EC 2228 Problem Set: Moneyball on a Budget at Fenway

Instructor: Suraj Kumar

Due Date: TBA

Total: 100 points

Instructions

1. This homework has two parts: building the team models (questions 1 to 16, 50 points), and valuing the players and picking your team (questions 17 to 32, 50 points). You will be awarded a maximum of 100 points for this assignment.

2. In each question, the text highlighted in yellow tells you what to do. Green **Checkpoint** boxes show the numbers you should see if everything up to that point is correct. If your numbers differ, fix your code before moving on.

3. Create a do-file with your code, an automated log-file of your answers from that code, and fill out the provided Word template (`Moneyball_answer_template.docx` on Canvas) with your final answers as indicated.

4. Submit all three files (do-file, log-file, and Word document) as one zip file online via the Canvas homework tool. Failure to submit 1 out of 3 files will result in a 33.33% penalty on this homework grade. Failure to submit 2 out of 3 files will result in a 66.66% penalty on this homework grade.

5. This problem set is long, so we strongly recommend working in a group of two or three. You are encouraged to work with up to two partners (partners may be from other sessions, so in total a group may have a maximum of three students), but each student needs to submit solutions on Canvas. If you do work with partners, you should clearly state your partners' names in your Word file.

6. Name your files with your full name, beginning with your last name. For example, for John Smith the files would be named: `smith_john_moneyball.do`, `smith_john_moneyball.log`, and `smith_john_moneyball.docx`, all inside `smith_john_moneyball.zip`.

7. For any questions related to this assignment or any concerns related to grading, please contact Suraj (Raj) at `suraj.kumar@bc.edu`.

1 Moneyball, Part 1: What Wins Games?

A sports analyst and a financial analyst have a lot in common: both love numbers, and both have to extract the signal from the numbers.[1] The job of a financial analyst is to figure out the correct price of a financial security, to help decide whether to buy or sell it (buy if the correct price > the market price, and sell otherwise). A sports analyst working with a team is concerned with identifying which traits make a team win, and then how best to put together a team that has those traits within the team's budget.



Figure 1: The Green Monster at Fenway Park. Photo: Adam Chivers, Wikimedia Commons, CC BY-SA 4.0.

Imagine that you have just been appointed general manager of the Boston Red Sox.[2] Your new office looks out on the Green Monster.[3] You are excited about baseball, no doubt,[4] and the excitement is doubled by your recent promotion to general manager. One of your most important tasks as general manager is to pick the players for the season. The team owner has given you a budget of $150 million. You are worried about competing later in the season with

[1] This assignment is motivated by the students in my class who play baseball.

[2] The Red Sox play in the East division of the American League, with the New York Yankees and the Toronto Blue Jays. After the 2002 season, the Red Sox tried to hire Billy Beane, the Oakland general manager from *Moneyball*. When he turned them down, they hired Bill James, the founder of sabermetrics, as an adviser, and in 2004 they won their first World Series since 1918.

[3] The Green Monster is the 37-foot wall in left field at Fenway Park, the oldest ballpark in the major leagues (1912). It has been painted green since 1947, and its scoreboard is still changed by hand.

[4] If you have never watched a baseball game, look at the box score of Game 7 of the 2025 World Series, in which the Los Angeles Dodgers beat the Toronto Blue Jays 5–4 in 11 innings: `https://www.espn.com/mlb/boxscore/_/gameId/401809303`. It shows the runs each team scored in each inning, and the batting and pitching statistics of every player, many of which you will use in this problem set.

your rivals in the American League East, such as the New York Yankees and the Toronto Blue Jays, both of whom spent far more than your budget in 2025.

> **From: Team Owner**
>
> Congratulations on the job! I know we spent $193 million last year. This year you get $150 million, and the Yankees spend $278 million. Can we still beat them? I'd like an answer before spring training.

You want to make sure that every dollar you spend is worth it. As a Boston College graduate you love sports, and as an Economics major you have learned to analyze numbers, so by now you are very proficient in sabermetrics, the use of statistics to measure what actually wins baseball games.[5] You devise a strategy to select the best possible players given your budget.

> **From: Assistant GM**
>
> Boss, fun fact: in 2002 the Oakland A's had about a third of the Yankees' payroll and won the same number of games (103). Their secret was not a better scout. It was a regression.

Teams Data: Load, Check and Clean

> **From: Data Intern**
>
> I downloaded 150 years of baseball. One team was called the Brooklyn Bridegrooms, and the 1899 Cleveland Spiders went 20–134. Should we keep them in the sample?
>
> Also, the scoreboard on the Green Monster is still updated by hand. I asked if we could use Stata instead. They said no.

First, you want to understand which characteristics in batting, pitching (or bowling) and fielding make a winning team. You have baseball statistics from the SABR Lahman Baseball Database.[6]

1. Let's begin by loading the data from *Teams.csv*. Stata does not like variable names in the csv that begin with a number, such as *2B* or *3B*. Find out what names Stata has given them, and then rename *2B* as `doubles` and *3B* as `triples`. **Hint:** To import a csv into Stata we use something like `import delimited "name.csv", clear`, where `name.csv` stands for the name of the file being imported, and `clear` clears the existing data in Stata. Remember to always run `describe` after loading any dataset. To rename a variable we use `rename oldvarname newvarname`. **[3 points]**

2. Check the ID of the dataset. Is this cross-sectional, time-series or panel data? **Hint:** Cross-sectional data vary across different units at one point in time, whereas time-series data

[5] The name comes from SABR, the Society for American Baseball Research. Sabermetrics became famous through *Moneyball* (Michael Lewis, 2003), the story of the 2002 Oakland Athletics. They had one of the smallest payrolls in the league, but used statistics such as on-base percentage to find good players that other teams undervalued.

[6] 2025 release, `https://sabr.org/lahman-database/`. The data are free to use under a Creative Commons licence (CC BY-SA 3.0).

vary across different points in time for one unit. Panel data combine both, so they vary across both time and units. In this data, you can think of time as `yearid` and units as `teamid`. You check the ID of the data in Stata by typing `isid varA varB`, which checks whether var$A \times$ var$B$ uniquely identifies the dataset. Refer to Table 1 for some potential candidates. **[3 points]**

3. Describe the data. What does Stata show it is sorted by, if anything? You are not happy with it, and you want to set the structure of the data properly. To do that, you first want to *encode* `teamid` as a numeric variable instead of a string. You need this later to tell Stata the structure of the data. Rename `teamid` to `teamid_string`, and then `encode teamid_string` into a numeric `teamid`. Then set the structure of the data and describe it again. How is the data sorted now compared with earlier, and why has it changed? **Hint:** Panel data in Stata require the variable that identifies a unit to be numeric. To do that, you run `encode unit_name_in_string, gen(unit_name_in_numeric)`. Afterwards, you run `xtset unit_dimension_var time_dimension_var`, where `unit_dimension_var` identifies the cross-sectional variation and `time_dimension_var` identifies the time dimension. For more information on `xtset`, you can type `help xtset` in the Stata command window. **[3 points]**

4. Since baseball has changed over time, you want to focus on the more recent period, from 2000 onwards, to understand which baseball metrics matter for winning. So you want to filter out the observations before 2000. Also, there is something weird about the number of games played in 2020. Find out what it is, and then drop the observations for 2020. How many observations are you left with now, compared with what you started with in *Teams.csv*? **Hint:** To keep observations where a variable is greater than or equal to some value, you run `keep if varname >= value`, and to drop observations where a variable is equal to some value, you run `drop if varname == value`. Always watch out for `==` whenever testing equality. I always mess it up in class. **[4 points]**

> ✅ **Checkpoint:** if everything so far is correct, you should see the following.
> 750 observations: 30 teams in each of the 25 seasons from 2000 to 2025, without 2020.

Batting Model

> ✉ **From: Head Scout**
> I've found our guy. Good face, nice swing. Trust me, I've been doing this for 30 years. 🧢

You thank him, and open Stata instead. Now you finally have your sample, which you would like to analyze to build a batting model that identifies the traits that explain the runs scored.

5. Take a look at the batting variables in your data. Table 2 explains what they mean. In baseball, the most commonly used *statistics* are *OBP*, or on-base percentage, and *SLG*, or slugging. Think of on-base percentage as how often a batter turns a visit to the plate into reaching *base*, instead of getting out. Think of slugging as how many

Table 1: Candidate ID variables in *Teams.csv*

Variable	Label in Stata	Description
`yearid`	yearID	Season, from 1871 to 2025.
`lgid`	lgID	League: `AL` (American League) or `NL` (National League) since 1901; older leagues before that.
`teamid`	teamID	Three-letter team code, for example `BOS` for the Boston Red Sox and `NYA` for the New York Yankees.
`franchid`	franchID	Three-letter franchise code. It stays the same when a team moves city or changes its name.
`divid`	divID	Division within the league: `E` (East), `C` (Central) or `W` (West).
`name`		Team's full name, for example "Boston Red Sox".
`park`		Name of the team's home ballpark.
`teamidbr`	teamIDBR	Team code used by the Baseball-Reference website.

Notes: "Label in Stata" is the variable label shown by `describe`; it keeps the original column name from the csv file. *Teams.csv* has 3,614 rows. Source: SABR Lahman Baseball Database, 2025 release, `https://sabr.org/lahman-database/`.

bases a batter gets per at bat, which measures power. Table 3 shows how to derive OBP, SLG, OPS, AVG and K/BB from the variables in Table 2. Generate the on-base percentage, slugging, and the combined statistic *on-base plus slugging*, or OPS. **Hint:** To generate a new variable from some math, you type `gen newvar = math expression`. Refer to Table 3 for the mathematical expressions for `obp`, `slg` and `ops`. **[3 points]**

6. Let's look at runs per game. Generate a variable called `rpg` as the ratio of `r` and `g`, and summarize it. Is it more like points in American football or like goals in soccer? Or like any other sport that you are interested in? **Hint:** Compare per team and per game: a soccer team scores about 1.5 goals a game, an NFL team about 22 points, and a T20 cricket team about 160 runs. **[2 points]**

> **From: Assistant GM**
>
> Since 2000 we have hit 337 doubles a season; the average team hits 282. Thank the Green Monster: fly balls that are outs in other parks bounce off it for doubles. Then check what doubles do to SLG.

7. Before you run anything, guess: which matters more for scoring runs, getting on base (OBP) or hitting for power (SLG)? Write your guess down. You will check it in question (8). Now regress `rpg` first on `obp`, and call it Model $\mathcal{M}_1$, and then regress `rpg` on both `obp` and `slg`, and call it Model $\mathcal{M}_2$. Compare $\mathcal{M}_1$ and $\mathcal{M}_2$ on both R-squared and adjusted R-squared. Which model do you prefer, and why? Also consider a third model, $\mathcal{M}_3$, which regresses `rpg` on `ops`. How does $\mathcal{M}_3$ compare with $\mathcal{M}_2$? **[5 points]**

☑ **Checkpoint:** if everything so far is correct, you should see the following.

Means: `rpg` 4.549, `obp` 0.325, `slg` 0.414, `ops` 0.738.

$$\mathcal{M}_3: \quad \widehat{\text{rpg}} = \underset{(0.114)}{-4.406} + \underset{(0.154)}{12.131}\,\text{ops}, \qquad N = 750,\ R^2 = 0.892$$

Standard errors are in parentheses.

8. You think that $\mathcal{M}_3$ makes much better sense than $\mathcal{M}_1$ and $\mathcal{M}_2$, as it is much simpler. However, you want to be sure that `obp` and `slg` are not statistically different in $\mathcal{M}_2$. Test the hypothesis (H_0) that `obp` and `slg` contribute equally to `rpg` in model $\mathcal{M}_2$ with an F-test. What is your conclusion? Does it surprise you? **Hint:** In the regression `reg A B C`, to test for equality of the coefficients on `B` and `C`, we run `test B = C`.

   Has this changed over the years? Estimate $\mathcal{M}_2$ again on the entire history in *Teams.csv*, and do the F-test again. Re-importing *Teams.csv* would wipe out your 2000–2025 sample, which you need for the rest of the problem set, so use `preserve` and `restore`. `preserve` takes a snapshot of the data in memory. Whatever you do afterwards, `restore` brings the data back exactly as it was at `preserve`. Your scalars are not part of the data, so `restore` does not touch them. Copy the following code into your do-file, and run the whole block at once, from `preserve` to `restore`: if you run only part of it, Stata restores the data as soon as that part finishes.

```
preserve
    * the whole history, no years dropped
    import delimited "Teams.csv", clear
    rename b doubles
    rename v19 triples
    gen rpg = r/g
    gen obp = (h + bb + hbp)/(ab + bb + hbp + sf)
    gen slg = (h + doubles + 2*triples + 3*hr)/ab
    reg rpg obp slg
    test obp = slg
    * which seasons did the regression use?
    summ yearid if e(sample)
restore

* check: you should be back to your 2000-2025 sample
count
```

   Which seasons does the regression use, and why does it start where it does? Do `obp` and `slg` still contribute equally? Was your guess right? **Hint:** `obp` is missing in the seasons in which `hbp` or `sf` were not recorded. **[5 points]**

9. Save the coefficient on `ops` in $\mathcal{M}_3$ as the scalar `b_ops`. **Hint:** Remember `obp`, `slg` and `ops` are all fractions, so a unit change is too much; perhaps a better *change* is an increment of 0.01. Summarize them with `summ obp slg ops` to find out. To save the coefficient after running a regression such as `reg y x`, you can do `scalar b_x = _b[x]`, and check it by displaying it

afterwards with `display b_x`. Table 6 lists all the scalars you will store in this problem set. **[2 points]**

Table 2: Batting variables in *Teams.csv*

Variable	Label in Stata	Description
`g`	G	Games played in the season (162 in a normal season).
`r`	R	Runs scored by the team.
`ab`	AB	At bats: times a batter came to bat, not counting walks, hit by pitch and sacrifice flies.
`h`	H	Hits: the batter hits the ball into play and reaches base safely.
`b`	2B	Doubles: hits on which the batter reaches second base. Rename to `doubles`.
`v19`	3B	Triples: hits on which the batter reaches third base. Rename to `triples`.
`hr`	HR	Home runs: the ball leaves the park and the batter scores (like a six in cricket).
`bb`	BB	Walks ("base on balls"): the pitcher throws four bad balls and the batter goes to first base for free.
`hbp`	HBP	Hit by pitch: the pitch hits the batter, who goes to first base for free.
`sf`	SF	Sacrifice flies: the batter is caught out, but a teammate on third base runs home and scores.
`so`	SO	Strikeouts: the batter is out on three strikes without hitting the ball into play.

Notes: All variables are season totals for the team. Stata cannot start a variable name with a number, so *2B* and *3B* are imported as `b` and `v19`; the original names are kept as variable labels. Source: SABR Lahman Baseball Database, 2025 release.

Pitching and Fielding Model

> **From: Pitching Coach**
>
> Our closer[7] says walks don't count because nobody hit the ball. He also says Pesky's Pole is only 302 feet away, so home runs to right field don't count either. Please run a regression and show him he's wrong.

Now that you have identified the batting statistic that explains the runs a team scores, you want to look at the runs that opponents score against your team, so that you can work out the best pitching and fielding statistics. In baseball, *runs* are what you score, and *runs allowed* are what the other team scores against you. Table 4 explains the pitching and fielding variables, and Table 5 shows how to build the statistics from them.

10. A baseball game typically lasts 9 innings for both teams, but if the score is tied after 9 innings, the game goes on to extra innings until one team wins. Game 7 of the 2025 World Series, for example, needed 11 innings. This is similar to extra time in soccer, overtime in American football, or a super over in cricket. Each team bats in an inning

[7] A closer is the relief pitcher a team brings in for the last inning to protect a narrow lead, like the extra defender a soccer coach sends on in the last ten minutes to protect a one-goal lead. Pesky's Pole is the foul pole in right field at Fenway, only 302 feet from home plate.

Table 3: Common batting statistics and how to build them

Statistic	Stata formula	Meaning
Batting average (AVG)	`h/ab`	Share of at bats that end in a hit.
On-base percentage (OBP)	`(h + bb + hbp)/(ab + bb + hbp + sf)`	Share of times at the plate that the batter reaches base, by a hit, a walk or a hit by pitch. Think of it as how often a batter does not get out.
Slugging (SLG)	`(h + doubles + 2*triples + 3*hr)/ab`	Bases per at bat: a single counts 1, a double 2, a triple 3 and a home run 4. Think of it as power, like a batter's strike rate. Why add only 1, 2 and 3 for doubles, triples and home runs? Because `h` has already counted each of them once.
On-base plus slugging (OPS)	`obp + slg`	One number that adds the two skills together with equal weight.
Strikeouts per walk (K/BB)	`so/bb`	How often batters strike out for every walk they draw. Lower is better for a batting team.

Notes: The formulas use the variables in Table 2, after renaming *2B* and *3B* to `doubles` and `triples`.

until the other team gets 3 of its batters out. Generate a variable called `ip`, known as *innings pitched*, from `ipouts` by dividing it by 3. **Hint:** `gen ip = ipouts/3` **[2 points]**

11. Let's create some pitching statistics in line with what we created for batting. Generate *walks per 9 innings* (`bb9`), *home runs per 9 innings* (`hr9`), *earned runs allowed per 9 innings* (`era_calc`) and *strikeouts per 9 innings* (`k9`). Refer to Table 5 for how to generate these four statistics. *Teams.csv* already has a variable called `era`, so name yours `era_calc`. Summarize all four of them along with `era`. Does `era_calc` match `era`? **[3 points]**

12. Let's generate a fielding statistic, the defensive efficiency ratio (`der`). It is basically the share of balls hit into play that fielders turn into outs. It captures information that is independent of pitching and is related to the fielding of the team. We approximate it from the Lahman data as `gen der = (ipouts - soa)/(ipouts - soa + ha - hra)`. **[2 points]**

> ✅ **Checkpoint:** if everything so far is correct, you should see the following.
> Means: `bb9` 3.244, `hr9` 1.100, `k9` 7.527, `der` 0.716, and `era_calc` 4.230, the same as `era`.

13. Similar to `rpg`, let's create a variable (`rapg`) to denote *runs allowed per game*, as the ratio of `ra` and `g`. Regress `rapg` on `bb9`, `hr9` and `k9`, and call this model $\mathcal{M}_4$. Interpret the coefficients. **[3 points]**

14. As you may have noticed, $\mathcal{M}_4$ only captures the pitching statistics, so you may need fielding information in your *runs allowed* model as well. Regress `rapg` on `bb9`, `hr9`,

k9 and der, and call this model $\mathcal{M}_5$. Interpret its coefficients. Compare $\mathcal{M}_4$ and $\mathcal{M}_5$ on the basis of t-statistics, R-squared and adjusted R-squared. Which model do you prefer, and why? **[3 points]**

15. Professional baseball more commonly uses a metric called FIP, or fielding independent pitching, which combines the previous three pitching statistics (bb9, hr9, k9) in a predefined ratio, based on how many runs each pitching outcome allows. Let's generate a variable called fip_raw. It is then put on the scale of earned runs allowed per 9 innings by the following code.

```
* Raw FIP
gen fip_raw = (13*hra + 3*bba - 2*soa)/ip

* Estimating fip: the constant puts FIP on the ERA scale each year
bysort yearid: egen lg_er = total(er)
bysort yearid: egen lg_ip = total(ip)
bysort yearid: egen lg_raw = total(13*hra+ 3*bba - 2*soa)

gen cfp =  9*lg_er/lg_ip - lg_raw/lg_ip
gen fip = fip_raw + cfp

summ fip fip_raw era
```

Regress rapg on fip and der, and call this model $\mathcal{M}_6$. Interpret the coefficient on fip, and compare the R-squared of $\mathcal{M}_6$ and $\mathcal{M}_5$. Is the coefficient on fip in $\mathcal{M}_6$ significantly different from 1? Suppose you decide to go with $\mathcal{M}_6$. Store the coefficient on fip as the scalar p_fip. You will use it in Part 2 to value the pitchers. **Hint:** In Stata you can test this with `test fip = 1` after running the $\mathcal{M}_6$ regression. Store the scalar as in question (9). **[4 points]**

☑ **Checkpoint:** if everything so far is correct, you should see the following.

Mean fip 4.229, close to the mean era (4.230); b_ops = 12.131 and p_fip = 1.006.

$$\mathcal{M}_6: \quad \widehat{\text{rapg}} = \underset{(0.439)}{15.514} + \underset{(0.015)}{1.006}\,\text{fip} - \underset{(0.582)}{21.260}\,\text{der}, \qquad N = 750,\ R^2 = 0.914$$

Standard errors are in parentheses.

16. Run an F-test of the null hypothesis (H_0) that in $\mathcal{M}_5$ $\beta_{hr9} = 13/9$, $\beta_{bb9} = 3/9$ and $\beta_{k9} = -2/9$. These are the weights historically used to compute FIP, so the test asks how well they are supported by our evidence. **Hint:** In Stata, you can run this F-test with `test (hr9 = 13/9) (bb9 = 3/9) (k9 = -2/9)` after running the $\mathcal{M}_5$ regression. **[3 points]**

Table 4: Pitching and fielding variables in *Teams.csv*

Variable	Label in Stata	Description
`ra`	RA	Runs allowed: runs scored by the opponents against the team.
`er`	ER	Earned runs allowed: runs allowed that did not come from a fielding error.
`ipouts`	IPouts	Outs recorded by the team's pitchers and fielders (innings pitched × 3).
`ha`	HA	Hits allowed.
`hra`	HRA	Home runs allowed (like sixes conceded).
`bba`	BBA	Walks allowed (like wides and no-balls).
`soa`	SOA	Strikeouts by the team's pitchers (like bowled or LBW wickets).
`g`	G	Games played.

Notes: All variables are season totals for the team. The pitching variables end in `a` for "allowed" (`hra`, `bba`, `soa`); the same names without the `a` (`hr`, `bb`, `so`) are the team's own batting. Source: SABR Lahman Baseball Database, 2025 release.

Table 5: Common pitching and fielding statistics and how to build them

Statistic	Stata formula	Meaning
Innings pitched (IP)	`ipouts/3`	Three outs make one inning.
Walks per 9 innings (BB9)	`9*bba/ip`	Walks given away per full game. Lower is better.
Home runs per 9 innings (HR9)	`9*hra/ip`	Home runs given up per full game. Lower is better.
Strikeouts per 9 innings (K9)	`9*soa/ip`	Strikeouts per full game. Higher is better.
Earned run average (ERA)	`9*er/ip`	Earned runs allowed per full game. Lower is better. *Teams.csv* already has it as `era`.
Defensive efficiency (DER)	`(ipouts - soa)/(ipouts - soa + ha - hra)`	Share of balls hit into play, and kept in the park, that the fielders turn into outs. Higher is better.

Notes: The formulas use the variables in Table 4. A game is 9 innings, so "per 9 innings" means per full game, like a bowler's economy rate is per over. DER is an approximation of Bill James's Defensive Efficiency Record built from the Lahman data.

2 Moneyball, Part 2: Who Do We Sign?

> **From: Team Owner**
>
> Nice models. But a model can't play first base. I still need 26 players by spring training, and $150 million to spend on them. So, who are we signing? I want to hear Sweet Caroline in October. 🤔



Figure 2: Players from the 2004 World Series champion Red Sox at Fenway Park on Opening Day, April 9, 2024. Photo: Joshua Qualls, Massachusetts Governor's Press Office, Wikimedia Commons, public domain.

In Part 1 you built models for the runs your team scores and the runs it allows. Now you use them to put a price on a win and on each player, and then to pick your team within the $150 million budget. Keep working in the same do-file: you need the 2000–2025 sample and the scalars `b_ops` and `p_fip` from Part 1.

Contribution to Winning

> **From: Team Owner**
>
> The hitting coach and the pitching coach are arguing again about whose job matters more. Settle it. With math, not fists. 🥊

Now that you have models to explain the runs scored by your team (`rpg`) and the runs scored against your team (`rapg`), you want to test how much each of them leads to winning.

17. First, let's create the win percentage (`wpct`) as the ratio of `w` and `g`. Regress `wpct` on `rpg` and `rapg`. Do `rpg` and `rapg` contribute equally to winning? Store the coefficient on `rpg` as the scalar `w_r`. **Hint:** You can test this with the F-test `test rpg = -rapg`. Test at the 5% significance level. Refer to question (9) for how to store a scalar. **[5 points]**

Table 6: Scalars you will store and use later

Scalar	Meaning	Stored in	Used in
`b_ops`	Extra runs per game for a one-unit increase in OPS (model $\mathcal{M}_3$)	Q9	Q24
`p_fip`	Extra runs allowed per game for a one-unit increase in FIP (model $\mathcal{M}_6$)	Q15	Q25
`w_r`	Extra win percentage for one extra run per game	Q17	Q27
`lg_pa`	Plate appearances by a team in an average 2025 game	Q18	Q24
`lg_ops`, `lg_fip`, `lg_cfp`	2025 league averages	Q19	Q23–Q25
`price_win`	Market price of one win, in $ million	Q22	Q28

Notes: Scalars are not part of the data, so they survive `import ..., clear` and `use ..., clear`. They disappear when you close Stata. If you restart Stata halfway through, re-run the questions in the "Stored in" column first.

2025 League Averages

> **From: Data Intern**
>
> I'm storing the league averages as scalars called `lg_*`. I wanted to call them `mediocre_*`, but HR said no.

Before you proceed further, you would like to store the average batting and pitching statistics for 2025, as you will need them as a benchmark when selecting the players.

18. Generate a new variable called plate appearances (`pa`), which counts how many times a team's batters come to bat. It is the sum of `ab`, `bb`, `hbp` and `sf`. It is different from the deliveries faced by a batter: it is just the number of times a batter comes up to the *plate*. Summarize it for 2025, use `r(mean)` to get the average number of plate appearances by a team in a season, and compute the scalar `lg_pa` by dividing it by 162, the number of games in a season, to get the average number of plate appearances in a game. **Hint:** In the following code you have to fill in the XX, XXX, XXY and XXZ. They could be a variable name, `mean` or 2025. **[2 points]**

```
gen pa = XX + bb + XXX + sf
summ pa if yearid == XXY
scalar lg_pa = r(XXZ)/162
display lg_pa
```

19. In a similar way, let's store the average batting and pitching statistics for 2025 with the following code. **Hint:** Fill in the XXX and XXY. They could be `mean` or 2025. **[2 points]**

```
foreach v in obp slg ops bb9 hr9 k9 fip cfp {
    summ `v' if yearid == XXY
    scalar lg_`v' = r(XXX)
    display lg_`v'
}
```

✅ **Checkpoint:** if everything so far is correct, you should see the following.
`w_r` = 0.0954, `lg_pa` = 37.51, `lg_ops` = 0.719, `lg_fip` = 4.160 and `lg_cfp` = 3.269.

Table 7: Data files for the player selection

File	One row per	Variables
`batting_2025.dta`	player (1,470)	`playerid ab h doubles triples hr bb hbp sf`: 2025 batting totals. A player traded during the season has his teams added up.
`pitching_2025.dta`	player (873)	`playerid ipouts ip bb hr so`: walks, home runs and strikeouts **allowed** by the pitcher, like `bba`, `hra` and `soa` in *Teams.csv*.
`salaries_2025.dta`	player (954)	`playerid player team pos salary salary_mn`: 2025 salary in dollars and in $ million, with the player's name, team and position.
`payroll_mn.dta`	team (30)	`team payroll_mn`: total 2025 payroll in $ million, the sum of `salary_mn` over the team's players.

Notes: `playerid` is the Lahman player code, so it links the batting, pitching and salary files. Careful: `bb`, `hr` and `so` mean the team's own batting in *Teams.csv* and in `batting_2025.dta`, but what the pitcher gave up in `pitching_2025.dta`. Sources: SABR Lahman Baseball Database, 2025 release; 2025 salaries from the USA Today salary database.

Market Price of a Win

✉ **From: Team Owner**

The Mets spent $322 million and won 83 games. The Brewers spent $108 million and won 97. Please explain. Slowly. 💸

In this part, we will analyze how much money it usually takes to buy one win in the 2025 season. We will once again start with *Teams.csv*.

20. Table 7 describes the data files you will use from here on. Import *Teams.csv* again as in question (1), and keep the observations for 2025 only. Let's generate a nickname for each team, `team`, so that we can merge it with `payroll_mn.dta`, using the following code.

```
gen team = word(name, wordcount(name))
replace team = "Red Sox" if teamid == "BOS"
replace team = "White Sox" if teamid == "CHA"
replace team = "Blue Jays" if teamid == "TOR"
replace team = "Angels" if teamid == "LAA"
```

Keep only `lgid team teamid w`, and drop the other variables. Merge one-to-one (`1:1`) on `team` using `payroll_mn.dta`, and tabulate `_merge`. Did everything merge? Are there any unmatched observations in the *master* or the *using* data? **Hint:** In Stata, to merge one-to-one on a key variable we run `merge 1:1 key using data2.dta`. Remember to load the master data first. **[4 points]**

21. Sort the data first by league (`lgid`) and then in descending order of `payroll_mn`, to identify the highest- and lowest-paying teams in the *American League* and the *National League* in 2025. **Hint:** To sort data in ascending order of `varA` but descending order of `varB`, we run `gsort varA -varB`, with `-` indicating descending order. **[2 points]**

22. Regress `w` on `payroll_mn`. Store the price of one win as the scalar `price_win = 1/_b[payroll_mn]`. How much did teams usually spend to get one win in 2025? Interpret it. **Hint:** To see the data behind this regression, run `scatter w payroll_mn, mlabel(team)`. Which teams win far more, or far fewer, games than their payroll predicts? **[4 points]**

Building the Team

> **From: Head Scout**
>
> I hear you're picking players with a spreadsheet. I've printed my resignation letter. All 16 pages of it.

Now you have computed all the batting and pitching statistics, and compared your budget of $150 million with what the other teams spent in 2025. Next, you look at the hitters and pitchers. You follow a simple rule: pick the above-average hitters and pitchers, by comparing their hitting and pitching statistics with the league averages you computed in questions (18) and (19).

23. Load `batting_2025.dta`. Compute `pa` as in question (18), and keep only the hitters with `pa >= 400`. Generate `obp`, `slg` and `ops` (Table 3). To compare each hitter with the 2025 season average, compute `ops_gap` as the difference between the player's OPS and the 2025 average. **Hint:** `gen ops_gap = ops - lg_ops`. Remember that `lg_ops` denotes the 2025 season average of `ops`, computed in question (19). **[3 points]**

24. To measure how much better or worse than average a hitter is, we first multiply `ops_gap` by `b_ops` (stored in question (9)), which estimates how many extra runs per team-game, above average, a hitter contributes. To estimate the total impact over a season, we then multiply by the number of team-games of batting a hitter supplies, the ratio `pa/lg_pa`. This is fair, as more expensive players play more and so have higher `pa`.

$$\text{runs} = \underbrace{(\beta_{ops} \times \Delta ops)}_{\text{extra runs per team-game if everyone were him}} \times \underbrace{\frac{pa}{\bar{pa}_{2025}}}_{\text{team-games of batting he supplied}} \tag{1}$$

Assign the player type as `hitter` with `gen type = "hitter"`, and save the data as `hit_value_2025.dta`. **Hint:** In Stata, the above equation can be implemented as `gen runs = (b_ops*ops_gap) * pa/lg_pa`. To save a dta file in Stata, run `save dta_name.dta, replace`. **[4 points]**

From: Team Owner

Section 42, Row 37, Seat 21 in the bleachers is painted red. Ted Williams hit a ball there in 1946, 502 feet from home plate. Find me someone who can do that for less than $10 million.

25. Similarly, load `pitching_2025.dta` and keep only the starting pitchers by filtering on `ip >= 120`. Compute `fip` as in question (15), but with the 2025 constant `lg_cfp`. To estimate the runs a pitcher saves in a season, we compute `lg_fip - fip`, the runs saved per game compared with an average pitcher, multiply it by `p_fip`, and then multiply by `ip/9`, the number of full games he pitched. This makes sense, as pitchers with higher salaries also pitch more.

$$\text{runs} = \underbrace{\beta_{fip} \times \left(\bar{fip}_{2025} - fip\right)}_{\text{extra runs saved per game if everyone were him}} \times \underbrace{\frac{ip}{9}}_{\text{games he pitched}} \tag{2}$$

```
* FIP: raw FIP + the 2025 constant
gen fip = (13*hr + 3*bb - 2*so)/ip + lg_cfp

* Converting FIP into runs saved
gen runs = p_fip * (lg_fip - fip) * ip/9
```

Assign the player type as `pitcher` with `gen type = "pitcher"`, and save the data as `pitch_value_2025.dta`. **Hint:** In `pitching_2025.dta`, `bb`, `hr` and `so` are what the pitcher gave up, like `bba`, `hra` and `soa` in *Teams.csv*. **[4 points]**

Checkpoint: if everything so far is correct, you should see the following.

214 hitters in `hit_value_2025.dta` and 105 pitchers in `pitch_value_2025.dta`.

26. Load `hit_value_2025.dta` and append `pitch_value_2025.dta` to it. Tabulate `type`. Check whether `playerid` is the ID of this dataset. **Hint:** To append data2.dta to data1.dta, first load data1.dta with `use data1.dta, clear`, and then append with `append using data2.dta`. To tabulate, run `tabulate type`. **[2 points]**

27. Generate `wins` from `runs` using `w_r`, stored in question (17), as `gen wins = w_r * runs`. Merge one-to-one on `playerid` with `salaries_2025.dta`. Why are some observations not matched in the *master* or the *using* data? Keep only the observations where `_merge == 3`, and then `drop _merge`. **[4 points]**

Checkpoint: if everything so far is correct, you should see the following.

304 players left after `keep if _merge == 3`.

28. Estimate the player surplus, a concept similar to consumer surplus, as `gen surplus_mn = price_win * wins - salary_mn`. It captures the value of a player to the team minus the cost of the player in the season. Save the data as `player_menu_2025.dta`. **[2 points]**

29. Load `player_menu_2025.dta`. Sort the data by type and in decreasing order of surplus, and rank the players by surplus within each type. **Hint:** First, use `gsort` to sort the data by `type` and `-surplus_mn`. Second, rank within the hitters and the pitchers with `by type: gen rank = _n`, where `_n` counts the observations within each type. **[2 points]**

30. You aim to pick the top 9 hitters and the top 5 pitchers in your ranking. **Hint:** Define a variable called `pick` that is true for either the top 9 hitters OR the top 5 pitchers. In Stata, you can type `gen pick = condition_H | condition_P`, where `|` denotes the *or* condition. You can define `condition_X` as `type == "X" & rank <= Y`, where X is the `type` and Y is how many players of that type you would like to choose. **[2 points]**

31. Summarize `salary_mn` for the players with `pick == 1`, and store the total of `salary_mn` as `total_salary_14`. What is the total salary paid to the 14 players? Which hitters and pitchers have you picked? **Hint:** After `summ salary_mn if pick == 1`, `r(sum)` holds the total salary paid to the 14 players. **[2 points]**

32. However, you are still not done. Oh my God! Still going. A baseball team has 26 players on its roster, and you decide to spend about \$0.76 million, the league minimum salary, on each of the other 12. So what is the total salary cost now, compared with your budget of \$150 million? **Hint:** `scalar total_salary = total_salary_14 + 0.76*12`

    Finally, how many games would your team win? An average team wins 81 of its 162 games. If the other 12 players are average, your team wins 81 plus the total `wins` of your 14 picks. How many is that? The all-time record is 116 wins, set by the 1906 Chicago Cubs and matched by the 2001 Mariners. Is your number believable? If not, why might the model overstate it? **Hint:** `summ wins if pick == 1`, and then `display 81 + r(sum)`. **[6 points]**

> **From: Assistant GM**
>
> Boss, did you see who the second-best value hitter in baseball is? A catcher in Seattle. Can we trade for him? 👀

> **From: Team Owner**
>
> 14 players, and the rest at the minimum. Great. So we win the World Series, right? (We have won it four times since 2004, and Fenway expects a fifth. No pressure.) 🏆