Information Visualization

CHECKPOINT II: Data cleaning and processing

G13-A

1. Initial Dataset

Static tables from esportsearnings.com and worldbank.org's APIs, and scraping from esportsearnings.com. Our initial data comprised 4000 players (786 kB) and their earnings per age (923 kB), slightly under 30 000 tournaments (7.46 MB), 385 games (56.2 kB), 687 teams (111kB) and countries (43 kB).

Samples: [changed samples to tables after feedback]

TeamId	TeamName	TotalUSDPrize	TotalTournaments
102	Team Liquid	23951642.48	1395

GameId	GameName	TotalUSDPrize	TotalTournaments	TotalPlayers
151	StarCraft II	27674383.81	5113	1819

PlayerId	NameFirst	NameLast	CurrentHandle	CountryCode	TotalUSDPrize
3145	Kuro	Takhasomi	KuroKy	de	4097926.95

PlayerId	EarningsPerAge	
	Age	EarningsUSD
1872	24	22,000.00
	23	5,218.32
	22	15,380.29
	21	4,570.61

CountryCode	annual_GDP_USD	country_name	unemployment_total_percentage	urban_population_percentage
gb	2885570309160.9	United Kingdom	5.3	82.6

TournamentId	GameId	TournamentName	StartDate	EndDate	Location	Teamplay	TotalUSDPrize
30100	245	ESL Pro European Championship 2018	2018-10-13	2018-10-14	Poznań, Greater Poland, Poland	1	80965.50

2. Selected/Derived Data

We removed players' currentHandle, NameFirst, nameLast and PlayerId, game and team IDs, tournaments' Id, TournamentName, EndDate, Location, Teamplay and TotalUSDPrize. While the Location of tournaments would have been an interesting attribute to display, we had to remove it because the data is user-submitted and doesn't adhere to a specific format.

We didn't calculate any derived measures. The only table where we have missing data is the earningsPerAge table.

After processing, we ended up with tables for countries and their players/earnings, what months and years tournaments happened in, and the player earnings sorted by age. We also left the games and teams tables mostly intact (more specific descriptions below.)

3. Data abstraction

Countries: static table countries.json, containing a country's name and code, annual GDP, and unemployment/urban population metrics, number of players and the sum of their earnings. Attributes:

- countryName, countryCode: nominal; both identify a country.
- annualGDPUSD: quantitative, ratio; country's annual GDP.
- urbanPopPercentage: quantitative, ratio; country's urban population.
- unemploymentPercentage: quantitative, ratio; country's unemployment.
- players: quantitative, ratio; number of players with that country's nationality.
- totalUSDPrize: quantitative, ratio; sum of earnings of the country's players.

Games: static table games.json, containing a game's name, its earnings (total prize), tournaments and players. Attributes:

- gameName: nominal; identifies a game.
- totalUSDPrize: quantitative, ratio; the game's total prize money.
- totalTournaments: quantitative, ratio; number of tournaments for that game.
- totalPlayers: quantitative, ratio; number of players for that game.

Teams: static table teams.json, containing a team's ID, name, tournaments and earnings. Attributes:

- teamId, teamName: nominal; both identify a team/organisation.
- totalTournaments: quantitative, ratio; number of tournaments the team participated in.
- totalUSDPrize: quantitative, ratio; team's total earnings.

Earnings: static table earningsByAge.json, containing an age and the sum of earnings from all players when they were that age. Attributes:

- age: quantitative, ratio; an age number.
- earnings: quantitative, ratio: the sum earnings all players made when he was at that age.

Tournaments: static table tournaments.json, containing months and years, and how many tournaments happened in a month of a certain year. Attributes:

- tournaments: quantitative, ratio; how many tournaments happened in that month.
- startMonth: quantitative, hierarchical; month of each year.
- startYear: quantitative, hierarchical; year.

4. Dataset processing

The data for games, teams and players was obtained directly from the esportsearnings.com API. Player earnings by age data was scraped from the same site (using a node.js script to go to each player's "Tournaments won by age" page and making a .json file from it).

Country data was obtained from the worldbank.org API in .xlsx format and converted to .json afterwards, with the use of a Python script.

To make the countries table we used the countries and players data, and for the other tables we just removed attributes from the original data.

Problems: The esportsearnings.com API only let us get 100 players/tournaments/teams and one game at a time, with a limit of one query every 2 seconds, so we made a script to automate data collection. Some data, such as age and earnings per age, wasn't available in the API so we had to scrape it from each player page. The earningsPerAge table had missing values for some players, so we ignored them while making the earnings table.

The final data is smaller, comprising 240 months where there were tournaments (26.1 kB). The earnings by age go from 12 to 39 years old (2.35 kB).

5. Mapping (Data sample / Questions)

- **What countries have the highest earnings?**
  Use the earnings attribute from the countries table.
- **What is the age at which players earn the most?**
  Compare the data from the earnings table.
- **What organizations earned the most?**
  Display the (already sorted by earnings) teams table.
- **What games have the most earnings?**
  Display the (already sorted by earnings) games table.
- **What months are the most active in esports?**
  Use the data from the (already organised by month and year) tournaments table.
- **How does unemployment correlate with player earnings?**
  Use the data from the country table.