1. Number of sessions (total and valid) of a game and the average session time(only valid) from the given dataset are shown below -

game_id	Total Session	Valid Session	Average Session Time(seconds)
10483946	38	22	403
10655437	1528	428	205
18121481	1199	383	335
19049241	38	25	446
21554188	4	3	154
30900473	299	178	255
32950170	54	39	352
35643983	3	0	NA
40904844	3	3	231
43346372	644	440	398
55107008 **55107008**	***19639*** **5866**	***13713*** **3973**	***507*** **440**
60374084	6	3	318
67544688	13	5	118

Device with 034337e7ebb005a30525e7b71174155c ai5 looks like an outlier (maybe a test device) because of very huge number of sessions. In general, valid sessions per device is in range 1-8 and for this device it is 9740 and can be considered as outlier. Session details shown above for 55107008 game_id including this device is shown in red and excluding this device is shown in black.

Calculation Strategy -
Script goes to the log file line by line to read the event and stores it in a active_session dictionary with key being the combination of game_id and ai5 and value stores a dictionary with multiple key value players corresponding to ai5, ggstart , ggstop, session time, ggstartA. ggstartA will be inserted in the dictionary when there is a continuation of the session within 30 secs and it will help in calculating the session time because the time between ggstop and next ggstart event is not considered while calculating session time. In case of consecutive ggstart (without ggstops) events, only the last one is considered (followed by ggstop) because of the data loss and vice versa for multiple ggstop events.

Data is being stored in MySQL for this calculation and given the size of dataset for 1 hour and assuming it will increase with the constant rate, we can store the data in structured format in MySQL for analysis purpose. Basically, the script goes over the log and stores the session details per game per user with validity of session in MySQL.
Snippet below provides a better understanding of the logic for e8288d45e9efa574b3ae922ba218bc3d ai5 and 55107008 game_id

Entered First time in Active with ggstart 2016-05-09 02:05:28.072082
Entered First Time in Active with ggstop 2016-05-09 02:07:10.201470
STOP W Start 55107008 -> ggstop -> 2016-05-09 02:07:10.261133 e8288d45e9efa574b3ae922ba218bc3d
STOP W Start 55107008 -> ggstop -> 2016-05-09 02:07:10.538015 e8288d45e9efa574b3ae922ba218bc3d
Entered in Active with ggstartA 2016-05-09 02:07:10.623599
Updating ggstop in Active from 2016-05-09 02:07:10.201470 to this 2016-05-09 02:07:10.876070
Entered in Active with ggstartA 2016-05-09 02:07:11.219772
Updating ggstop in Active from 2016-05-09 02:07:10.876070 to this 2016-05-09 02:07:11.570300
Entered in Active with ggstartA 2016-05-09 02:07:11.621287
Updating ggstop in Active from 2016-05-09 02:07:11.570300 to this 2016-05-09 02:07:11.909625
STOP W Start 55107008 -> ggstop -> 2016-05-09 02:07:11.993565 e8288d45e9efa574b3ae922ba218bc3d
Entered in Active with ggstartA 2016-05-09 02:07:12.472320
Updating ggstartA in Active from 2016-05-09 02:07:12.472320 to this 2016-05-09 02:07:12.545565
Session Time for device e8288d45e9efa574b3ae922ba218bc3d playing game 55107008 is between 2016-05-09 02:05:28.072082 and 2016-05-09 02:07:11.909625 i.e. 103.020725

2. Game with **55107008** game_id is the most famous and most played game among the users in the given dataset.

3. Data Discrepancies -

- Device 034337e7ebb005a30525e7b71174155c with huge number of sessions is affecting the session calculation.
  Workaround - Mark it as outlier and ignore in calculation.
- Due to data loss i.e. missing of ggstop/ggstart event corresponding to previous respective event, session calculation is not accurate.
  Workaround - In case when multiple same events are encountered with less than 30 seconds gap, we can find an average of the differences and can add it to the session time. In case with gap more than 30 seconds, it will be difficult to do the same and it highly depend on the time gap between suck events.