_week"][1 ;; 3]
```

Out[412]=

2022–10–16	6	21 636
	7	99 213
	5	6115
	1	1972
	4	630
	8 total ›	
2022–10–20	4	112 599
	2	7668
	3	24 197
		87
	1	594
	8 total ›	
2022–10–18	2	38 865
	7	2526
	1	6971
	3	286
	4	53
	8 total ›	

Флаг нажатия на кнопку

Параметр button_tap_flg является флагом, то есть должен принимать значения 0 или 1. Однако среди представленных данных есть строки, в которых данный флаг почему-то принимает значения выше 1.

In[5]:=
```
data[Counts, "button_tap_flg"] // KeySort
```

Out[5]=

0	558798
1	84383
2	2690
3	1
4	1
6	1

Таких строк мало, поэтому мы решили расценивать их как выбросы и поменять в них значение флага на 1.

In[22]:=
```
data = Normal[data];
data[[All, "button_tap_flg"]] = data[[All, "button_tap_flg"]] /. 2 | 3 | 4 | 5 | 6 → 1;
data = Dataset[data];
data[Counts, "button_tap_flg"] // KeySort
```

Out[25]=

0	558798
1	87076

Названия

У большинства историй в базе данных не указано название. В дальнейшем мы всё равно делим истории на категории по их названиям, однако, из-за отсутствия названия у большого количества историй, анализ по таким категориям не является полным.

In[414]:=
```
ReverseSort[data[Counts, "name"]][1 ;; 10]
```

Out[414]=

	264 650
Страхи в начале бизнеса	22 971
Частые траты	19 412
Ваши траты на продукты	16 292
ПДД для самокатов	12 791
Больше в ленте «Для вас»	9370
Ваши инвестиции за неделю	8921
Сделайте напиток из эспрессо	7079
Ваши траты на авто	6797
Топ–6 историй	6722

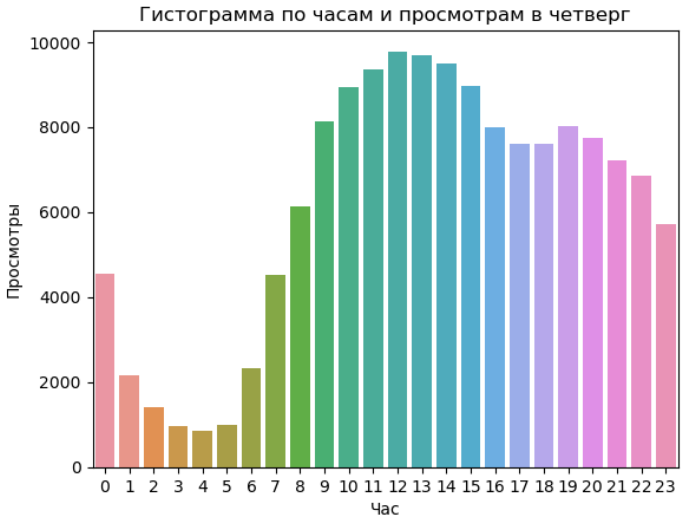
Айди историй

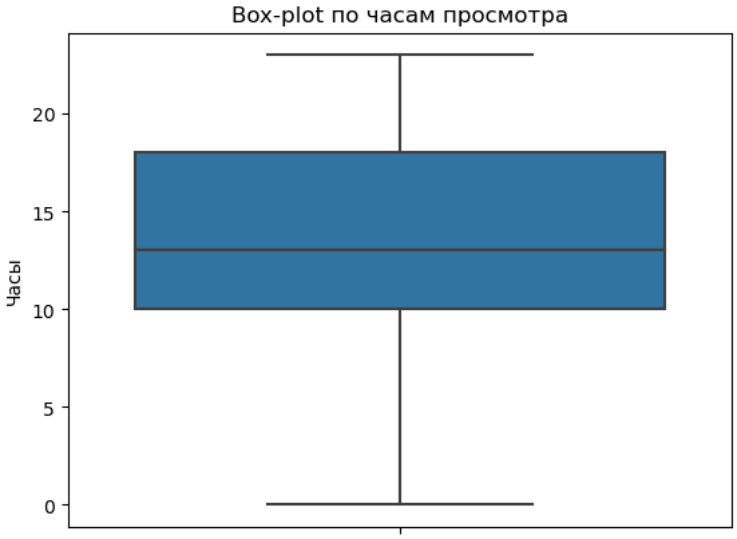
Из-за маленького количества данных о самих историях, мы решили попробовать получить недостающие данные с сайта Тинькофф. Однако выяснилось, что на сайте истории используют буквенные айди, тогда как в базе данных айди историй являются числами. В связи с этим мы не можем получить никакие данные об историях, не предоставленные нам в базе данных.

Предварительный анализ

Количество просмотров по часам

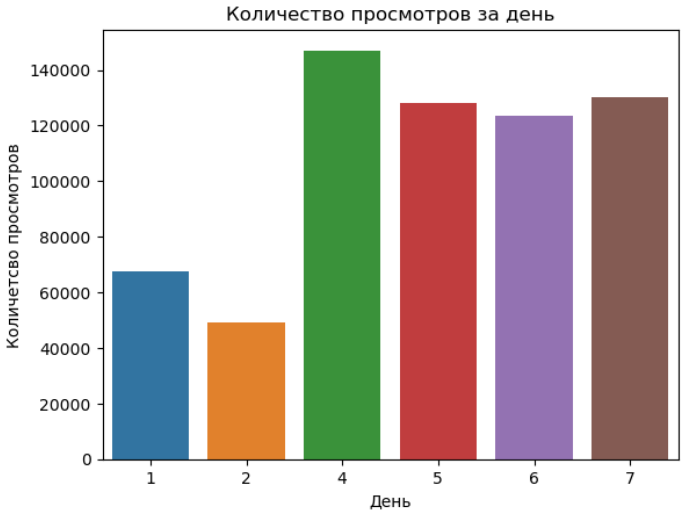
Анализ просмотров по часам показал ожидаемый результат - люди меньше смотрят истории ночью, когда они спят, а пик активности приходится на полдень.

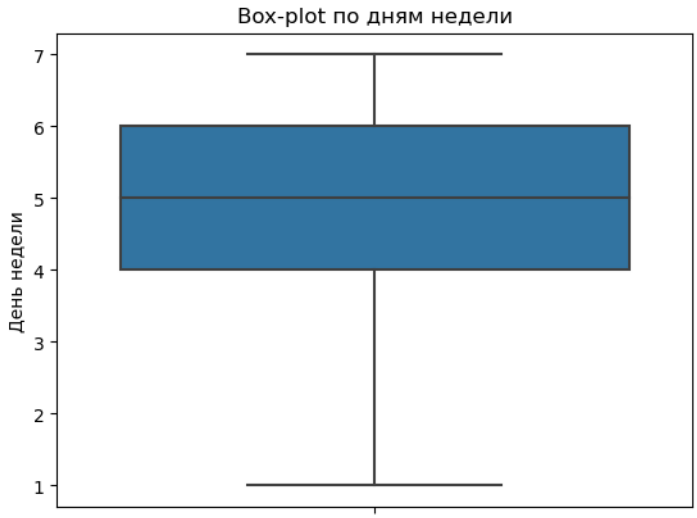




Количество просмотров по дням недели

Здесь результат получится менее очевидным - по какой то причине активность в понедельник и вторник сильно ниже, чем активность в остальные дни недели.





Мы не смогли выяснить точную причину, но, скорее всего, это просто обусловлено маленьким размером выборки и тем, что конкретно на этой неделе в понедельник и вторник было меньше интересных историй.

Основной анализ

Индекс активности

Для дого чтобы считать среднюю активность за какой-то срок, сначала нам нужно определить общую активность для каждого просмотра истории. Однако к каждому просмотру у нас есть 4 флага для разных видов активностей. Поэтому, для дальнейшего анализа, необходимо высчитать общий индекс активности для каждого просмотра.

Равнозначный индекс активности

Нахождение и анализ индекса

Для начала попробуем просто найти среднее арифметическое между всеми четыремя видами активности.

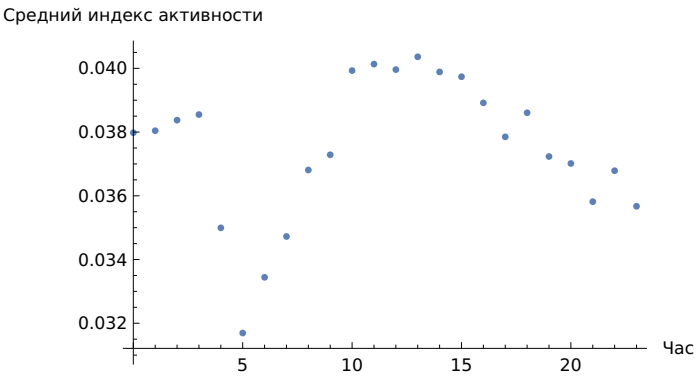
Добавим в датасет новую колонку act_index.

In[313]:=
```
dataAct = data[All, Append[#, "act_index" →
        (#"button_tap_flg" + #"like_tap_flg" + #"share_tap_flg" + #"favorite_tap_flg") / 4] &]
    ;
```

Строим график среднего индекса активности по часам:

In[416]:=
```
dataAct[GroupBy["user_hour"], Mean, "act_index"] //
 ListPlot[#, AxesLabel → {"Час", "Средний индекс активности"}] &
```

Out[416]=



Сразу видно, что данный график по форме напоминает график количества просмотров по часам.

Нужно проверить их корреляцию:

In[33]:=
```
Normal@Values@dataAct[Counts, "user_hour"]~
  Correlation ~
  Normal@Values@dataAct[GroupBy["user_hour"], Mean, "act_index"] // N
```

Out[33]=
```
0.712233
```

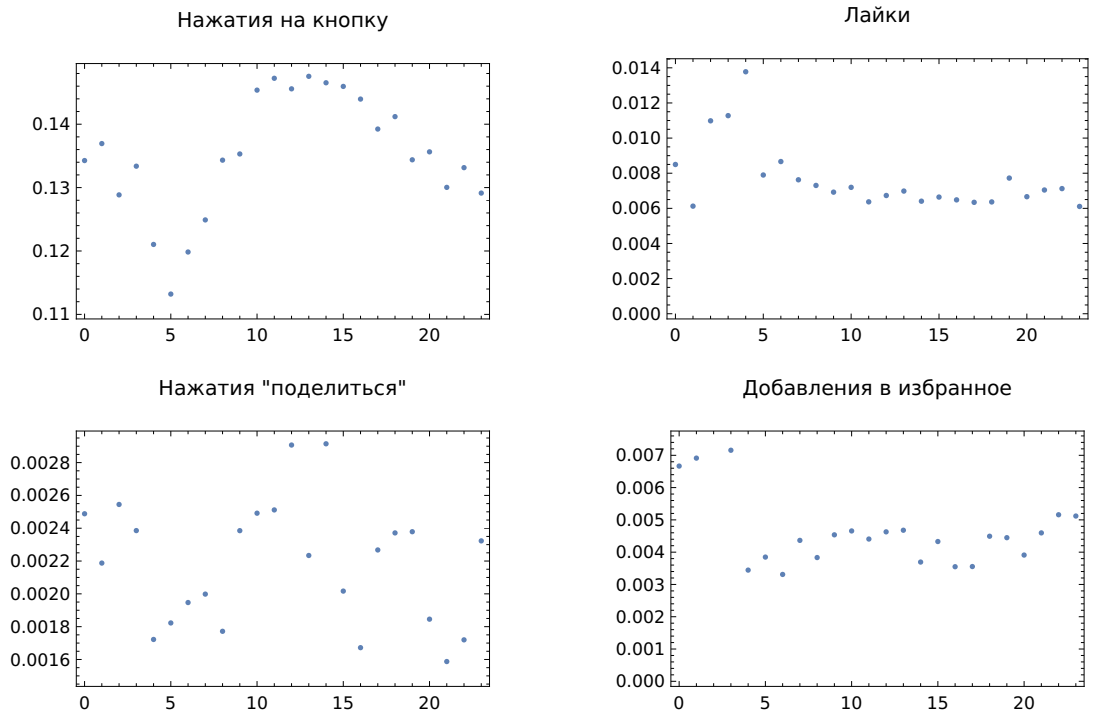
И действительно, корреляция равняется 0.71, то есть графики практически одинаковые, за исключением показателей ночью, которые в графике активности выше (относительно значений того же графика).

Анализ отдельных видов активности

Построем графики для каждого вида активности по отдельности, чтобы понять, похожи ли они все на график просмотров. Находим среднее значение каждого флага за каждый час.

In[417]:=
```
GraphicsGrid[{
  {dataAct[GroupBy["user_hour"], Mean, "button_tap_flg"] //
    ListPlot[#, PlotLabel → "Нажатия на кнопку", Frame → True] &,
   dataAct[GroupBy["user_hour"], Mean, "like_tap_flg"] //
    ListPlot[#, PlotLabel → "Лайки", Frame → True] &},
  {dataAct[GroupBy["user_hour"], Mean, "share_tap_flg"] //
    ListPlot[#, PlotLabel → "Нажатия \"поделиться\"", Frame → True] &,
   dataAct[GroupBy["user_hour"], Mean, "favorite_tap_flg"] //
    ListPlot[#, PlotLabel → "Добавления в избранное", Frame → True] &}
 }]
```

Out[417]=



Мы видим, что на график просмотров похож только график нажатий на кнопку, однако фактические значения на этом графике сильно выше, чем на остальных, поэтому и среднее между всеми видами активности будет похоже на него.

Тут надо уточнить, что это вообще за нажатия на кнопку.

Лирическое отступление про кнопку

Зачем банку нужны истории в приложении? У всех историй есть одна общая цель - добавить

интересный контент в приложение, чтобы привлекать туда больше людей и увеличивать их содержание.

Однако в некоторых историях, помимо самого содержания, в конце находится кнопка - она может вести, например, на страницу какой - то акции в приложении, или на статью в Тинькофф Журнале. У таких сториз есть еще одна цель: побудить как можно больше пользователей перейти по ссылке и совершить целевое действие.

Нажатия на кнопку часто являются самым полезным для банка видом активности в историях, поскольку они могут приносить прибыль напрямую, а не косвенно . То, что количество нажатий кнопки сильно коррелируют с количеством просмотров, означает, что какое - то количество пользователей стабильно нажимают на кнопку в историях. А значит SMM - щики Тинькофф молодцы и делают привлекающий людей контент, а разработчики Тинькофф молодцы и делают большие кнопки ☺

Однако в нашем исследовании мы измеряем уровень активности, а не прибыль, которую истории приносят банку (для такого исследования у нас просто недостаточно данных) . Поэтому такой график, в котором ничего, кроме нажатий на кнопку не имеет значения, нам не подходит .

Взвешенный индекс активности

Нахождение индекса

Раз обычное среднее арифметическое нам не подходит, необходимо использовать взвешенное среднее арифметическое. Мы хотим, чтобы все виды активности учитывались равномерно, вне зависимости от их общей популярности. Поэтому флаги активностей нужно домножить на значения, обратно пропорциональные популярности этих флагов.

```
(* Ищем общее количество каждого вида активности *)
buttonCoeff = data[Total, "button_tap_flg"];
likeCoeff = data[Total, "like_tap_flg"];
shareCoeff = data[Total, "share_tap_flg"];
favoriteCoeff = data[Total, "favorite_tap_flg"];

(* Составляем обратную пропорцию *)
buttonCoeff = 1/buttonCoeff;
likeCoeff = 1/likeCoeff;
shareCoeff = 1/shareCoeff;
favoriteCoeff = 1/favoriteCoeff;

(* Приводим значения так, чтобы в сумме они давали 1 *)
s = buttonCoeff + likeCoeff + shareCoeff + favoriteCoeff;
buttonCoeff /= s;
likeCoeff /= s ;
shareCoeff /= s;
favoriteCoeff /= s;

(* Выводим веса *)
Print["Вес нажатия кнопки: ", N[buttonCoeff]]
Print["Вес лайка: ", N[likeCoeff]]
Print["Вес отправки: ", N[shareCoeff]]
Print["Вес сохранения: ", N[favoriteCoeff]]
```

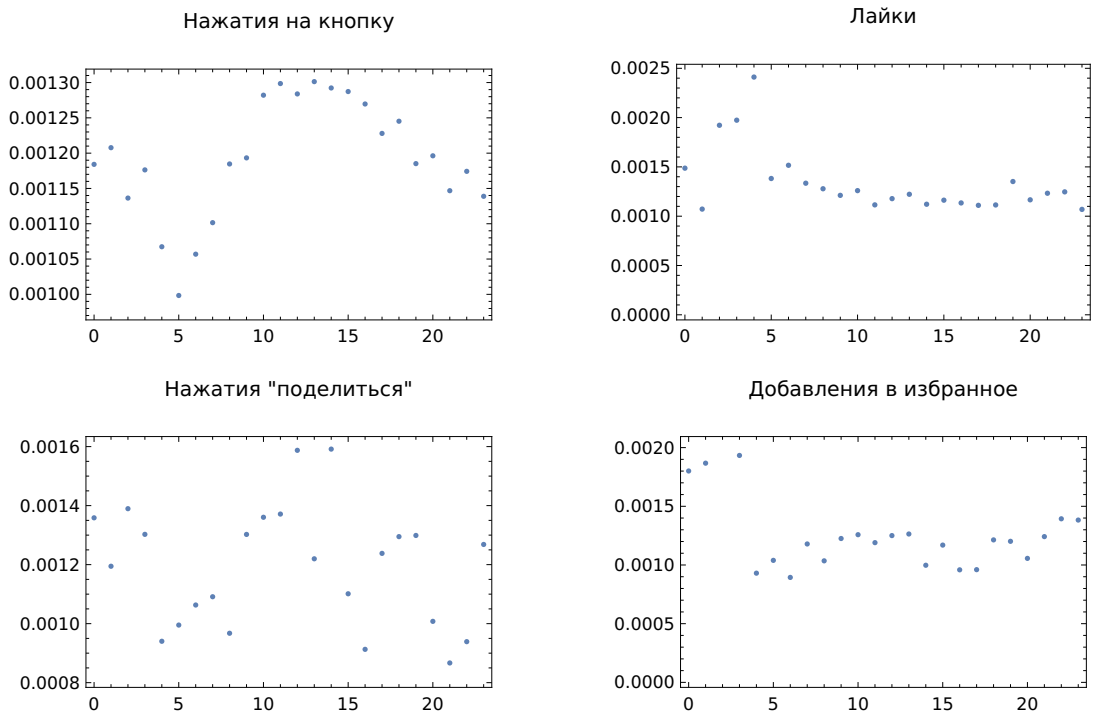
```
Вес нажатия кнопки: 0.00881922
Вес лайка: 0.175012
Вес отправки: 0.546038
Вес сохранения: 0.270131
```

Мы не можем объективно оценить сложность каждого вида активности, но по логическим предположениям такое распределение весов активностей как раз связано с их сложностью и необходимым уровнем вовлечения для пользователя: отправка истории другу - самое сложное действие, требующее максимального вовлечения, а нажатие кнопки - наоборот.

Смотрим, как меняются графики отдельных видов активностей после домножения на веса.

In[419]:=
```
GraphicsGrid[{
  {data[All, Append[#, "w_button" → buttonCoeff * #"button_tap_flg"] &][
     GroupBy["user_hour"], Mean, "w_button"] //
    ListPlot[#, PlotLabel → "Нажатия на кнопку", Frame → True] &,
   data[All, Append[#, "w_like" → likeCoeff * #"like_tap_flg"] &][
     GroupBy["user_hour"], Mean, "w_like"] //
    ListPlot[#, PlotLabel → "Лайки", Frame → True] &},
  {data[All, Append[#, "w_share" → shareCoeff * #"share_tap_flg"] &][
     GroupBy["user_hour"], Mean, "w_share"] //
    ListPlot[#, PlotLabel → "Нажатия \"поделиться\"", Frame → True] &,
   data[All, Append[#, "w_favorite" → favoriteCoeff * #"favorite_tap_flg"] &][
     GroupBy["user_hour"], Mean, "w_favorite"] //
    ListPlot[#, PlotLabel → "Добавления в избранное", Frame → True] &}
 }]
```

Out[419]=



Теперь область значения для каждого графика одинаковая, а значит, все активности будут равномерно влиять на индекс активности.

Добавляем в таблицу параметр act_index, который вычисляется как взвешенное среднее арифметическое четырех флагов активности с подсчитанными весами.

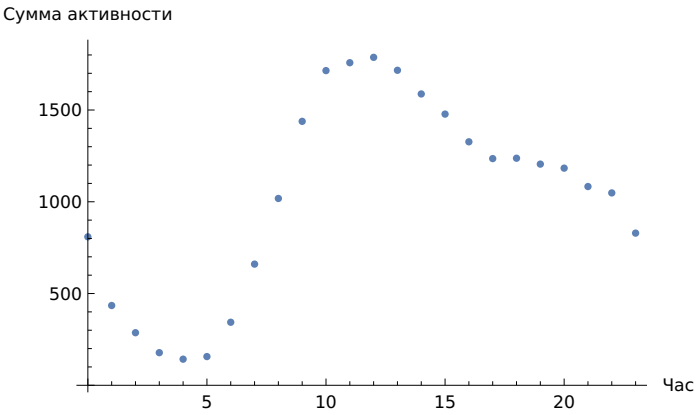
In[21]:=
```
dataAct = data[All, Append[#,
    "act_index" → buttonCoeff * #"button_tap_flg" + likeCoeff * #"like_tap_flg" +
      shareCoeff * #"share_tap_flg" + favoriteCoeff * #"favorite_tap_flg"] &];
```

Построение графика

Сначала пробуем построить график абсолютной активности - то есть суммы индексов за каждый час.

In[422]:=
```
dataAct[GroupBy["user_hour"], Total, "act_index"] //
 ListPlot[#, AxesLabel → {"Час", "Сумма активности"}] &
```

Out[422]=



Видно, что этот график практически совпадает с графиком просмотров. Можно построить их корреляцию.

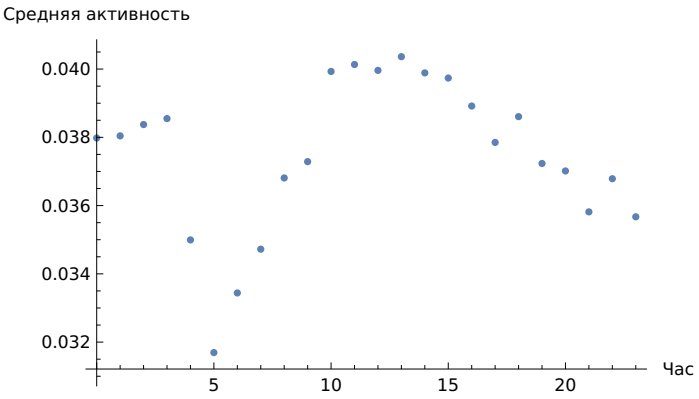
In[23]:=
```
Normal@Values@dataAct[Counts, "user_hour"]~
  Correlation ~
  Normal@Values@dataAct[GroupBy["user_hour"], Total, "act_index"] // N
```

Out[23]=
```
0.989003
```

Это правда практически один и тот же график. Поэтому мы делаем график относительной активности - то есть среднего индекса активности за каждый час.

In[423]:=
```
dataAct[GroupBy["user_hour"], Mean, "act_index"] //
 ListPlot[#, AxesLabel → {"Час", "Средняя активность"}] &
```

Out[423]=



Ура, у нас наконец то получился нормальный график.

Выводы

По графику видно, что пик активности как раз приходится на ночь, в период с 0 до 5 часов. Следовательно, наша первая гипотеза подтвердилась. А вот вторая гипотеза оказалась неверной - даже если убрать с графика ночь, уровень активности в целом падает в течении дня, а его пик происходит в полдень.

Большой всплеск относительной активности связан с высоким относительным количеством лайков и добавлений в “избранное” в это же время. Повышение активности около полудня происходит из за повышения в это время относительного количества нажатий на кнопку и на “поделиться”.

Доступных данных недостаточно, чтобы точно объяснить причину такого распределения, но, скорее всего, люди просто чаще заходят в приложение в середине дня, когда совершают денежные переводы или покупки. Ночная активность обусловлена тем, что ночью люди заходят в приложение не ради перевода денег, а именно для просмотра контента в приложении, поэтому взаимодействия с контентом будет больше.

Дополнительный анализ

Самые популярные истории

Рассмотрим, какие именно истории набирали наибольшую популярность. Для этого посчитаем суммарный индекс активности для каждой истории.

In[358]:=
```
ReverseSort[dataAct[GroupBy["story_id"], Total, "act_index" , N]][[1 ;; 5]]
```

Out[358]=

20285	6431.75
20276	5321.5
20458	1508.25
17281	1046.75
20055	377.25

К сожалению, для 5 самых популярных историй в нашей базе данных не указано название. Поэтому посмотрим общую активность только для тех историй, где оно указано.

In[359]:=
```
ReverseSort[
  dataAct[Select[#"name" ≠ "" &]][GroupBy["story_id"], Total, "act_index" , N]][[1 ;; 5]]
```

Out[359]=

20458	1508.25
12739	312.5
20554	268.25
10788	229.0
18751	223.75

In[363]:=
```
data[Select[#"story_id" == 20458 &]][1, "name"]
data[Select[#"story_id" == 12739 &]][1, "name"]
data[Select[#"story_id" == 20554 &]][1, "name"]
data[Select[#"story_id" == 10788 &]][1, "name"]
data[Select[#"story_id" == 18751 &]][1, "name"]
```

Out[363]=
```
Сделайте  напиток  из эспрессо
```

Out[364]=
```
Ваши траты на продукты
```

Out[365]=
```
1000   за друга
```

Out[366]=
```
Топ-6 историй
```

Out[367]=
```
Ваши  инвестиции  за неделю
```

Нельзя сказать, что все популярные истории принадлежат к какой-то одной категории, но видно, что все эти названия привлекают внимание, а еще во всех этих историях скорее всего есть кнопка.

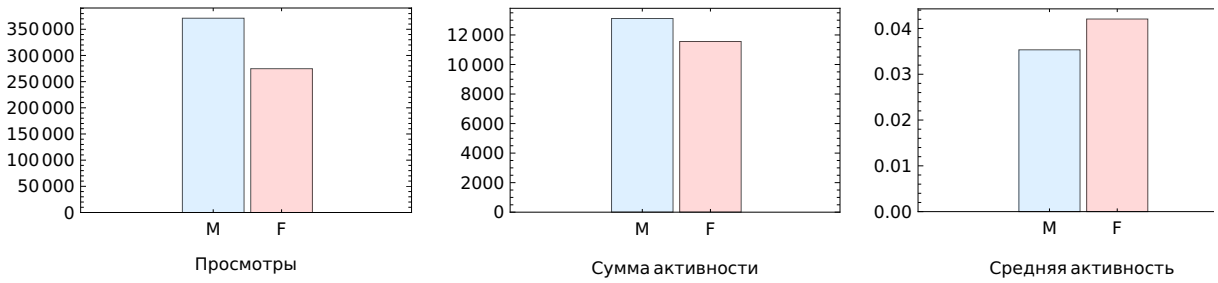
Анализ по полу

Сравним количество просмотров, сумму активности и среднюю активность для мужчин и женщин.

In[401]:=
```
genderChart[data_, label_] := BarChart[data, "ChartLabels" → {"M", "F"},
  "ChartStyle" → {LightBlue, LightRed}, "Frame" → True, "FrameLabel" → label]

GraphicsRow[{
  genderChart[data[Counts, "gender"], "Просмотры"],
  genderChart[
   dataAct[GroupBy["gender"], Total, "act_index"], "Сумма активности"],
  genderChart[dataAct[GroupBy["gender"], Mean, "act_index"], "Средняя активность"]
 }]
```

Out[402]=



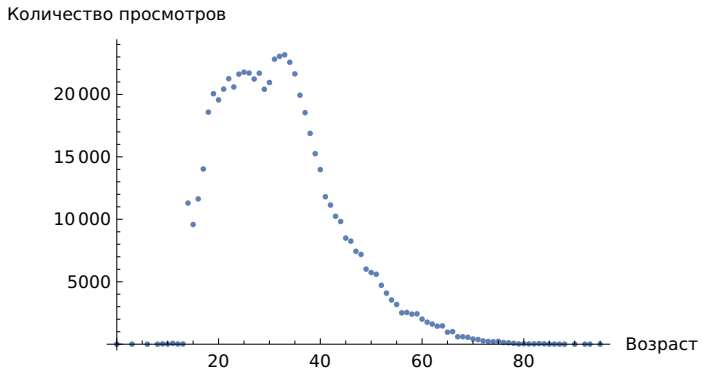
Мужчины больше смотрят истории, в следствии чего у них выше сумма индексов активности, однако средний индекс активности выше у женщин. Разница в обоих случаях недостаточно значительная, чтобы как-то влиять на данные.

Анализ по возрасту

Построим график просмотров по возрастам:

In[424]:=
```
ListPlot[data[Counts, "age"], AxesLabel → {"Возраст", "Количество просмотров"}]
```

Out[424]=



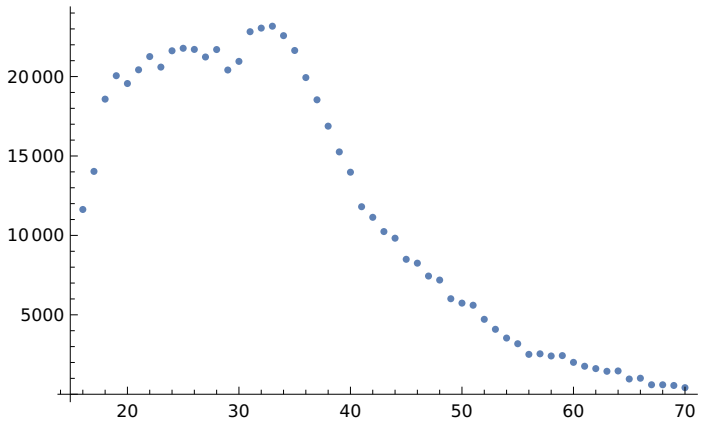
Видно, что людей младше 16 лет и старше 70 лет практически нет, поэтому уберем такие записи из выборки, и будем смотреть только на просмотры людей от 16 до 70 лет.

In[404]:=
```
dataActA = dataAct[Select[16 ≤ #"age" ≤ 70 &]];
```

Снова построим график просмотров:

```
ListPlot[dataActA[Counts, "age"],
 AxesLabel → {"Возраст", "Количество просмотров"}]
```

Out[406]=



А также график средней активности по возрастам:

In[425]:=
```
dataActA[GroupBy["age"], Mean, "act_index"] //
 ListPlot[#, AxesLabel → {"Возраст", "Средняя активность"}] &
```

Out[425]=

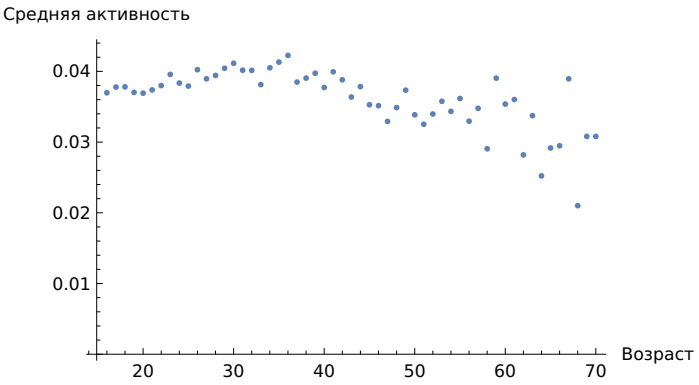


График суммы активностей будет такой же, как график просмотров.

Пик активности (как средней, так и суммарной) находится в возрасте ±35 лет, после чего активность падает, т.к. чем старше человек, тем меньше он интересуется контентом в историях. В целом высокая активность у людей с 20 до 35 лет, потому что люди в этом возрасте, как правило, работают, а значит, часто пользуются приложением банка.