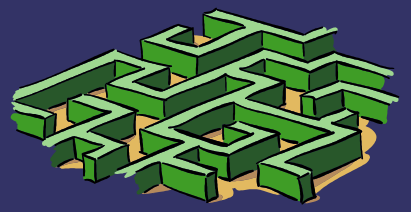
Project 1
Working with Hive on Wikipedia Datasets

Querying Data from English Wikipedia with Hive



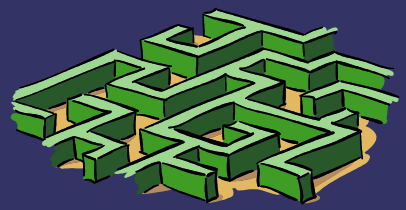
Q1:
Which English Wikipedia article got the most traffic on October 20?

Downloaded files using wget.

Create a table for our page views.

Aggregate the Sum of the count_views of all the views from the 24 hours of October 20th. Limiting the domain code to English Wikipedia but including both mobile, website.

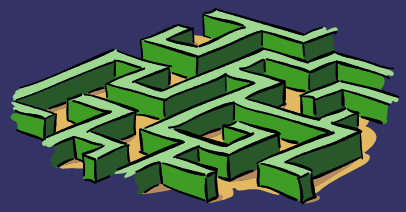
```
wget https://dumps.wikimedia.org/other/pageviews/2020/2020-10/
pageviews-20201020-{00..23}0000.gz

CREATE EXTERNAL TABLE PAGEVIEW
    (DOMAIN_CODE STRING,
    PAGE_TITLE STRING,
    COUNT_VIEWS INT,
    TOTAL_RESPONSE_SIZE INT)
    ROW FORMAT DELIMITED
    FIELDS TERMINATED BY ' '
    LOCATION '/user/tmickle/project1/oct20/oct20';

    CREATE TABLE PAGEVIEW_COUNTS
    AS SELECT PAGE_TITLE, SUM(COUNT_VIEWS) AS total_views FROM PAGEVIEW
    WHERE DOMAIN_CODE LIKE "en%"
    GROUP BY PAGE_TITLE;

--- Top 10 total views for Oct 20 --
    select * from PAGEVIEW_COUNTS
    order by total_views desc
    limit 10;
```



Q1:
Which English wikipedia article got the most traffic on October 20?

Our Results:

page_title	total_views
Main_Page	5993199
Special:Search	1567851
-	556527
Jeffrey_Toobin	321459
C._Rajagopalachari	211147
The_Haunting_of_Bly_Manor	185139
Robert_Redford	178779
Jeff_Bridges	159163
Bible	151535
Chicago_Seven	149966



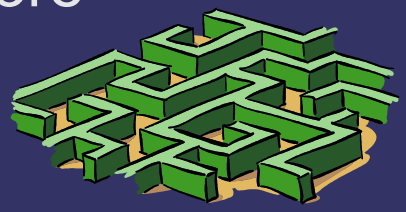
Q2.
What English Wikipedia article has the largest fraction of its readers follow an internal link to another Wikipedia article?

- Side notes on this solution:
-

-- We are answering **only for September** because of my limited HD space and computing power on a standalone machine. Ideally we would work on longer timescales.

– I'll present two solutions; one produces some outlier anomalies where a page is internally linked more than it is actually viewed.

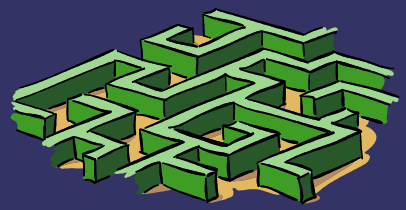
– The other method ensures clean results in our ratios but isn't seemingly as accurate, if we trust the Pageview data more than the Clickstream data.



First approach to question 2:

- In this solution we are joining a table of total page views for September with the click stream totals of internal link referrers on page titles. We are then checking out of the total views how many users followed a link out of the current page.

```
--  First possible solution (we get many outliners though, with 4x
the followed links to views of pages)
SELECT PAGE_TITLE, total_views, links_follwed, ((links_follwed/
total_views) * 100) AS precentage_followed_link
  FROM TOTAL_PAGEVIEW_COUNTS_SEPT
  inner join clickstream_links_followed
  on prev = PAGE_TITLE
  order by precentage_followed_link desc
  limit 20;
```

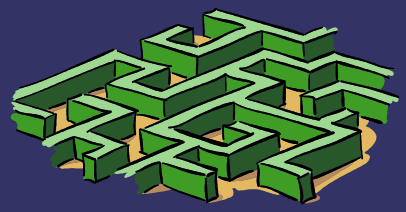


page_title	total_views	links_follwed	precentage_followed_link
/r/	1	64	6400.0
/\	2	56	2800.0
Health//Disco	8	209	2612.5
Strange_haircuts_//_cardboard_guitars_//_and_computer_samples	1	26	2600.0
List_of_listed_buildings_in_Musselburgh,_East_Lothian	28	662	2364.285714285714
Flourish_//_Perish	1	19	1900.0
Lost_Forever_//_Lost_Together	29	463	1596.551724137931
2006_Chicago_Rush_season	12	185	1541.6666666666665
Baeolidia_gracilis	8	121	1512.5
Finally_//_Beautiful_Stranger	19	282	1484.2105263157896
/2016Album/	33	471	1427.2727272727273
/pol/	490	6927	1413.6734693877552
Whole_New_Thing_(disambiguation)	1	13	1300.0
Deutsch-Französische_Gymnasium	43	540	1255.813953488372
/dev/full	8	92	1150.0
.hack//Sign	148	1665	1125.0
Thirteen_Songs	2	22	1100.0
De_Bellaigue	1	11	1100.0
.hack//Link	21	224	1066.6666666666665
/boot/	12	124	1033.3333333333335

Q2: Take Two

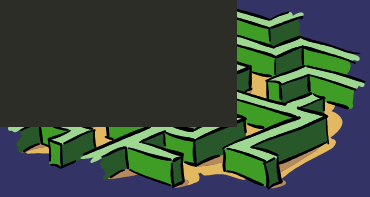
- Because of the issue with the Pageview aggregation, I stuck with using only the Clickstream data as our single source of truth for this question.

- I created a new total for each page for the Month of September by using just the Clickstream data (I left out 'other' because they are either searches to another page / spoofed)
- What we are leveraging is that every referring link is a guaranteed view + every page that's reached from an external link we know is a view.



Q2: Results

```
--- Because of the outliners of the previous approach; We stick to using
one set of data as our source of measurement. (Clickstream data)
--- Table for total views from the clickstream data by adding external
links + interal links (We leave out 'other' because they are for searches
to a page / spoofed )
create table clickstream_total_views AS
select curr as title, (external_links + links_follwed) as total from
CLICKSTREAM_EXTERNAL_LINKS
inner join clickstream_links_followed
on prev = curr
order by total desc;

-- Our final solution Query.
-- We no longer get over 100% results; though out total views from this
approach no longer match the total views total from the a
TOTAL_PAGEVIEW_COUNTS_SEPT.
   create table links_followed_ratio AS SELECT title, total,
   links_follwed, ((links_follwed/ total) * 100) AS
   precentage_followed_link
    FROM clickstream_total_views
    inner join clickstream_links_followed
    on title = prev
    order by precentage_followed_link desc
    limit 25;
```



No constraints except order

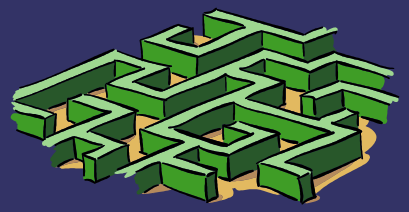
links_followed_ratio.title	links_followed_ratio.total	links_followed_ratio.links_follwed	links_followed_ratio.precentage_followed_link
January_1907_Russian_legislative_election	937	926	98.82604055496265
Oyiradai	1147	1131	98.6050566695728
1967_Japanese_general_election	887	874	98.53438556933483
1988_United_States_Senate_election_in_West_Virginia	626	616	98.40255591054313
2012_Delaware_gubernatorial_election	839	825	98.33134684147795
October_1907_Russian_legislative_election	764	750	98.1675392670157
Jimmy_Dunne_(disambiguation)	628	616	98.08917197452229
1998_United_States_Senate_election_in_Wisconsin	834	818	98.08153477218225
November_1989_Greek_legislative_election	559	548	98.03220035778175
Yakuza_(disambiguation)	1521	1491	98.0276134122288
2002_Montenegrin_presidential_election	506	496	98.02371541501977
1954_California_gubernatorial_election	1133	1110	97.96999117387467
Bridei_VI	532	521	97.93233082706767
News_of_the_World_(disambiguation)	577	565	97.92027729636048
1992_United_States_Senate_election_in_Iowa	1097	1074	97.90337283500456
Prosomapoda	761	745	97.89750328515112
Al-Mu'tadid_I	517	506	97.87234042553192
IUCN_Red_List_of_extinct_species	26961	26387	97.87099885019101
1998_United_States_Senate_election_in_Iowa	1123	1099	97.86286731967942
Abbas_ibn_Shith	888	869	97.86036036036036
Baldur's_Gate_(disambiguation)	828	810	97.82608695652173
1969_Japanese_general_election	825	807	97.81818181818181
1979_Japanese_general_election	1044	1021	97.79693486590038
1969_Turkish_general_election	498	487	97.79116465863453
2012_Vermont_gubernatorial_election	1395	1364	97.77777777777777

With Links Followed floor of 200,000

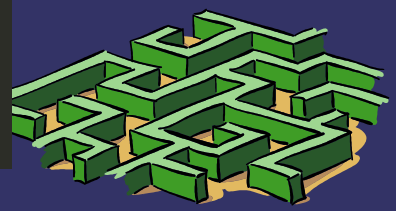
links_followed_ratio.title	links_followed_ratio.total	links_followed_ratio.links_follwed	links_followed_ratio.precentage_followed_link
List_of_Hindi_film_families	242165	219340	90.57460822166703
Associate_Justice_of_the_Supreme_Court_of_the_United_States	507604	441744	87.02531894941727
List_of_serial_killers_in_the_United_States	514546	420780	81.77694511277903
Edward_VII	274872	220265	80.13366221368491
George_V	388948	309566	79.59058794491808
2020_US_Open_-_Women's_Singles	347362	274380	78.98964192974476
The_Karate_Kid_(franchise)	368469	285573	77.50258502072087
UFC_Fight_Night:_Overeem_vs._Sakai	298715	224858	75.27509499020806
2020_US_Open_-_Men's_Singles	486360	365165	75.08121556049016
UFC_Fight_Night:_Waterson_vs._Hill	353036	261619	74.10547366274261
Atlético_Madrid	317796	233889	73.59721330664955
List_of_pornographic_performers_by_decade	635669	467454	73.53732838946055
2004_United_States_presidential_election	330196	242779	73.5257241153739
Wayans_family	330834	242476	73.29234601038588
The_Suicide_Squad_(film)	325162	236481	72.72713293681304
Eternals_(film)	412189	298579	72.4374012892144
UFC_Fight_Night:_Covington_vs._Woodley	640340	458650	71.6260111815598
George_VI	505928	359992	71.15478882370614
1992_United_States_presidential_election	292352	207533	70.98737138791593
2020-21_Premier_League	490358	347150	70.79521492460611
Inter_Milan	376595	265848	70.59254636944198
Everton_F.C.	707890	497610	70.29481981663818
A.C._Milan	398285	279162	70.09101522778914
Marvel_Cinematic_Universe:_Phase_Four	378636	264393	69.82775013469401
Newcastle_United_F.C.	321373	223015	69.39444197241212

Q3:
What series of wikipedia articles, starting with [Hotel California] keeps the largest fraction of its readers clicking on internal links?

- One way to approach this is by repeatably iterating queries on a Clickstream table, setting a conditional with WHERE prev = {the referring page} with the results of previous query, ordering by the most linked to page title.

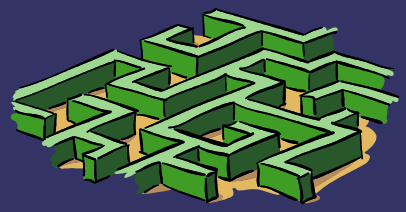


```
-- Hotel_California -> 2222 Hotel_California_(Eagles_album)
   SELECT prev, curr, type, n from CLICKSTREAM
   WHERE prev = "Hotel_California"
   order by n desc
   limit 1;

- Hotel_California_(Eagles_album) -> 2127 The_Long_Run_(album)
   SELECT prev, curr, n from CLICKSTREAM
   WHERE prev = "Hotel_California_(Eagles_album)" AND type = "link"
   order by n desc
   limit 1;
- The_Long_Run_(album)   -> 1322 Eagles_Live
   SELECT prev, curr, n from CLICKSTREAM
   WHERE prev = "The_Long_Run_(album)" AND type = "link"
   order by n desc
   limit 1;

-  Eagles_Live  -> 1136 Eagles_Greatest_Hits,_Vol._2
   SELECT prev, curr, n from CLICKSTREAM
   WHERE prev = "Eagles_Live" AND type = "link"
   order by n desc
   limit 1;

- Eagles_Greatest_Hits,_Vol._2  | The_Very_Best_of_the_Eagles  | 996
    SELECT prev, curr, n from CLICKSTREAM
   WHERE prev = "Eagles_Greatest_Hits,_Vol._2" AND type = "link"
   order by n desc
   limit 1;

- The_Very_Best_of_the_Eagles  | Hell_Freezes_Over  | 892  |
   SELECT prev, curr, n from CLICKSTREAM
   WHERE prev = "The_Very_Best_of_the_Eagles" AND type = "link"
   order by n desc
   limit 1;
```



Another way is cramming it all into one query, like so:

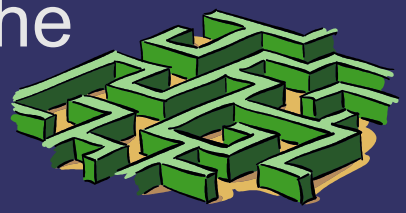
```
--- One possible way to iterate the results of our query. Very ugly though

    select * from CLICKSTREAM
    where prev in (
    select curr from CLICKSTREAM
    where prev in (
    SELECT curr from CLICKSTREAM
    where prev in
    ( SELECT curr from CLICKSTREAM
    WHERE prev = "Hotel_California" AND type = "link"
    order by n desc limit 1)
    order by n desc limit 1)
    order by n desc limit 1)
    order by n desc limit 1 ;
```



Q4

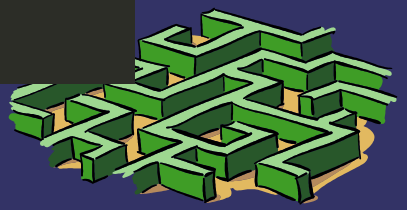
Find an example of an English wikipedia article that is relatively more popular in the UK. Find the same for the US and Australia.

- Answering this we will make use of the Wiki history dataset and make some assumptions about UK/US/AUS users of Wikipedia.
- One of those assumptions is that people are most often to engage Wikipedia when its about the afternoon in their time zone. (The event_timestamp is set with UTC)
- The other assumption is that more popular pages are more heavily edited as well for better or worse.
- We start out then by creating tables for the top_revised articles for these times.



Q4:
Dividing by Timezone

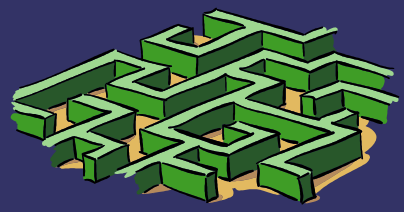
```
  -- 10 - 2 pm
create table UK_top_articles as  select  page_title, count(page_title) as n
    from revisions
    where (event_entity = "page" or event_entity = "revision")
    and hour( event_timestamp) > 10 and hour(event_timestamp) < 14
    group by page_title
    order by n desc
     limit 100;

-- e coast 12 - 6pm w coast 9am -3pm
   create table US_top_articles as  select  page_title, count(page_title) as n
    from revisions
    where (event_entity = "page" or event_entity = "revision")
    and hour( event_timestamp) > 17 and hour(event_timestamp) < 23
    group by page_title
    order by n desc
     limit 100;

     create table Aus_top_articles as select  page_title, count(page_title) as
     n
    from revisions
    where (event_entity = "page" or event_entity = "revision")
    and hour( event_timestamp) > 3 and hour(event_timestamp) < 9
    group by page_title
    order by n desc
     limit 100;
```



Q4 cont-

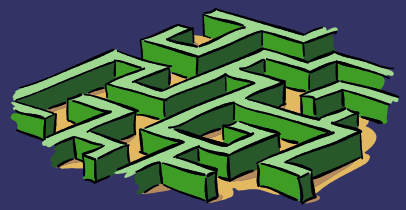
We then are going to make some queries to get a rough deviation from an average of revisions on shared articles between the regions. We will be able to have a rough view of interest across time zones, but no guarantee of user's individual location. We need to keep in mind as well population sizes of the respective regions.

- 328.2 million US
- 66.65 million UK
- 24.99 million Aus
-
- The reason for this is because of privacy concerns for Wikipedia's users.



- Here is very ugly query
-
- Joining our Top_articles for each “region”
- Page_title
- Average – is the average revision to an article across regions
- Deviation – is over-preforming or under-preforming of a region to the average

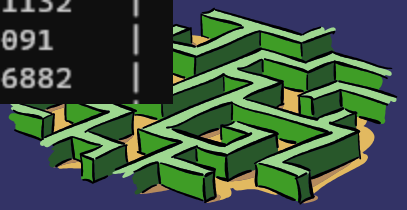
```
select UK_top_articles.page_title, FLOOR((UK_top_articles.n + US_top_articles.n + Aus_top_articles.n)/3) as average,
(UK_top_articles.n / FLOOR((UK_top_articles.n + US_top_articles.n + Aus_top_articles.n)/3)) as deviation  from UK_top_articles
inner join US_top_articles
on UK_top_articles.page_title = US_top_articles.page_title
inner join Aus_top_articles
on UK_top_articles.page_title = Aus_top_articles.page_title
order by deviation desc ;
```



UK Deviations

- Following slides are ordered by deviation from average revisions in October.

uk_top_articles.page_title	average	deviation
2020_Nagorno-Karabakh_conflict	1124	0.9608540925266904
Usernames_for_administrator_attention/Bot	211	0.933649289099526
Usernames_for_administrator_attention	749	0.931909212283044
Administrator_intervention_against_vandalism	1577	0.8902980342422321
In_the_news/Candidates	488	0.8709016393442623
Mccapra/sandbox	259	0.8262548262548263
Baath_Party_(disambiguation)	240	0.8208333333333333
AmandaNP/SPI_case_list	313	0.792332268370607
Murder_of_Samuel_Paty	206	0.7718446601941747
Teahouse	769	0.7685305591677504
Deaths_in_2020	465	0.7677419354838709
AmandaNP/UAA/Wait	534	0.7584269662921348
WikiProject_Spam/LinkReports	285	0.7578947368421053
Requested_moves/Current_discussions_(alt)	252	0.75
Requested_moves/Current_discussions	255	0.7490196078431373
Dashboard/Requested_moves	242	0.7479338842975206
Help_desk	316	0.7373417721518988
Cyberbot_I/Requests_for_unblock_report	265	0.7320754716981132
AmandaNP/unblock_table	275	0.730909090909091
Reliable_sources/Noticeboard	263	0.7300380228136882

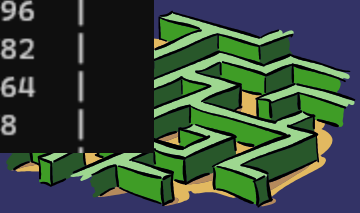


Australian timezone deviations

aus_top_articles.page_title	average	deviation
Murder_of_Samuel_Paty	206	1.2378640776699028
Donald_Trump	305	1.2065573770491804
WikiProject_Spam/LinkReports	285	1.1964912280701754
COVID-19_pandemic_data	576	1.1961805555555556
2020_Pacific_typhoon_season	231	1.1818181818181819
WikiProject_Ghana/Discussions	217	1.1705069124423964
AmandaNP/UAA/Time	805	1.15527950310559
SDZeroBot/GAN_sorting/styles.css	272	1.1544117647058822
SDZeroBot/PROD_sorting/beta/styles.css	272	1.1544117647058822
SDZeroBot/Peer_reviews/styles.css	272	1.1544117647058822
SDZeroBot/PROD_sorting/styles.css	272	1.1544117647058822
SDZeroBot/Redirectify_Watch/styles.css	272	1.1544117647058822
SDZeroBot/Declined_AFCs/styles.css	273	1.15018315018315
DatBot/pendingbacklog	424	1.1485849056603774
Community_portal/Opentask	275	1.1454545454545455
White_House_COVID-19_outbreak	486	1.1296296296296295
Requested_moves/Current_discussions_(table)	475	1.0863157894736841
AmandaNP/UAA/Wait	534	1.0674157303370786
Administrator_intervention_against_vandalism/TB2	593	1.0657672849915683
Sandbox	673	1.0386329866270432
DannyS712_bot_III/Redirects.json	439	1.029612756264237
Baath_Party_(disambiguation)	240	1.0
Good_article_nominations	223	0.9775784753363229
Teahouse	769	0.9726918075422627

US Deviations

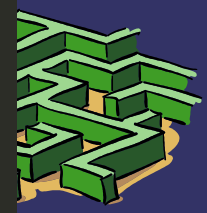
us_top_articles.page_title	average	deviation
2020_Atlantic_hurricane_season	406	1.9532019704433496
Biden-Ukraine_conspiracy_theory	265	1.8150943396226416
Administrators'_noticeboard	306	1.7124183006535947
Mccapra/sandbox	259	1.6254826254826256
Administrators'_noticeboard/Incidents	776	1.5850515463917525
Administrators'_noticeboard/Edit_warring	223	1.5560538116591929
White_House_COVID-19_outbreak	486	1.52880658436214
Proud_Boys	241	1.5145228215767634
Articles_for_creation/recent	187	1.4919786096256684
Requests_for_undeletion	217	1.4884792626728112
In_the_news/Candidates	488	1.4877049180327868
Did_you_know	164	1.4817073170731707
AnomieBOT/SPERTable	330	1.4666666666666666
AmandaNP/SPI_case_list	313	1.4664536741214058
Help_desk	316	1.4620253164556962
Requested_moves/Current_discussions	255	1.4392156862745098
Dashboard/Requested_moves	242	1.43801652892562
Requested_moves/Current_discussions_(alt)	252	1.4365079365079365
Requests_for_page_protection	700	1.4185714285714286
Good_article_nominations	223	1.3946188340807175
Cyberbot_I/Requests_for_unblock_report	265	1.3924528301886792
AmandaNP/unblock_table	275	1.3890909090909092
Deaths_in_2020	465	1.3698924731182796
WikiProject_Articles_for_creation/Help_desk	174	1.3505747126436782
Reliable_sources/Noticeboard	263	1.3422053231939164
DannyS712_bot_III/Redirects.json	439	1.284738041002278



Q5. Analyze how many users will see the average vandalized Wikipedia page before the offending edit is reversed.

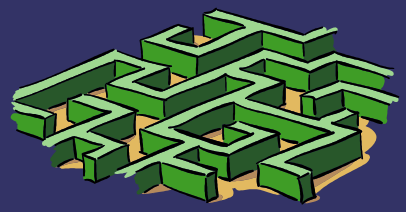
- To answer this we query the wiki history, specifically the event_entity for revisions, which has a value for revisions which are reverted to their previous identity.
- I am making an assumption that vandalized pages will receive close to this range, even though we are **not explicitly checking for pages with vandalized information or targeted attacks**. Vandalized reverts vs innocent reversions are indistinguishable without additonal modeling.

```
-- Average time to revert

select Count(*) AS Total_Revisions,
       Round(AVG(revision_seconds_to_identity_revert)) AS averageSecondsToRevert ,
       Round(AVG(revision_seconds_to_identity_revert)/60) AS averageMinToRevert,
       Round(AVG(revision_seconds_to_identity_revert)/3600) AS averageHourToRevert,
       Round(AVG(revision_seconds_to_identity_revert)/86400, 3) AS averageDayToRevert
  from revisions
 where revision_is_identity_reverted = true;
```



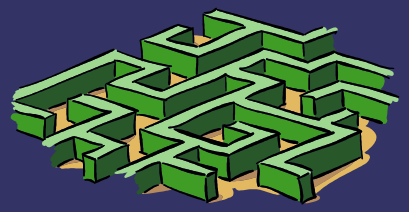
The averages from the previous queries.

total_revisions	averagesecondstorevert	averagemintorevert	averagehourtorevert	averagedaytorevert
483057	96793.0	1613.0	27.0	1.12



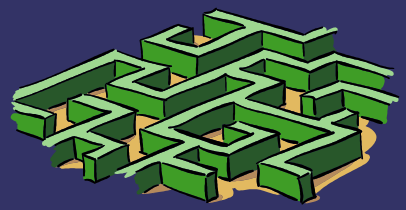
A bit additional thoughts:

- I thought about taking the average views pages receive and then comparing the time that a vandalized page might stay up.
- The problem with this approach from my perspective, is the average view for a page is about 12 views a day throughout the month of Sept.
- While the Average revision times given are a good base to work from, the difference between high traffic pages vs low traffic pages evidently swing wide .



Q6:
Who is the user that has produced the most information by bytes for Wikipedia?

```
-- revisions / page
    create table Top_Contrib as select  event_user_id, sum(revision_text_bytes) as
    revisionByteSize from revisions
    where not event_entity = "user"
    and EVENT_TYPE = "create"
    and event_user_id > 0
    group by event_user_id
    order by revisionByteSize desc
    limit 20 ;

    select event_user_text, Sum(revisionByteSize) as revisionByteSize  from Top_Contrib
    inner join revisions
    on Top_Contrib.event_user_id = revisions.event_user_id
    group by event_user_text
    order by revisionByteSize desc
    limit 20;
```



Very productive bots

event_user_text	revisionbytesize
Citation bot	706316255112424
AnomieBOT	122254036458051
WP 1.0 bot	84187223486865
DeltaQuadBot	60365958953370
ClueBot NG	46507345207167
AAlertBot	34894290179815
WikiCleanerBot	25466313954033
Materialscientist	16969743879800
Lowercase sigmabot III	16367703517710
RMCD bot	15302873270711
InceptionBot	14825399129772
Tom.Reding	13811528098125
Liz	12889857415529
SDZeroBot	10350410986330
Monkbot	9321963914959
MediaWiki message delivery	8360287391568
FrescoBot	7298309415267
Legobot	4037496775094
Celestina007	3097396066744
Lima Bean Farmer	423540579410