REPORT

By Neelanjan Barman

Analytics Position Case Study

Problem Statement:

ABC is a real-money online gaming company providing multiplayer games such as Ludo. A user can register as a player, deposit money on the platform, and play games with other players on the platform. If he/she wins the game then they can withdraw the winning amount while the platform charges a nominal fee for the services. To retain players on the platform, the company ABC gives loyalty points to their players based on their activity on the platform. Loyalty points are calculated on the basis of the number of games played, deposits, and withdrawals made on the platform by a particular player.

The criteria to convert the number of games played, deposits, and withdrawals into points is given below:

Type of Action	Weightage per activity	Formulae	Example
Deposit of money on the platform	0.01	0.01 * Deposit Amount	0.01 * (1000 RS Deposit) = 10 Points
Withdrawal of money from the platform	0.005	0.005 * Withdrawal Amount	0.005 * (500 Rs Withdrawal) = 2.5 Points
How many more times did a player do deposit than withdrawal	0.001	0.001 * maximum of (#deposit - #windrawal) or 0	0.001 * max((5-3, 0)) = 0.001 * 2 = 0.002 points where number of deposit = 5 and number of withdrawal = 3
Number of games played	0.2	0.2 *Number of Games Played	0.2 * (50 Total Games Played) = 10 Points

Final Loyalty Point Formula

***Loyalty Point** = (0.01 * deposit) + (0.005 * Withdrawal amount) + (0.001 * (maximum of (#deposit - #withdrawal) or 0)) + (0.2 * Number of games played)*

At the end of each month, total loyalty points are allotted to all the players. Out of which the top 50 players are provided cash benefits.

We have three datasets –
User Gameplay Data, Deposit Data, and Withdrawal Data.

Column Definition for Datasets:

1. User ID: Unique ID for every user.
2. Games Played: Number of games played at that time.
3. Datetime: Timestamp.
4. Amount: Amount deposited/withdrawn by user.

Part A - Calculating loyalty points

On each day, there are 2 slots for each of which the loyalty points are to be calculated:
S1 from 12 am to 12 pm
S2 from 12 pm to 12 am
Based on the above information and the data provided answer the following questions:

1. Find Player wise Loyalty points earned by Players in the following slots:-
   a. 2nd October Slot S1
   b. 16th October Slot S2
   c. 18th October Slot S1
   d. 26th October Slot S2

2. Calculate overall loyalty points earned and rank players on the basis of loyalty points in the month of October. In case of a tie, the number of games played should be taken as the next criteria for ranking.

3. What is the average deposit amount?

4. What is the average deposit amount per user in a month?

5. What is the average number of games played per user?

Part B - How much bonus should be allocated to leaderboard players?

After calculating the loyalty points for the whole month find out which 50 players are at the top of the leaderboard. The company has allocated a pool of Rs 50000 to be given away as bonus money to the loyal players. Now the company needs to determine how much bonus money should be given to the players. Should they base it on the amount of loyalty points? Should it be based on number of games? Or something else? That's for you to figure out.

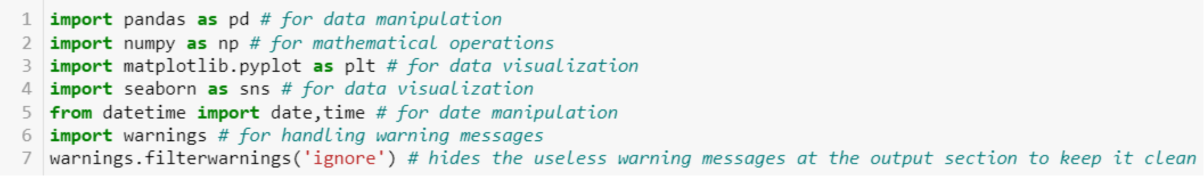
Suggest a suitable way to divide the allocated money keeping in mind the following points:
1. Only top 50 ranked players are awarded bonus

Part C -
Would you say the loyalty point formula is fair or unfair?
Can you suggest any way to make the loyalty point formula more robust?

Solution:

1. Importing the dependencies:

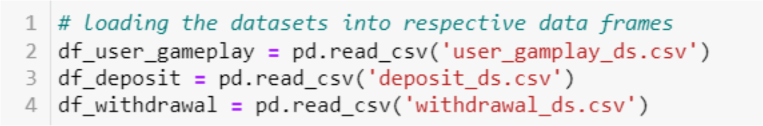
Imported the libraries that we need to solve this case study. The libraries imported are given below –



```
import pandas as pd # for data manipulation
import numpy as np # for mathematical operations
import matplotlib.pyplot as plt # for data visualization
import seaborn as sns # for data visualization
from datetime import date,time # for date manipulation
import warnings # for handling warning messages
warnings.filterwarnings('ignore') # hides the useless warning messages at the output section to keep it clean
```

2. Data Gathering:

Loaded the datasets into respective data frames.

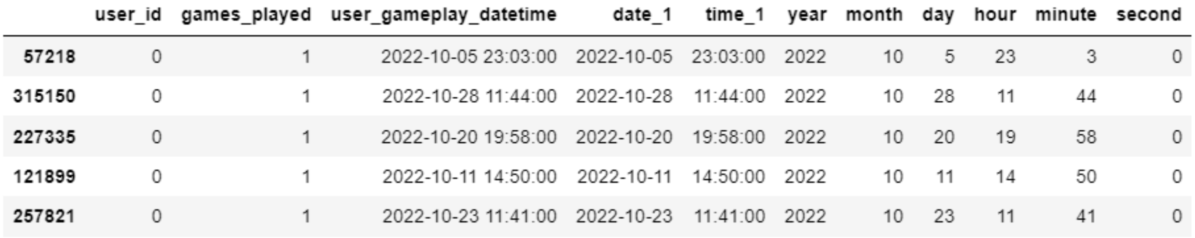


```
# loading the datasets into respective data frames
df_user_gameplay = pd.read_csv('user_gamplay_ds.csv')
df_deposit = pd.read_csv('deposit_ds.csv')
df_withdrawal = pd.read_csv('withdrawal_ds.csv')
```

3. Exploratory Data Analysis:

Exploring, understanding, preparing, and analyzing the datasets. Tasks performed in exploratory data analysis for respective data frames are –

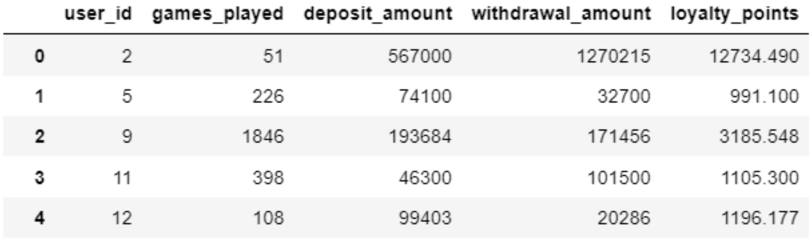
1. Checked the datatypes of each column.
2. Checked the number of rows and columns.
3. Checked the missing values in each column. (No missing values found)
4. Data Preparation.
5. Renamed the columns.
6. Sorted the data frames in ascending order based on their user ID.
7. Created a new data frame by merging other data frames which shows the number of games played, the amount deposited, and the amount withdrawn.
8. Added a column for the loyalty points by using the formula given in the problem statement.
9. Checking the statistical insights of the data frame.
10. Data Quality check (Checked whether there's a column with a value less than 0 or not. No such column found)
11. Separating date and time.
12. Extract year, month, day, hour, minute, and second and put them in separate columns.



	user_id	games_played	user_gameplay_datetime	date_1	time_1	year	month	day	hour	minute	second
57218	0	1	2022-10-05 23:03:00	2022-10-05	23:03:00	2022	10	5	23	3	0
315150	0	1	2022-10-28 11:44:00	2022-10-28	11:44:00	2022	10	28	11	44	0
227335	0	1	2022-10-20 19:58:00	2022-10-20	19:58:00	2022	10	20	19	58	0
121899	0	1	2022-10-11 14:50:00	2022-10-11	14:50:00	2022	10	11	14	50	0
257821	0	1	2022-10-23 11:41:00	2022-10-23	11:41:00	2022	10	23	11	41	0

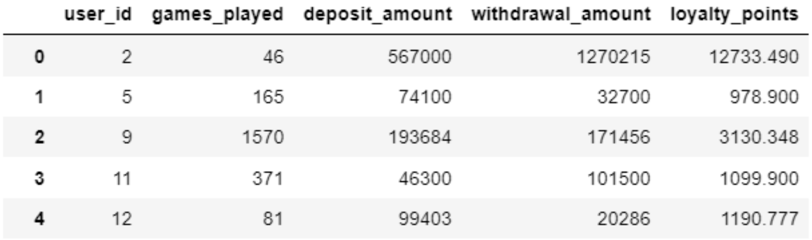
4. Part 1: Calculating Loyalty Points Slot Wise

- For Slot S1, filtered the rows with values of the hour column being less than or equal to 12.
- Created a data frame for slot s1 grouped by user ID.
- Added loyalty points to the slot 1 data frame.



	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	51	567000	1270215	12734.490
1	5	226	74100	32700	991.100
2	9	1846	193684	171456	3185.548
3	11	398	46300	101500	1105.300
4	12	108	99403	20286	1196.177

- For Slot S2, filtered the rows with values of the hour column being greater than 12.
- Similarly, created the Slot S2 data frame.

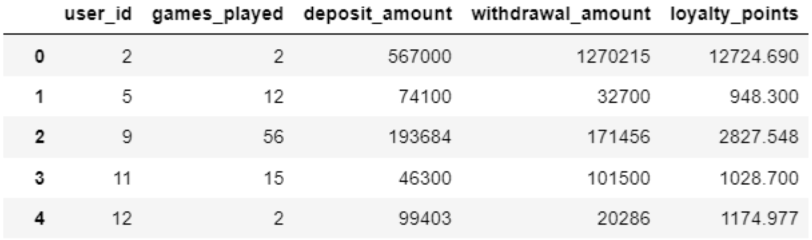


	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	46	567000	1270215	12733.490
1	5	165	74100	32700	978.900
2	9	1570	193684	171456	3130.348
3	11	371	46300	101500	1099.900
4	12	81	99403	20286	1190.777

5. Solving the questions from the problem statement:

Q1. Find player-wise loyalty points earned by players in the following slots –

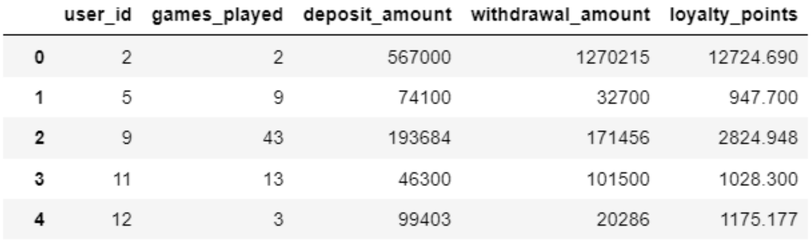
a. *2nd October Slot S1*



	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	2	567000	1270215	12724.690
1	5	12	74100	32700	948.300
2	9	56	193684	171456	2827.548
3	11	15	46300	101500	1028.700
4	12	2	99403	20286	1174.977

which is the required player-wise loyalty points earned by players in slot 1 of 2nd October.

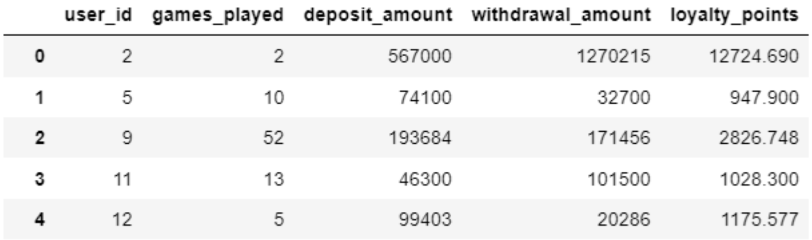
b. *16th October Slot S2*



	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	2	567000	1270215	12724.690
1	5	9	74100	32700	947.700
2	9	43	193684	171456	2824.948
3	11	13	46300	101500	1028.300
4	12	3	99403	20286	1175.177

which is the required player-wise loyalty points earned by players in slot 2 of 16th October.

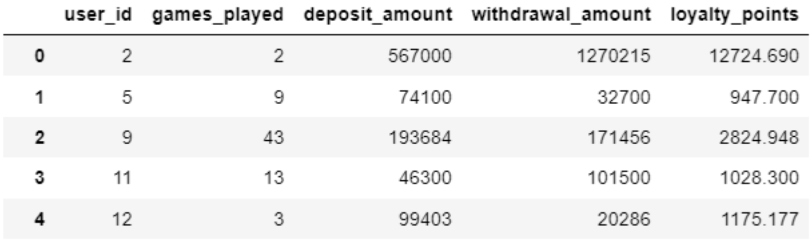
c. *18th October Slot S1*



	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	2	567000	1270215	12724.690
1	5	10	74100	32700	947.900
2	9	52	193684	171456	2826.748
3	11	13	46300	101500	1028.300
4	12	5	99403	20286	1175.577

which is the required player-wise loyalty points earned by players in slot 1 of 18th October.

d. *26th October Slot S2*

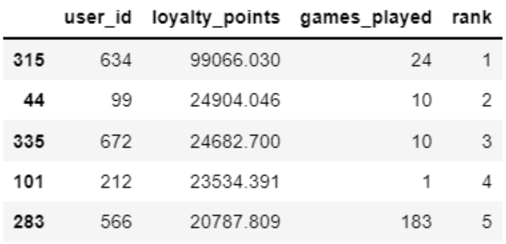


	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points
0	2	2	567000	1270215	12724.690
1	5	9	74100	32700	947.700
2	9	43	193684	171456	2824.948
3	11	13	46300	101500	1028.300
4	12	3	99403	20286	1175.177

which is the required player-wise loyalty points earned by players in slot 2 of 26th October.

Q2. Calculate overall loyalty points earned and rank players on the basis of loyalty points in the month of October. In case of a tie, the number of games played should be taken as the next criteria for ranking.

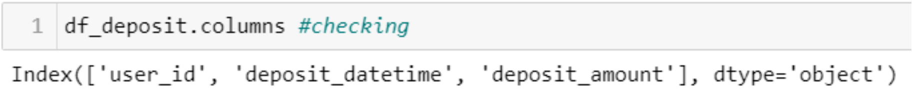
- Filtered rows with values of the month column being 10.
- Sorted the loyalty points column in descending order.
- Sorted the users based on loyalty points and games played in descending order.
- Incremented the index value by 1 to make the ranking start from 1 instead of 0.



	user_id	loyalty_points	games_played	rank
315	634	99066.030	24	1
44	99	24904.046	10	2
335	672	24682.700	10	3
101	212	23534.391	1	4
283	566	20787.809	183	5

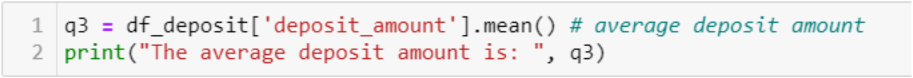
Hence, players are ranked based on their respective loyalty points in the month of October and in case of a tie, the number of games played has been taken as the next criteria for the ranking.

Q3. What is the average deposit amount?



```
df_deposit.columns #checking
```

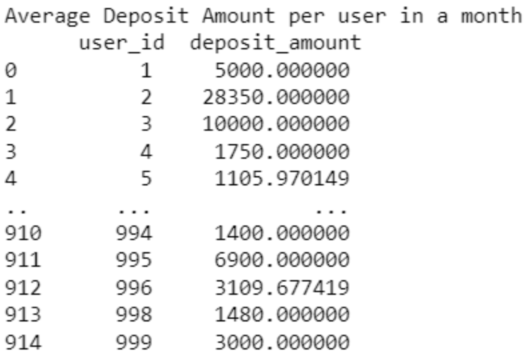
```
Index(['user_id', 'deposit_datetime', 'deposit_amount'], dtype='object')
```



```
q3 = df_deposit['deposit_amount'].mean() # average deposit amount
print("The average deposit amount is: ", q3)
```

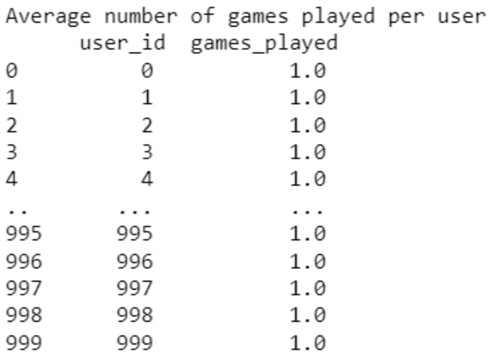
The average deposit amount is: 5492.185399701801

Q4. What is the average deposit amount per user in a month?



```
Average Deposit Amount per user in a month
      user_id  deposit_amount
0           1     5000.000000
1           2    28350.000000
2           3    10000.000000
3           4     1750.000000
4           5     1105.970149
..        ...             ...
910       994     1400.000000
911       995     6900.000000
912       996     3109.677419
913       998     1480.000000
914       999     3000.000000
```

Q5. What is the average number of games played per user?



```
Average number of games played per user
      user_id  games_played
0           0           1.0
1           1           1.0
2           2           1.0
3           3           1.0
4           4           1.0
..        ...           ...
995       995           1.0
996       996           1.0
997       997           1.0
998       998           1.0
999       999           1.0
```

6. Part B – How much bonus should be allocated to leaderboard players?

After calculating the loyalty points for the whole month find out which 50 players are at the top of the leaderboard. The company has allocated a pool of Rs 50000 to be given away as bonus money to the loyal players. Now the company needs to determine how much bonus money should be given to the players. Should they base it on the amount of loyalty points? Should it be based on number of games? or Something else?

Suggest a suitable way to divide the allocated money keeping in mind the following points:

a. *Only the top 50 ranked players are awarded bonus*

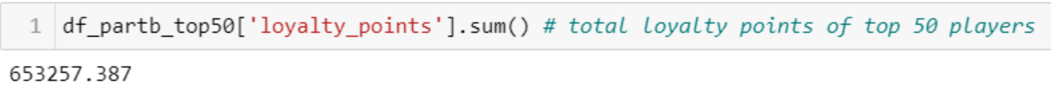
Ans:
Created a data frame that shows the top 50 players based on their loyalty points, and the number of games played in case of a tie.

I think the bonus money should be given to the players based on the amount of loyalty points because this is the only feature which carries an optimal weightage. (Loyalty Points have the weightage of the features games played, deposits, and withdrawals)

As we are distributing the 50000 prize pool among the top 50 players in the leaderboard based on their loyalty points, So we have to divide 50000 by Total Loyalty Point and multiply it by the individual loyalty points.

Therefore,

*Bonus = Loyalty Points * (50000 / Total Loyalty Point)*



```
df_partb_top50['loyalty_points'].sum() # total loyalty points of top 50 players
```

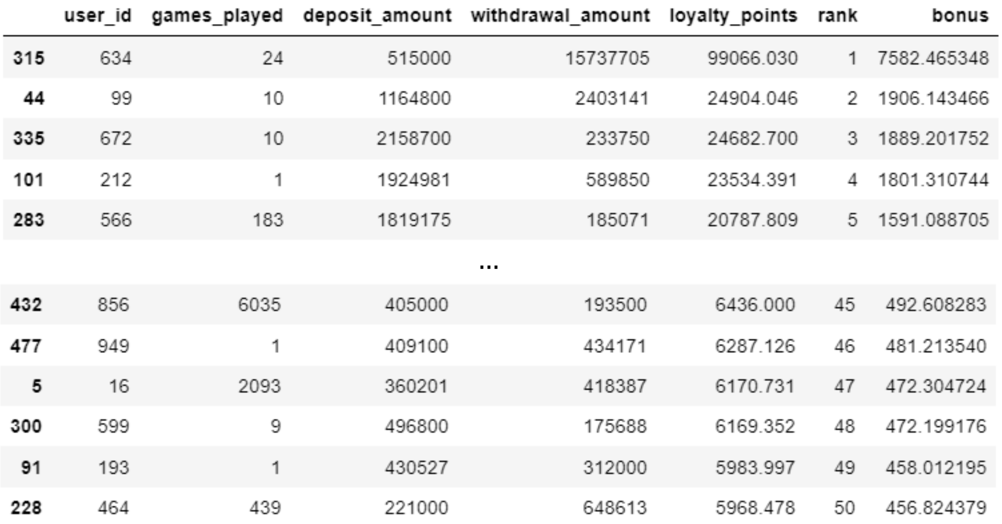
```
653257.387
```

So, the Total Loyalty Point of the Top 50 Players is 653257.387

Example:

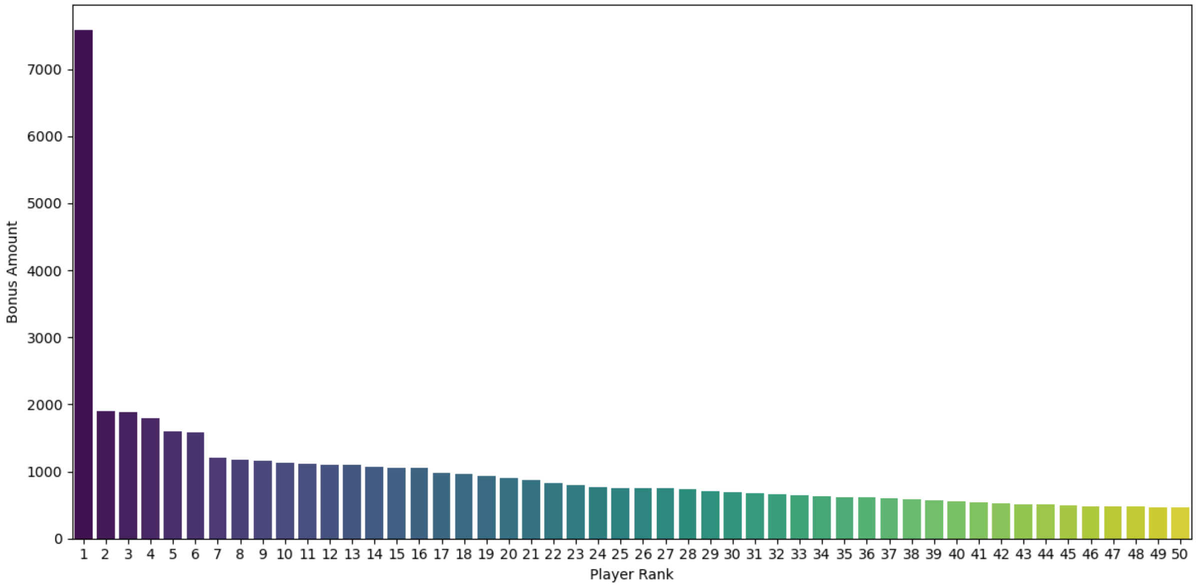
*Bonus (rank 1) = 99066.030 * (50000 / 653257.387)*
*= 99066.030 * 0.0765395095333*
= 7582.4653476134

- ➢ Creating a Bonus column.
- ➢ Displaying final data frame with bonus column.



	user_id	games_played	deposit_amount	withdrawal_amount	loyalty_points	rank	bonus
315	634	24	515000	15737705	99066.030	1	7582.465348
44	99	10	1164800	2403141	24904.046	2	1906.143466
335	672	10	2158700	233750	24682.700	3	1889.201752
101	212	1	1924981	589850	23534.391	4	1801.310744
283	566	183	1819175	185071	20787.809	5	1591.088705
				...			
432	856	6035	405000	193500	6436.000	45	492.608283
477	949	1	409100	434171	6287.126	46	481.213540
5	16	2093	360201	418387	6170.731	47	472.304724
300	599	9	496800	175688	6169.352	48	472.199176
91	193	1	430527	312000	5983.997	49	458.012195
228	464	439	221000	648613	5968.478	50	456.824379

- *Data Visualization* – Player Rank vs Bonus Amount using bar plot.



Hence, the top 50 players in the leaderboard received bonuses based on their loyalty points.

7. Part C –

Would you say the loyalty point formula is fair or unfair?

Ans: Yes, the loyalty point formula is fair enough but maybe we can do some adjustments to make it more robust or efficient.

Can you suggest any way to make the loyalty point formula more robust?

Ans: If we observe carefully, we can see that the weightage for the number of games played is quite high compared to others. Maybe we can reduce it a little bit to make it more robust. For Example: Currently, the weightage is (0.2 * number of games played)

- we can do either
    - (0.15 * number of games played) or
    - (0.1 * number of games played) to make it more robust.