Lab 1 - AI5308/AI4005 (Data Engineering) - 2024 Spring

Made by Mintaek Lim, Revised by Yoonjae, Sungho, Proofreaded by Sundong (Released at 3/12)

This tutorial is aligned with the Homework 1, so please follow with the contents carefully.

1. Implement simple python crawler!
2. Crawl data from https://www.mlb.com/stats/2022 - "MLB hitter stats"
3. Store data in Excel format
4. Conduct simple ML prediction

Homework was done following the example from the tutorial.

Import library and set web driver for Chrome browers

In [1]:
```
!pip install selenium
from selenium import webdriver
from selenium.webdriver.support.select import Select
from selenium.webdriver.common.by import By
import pandas as pd
import time

options = webdriver.ChromeOptions()
options.add_argument('--headless')
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(options=options)
```

```
Collecting selenium
  Downloading selenium-4.18.1-py3-none-any.whl (10.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.0/10.0 MB 21.0 MB/s eta 0:00:00
Requirement already satisfied: urllib3[socks]<3,>=1.26 in /usr/local/lib/python3.10/dist-packages (from selenium) (2.0.7)
Collecting trio~=0.17 (from selenium)
  Downloading trio-0.25.0-py3-none-any.whl (467 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 467.2/467.2 kB 18.4 MB/s eta 0:00:00
Collecting trio-websocket~=0.9 (from selenium)
  Downloading trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.10/dist-packages (from selenium) (2024.2.2)
Requirement already satisfied: typing_extensions>=4.9.0 in /usr/local/lib/python3.10/dist-packages (from selenium) (4.10.0)
Requirement already satisfied: attrs>=23.2.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (3.6)
Collecting outcome (from trio~=0.17->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.2.0)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from urllib3[socks]<3,>=
1.26->selenium) (1.7.1)
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 5.5 MB/s eta 0:00:00
Installing collected packages: outcome, h11, wsproto, trio, trio-websocket, selenium
Successfully installed h11-0.14.0 outcome-1.3.0.post0 selenium-4.18.1 trio-0.25.0 trio-websocket-0.11.1 wsproto-1.2.0
```

MLB 2022

Data crawling

In [2]:
```python
# If the below code is failed or print only a few players, please run the cell again.

## set driver url to crawler

driver.get('https://www.mlb.com/stats/2022')

## find element from web
table = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[1]/div/table')

header_items = table.find_elements(By.XPATH, './thead/tr/th')
```

```
headers = [h.text for h in header_items]

## print properties
print(headers)

## find element from web
body = []
while True:
  time.sleep(2)

  ## crawl from the first page and save it using driver.find_element method
  body_items = table.find_elements(By.XPATH, './tbody/tr')
  for b in body_items:
    row_title = b.find_elements(By.XPATH, './th')
    row_items = b.find_elements(By.XPATH, './td')
    row = [r.text for r in row_title] + [r.text for r in row_items]

    ## since the first element in each column is in the form '1\nShoheiOhtani\nDH',
    ## we tokenize it and make it an individual element.
    row = row[0].split('\n') + row[1:]
    body.append(row)
    print(row)
  try:
    cookie_button = driver.find_element(By.XPATH, '//*[@id="onetrust-accept-btn-handler"]')
    cookie_button.click()
  except:
    pass
  try:
    next_button = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[2]/div/div/div[last()]/butt
    next_button.click()
  except:
    break
```

```
['PLAYER', 'TEAM', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR', 'RBI', 'BB', 'SO', 'SB', 'CS', 'AVG', 'OBP', 'SLG', 'OPS']
['1', 'AaronJudge', 'CF', 'NYY', '157', '570', '133', '177', '28', '0', '62', '131', '111', '175', '16', '3', '.311', '.425', '.
686', '1.111']
['2', 'YordanAlvarez', 'DH', 'HOU', '135', '470', '95', '144', '29', '2', '37', '97', '78', '106', '1', '1', '.306', '.406', '.6
13', '1.019']
['3', 'PaulGoldschmidt', '1B', 'STL', '151', '561', '106', '178', '41', '0', '35', '115', '79', '141', '7', '0', '.317', '.404',
'.578', '.982']
['4', 'JoseAltuve', '2B', 'HOU', '141', '527', '103', '158', '39', '0', '28', '57', '66', '87', '18', '1', '.300', '.387', '.53
3', '.920']
['5', 'FreddieFreeman', '1B', 'LAD', '159', '612', '117', '199', '47', '2', '21', '100', '84', '102', '13', '3', '.325', '.407',
'.511', '.918']
['6', 'MannyMachado', '3B', 'SD', '150', '578', '100', '172', '37', '1', '32', '102', '63', '133', '9', '1', '.298', '.366', '.5
31', '.897']
['7', 'NolanArenado', '3B', 'STL', '148', '557', '73', '163', '42', '1', '30', '103', '52', '72', '5', '3', '.293', '.358', '.53
3', '.891']
['8', 'RafaelDevers', '3B', 'BOS', '141', '555', '84', '164', '42', '1', '27', '88', '50', '114', '3', '1', '.295', '.358', '.52
1', '.879']
['9', 'AustinRiley', '3B', 'ATL', '159', '615', '90', '168', '39', '2', '38', '93', '57', '168', '2', '0', '.273', '.349', '.52
8', '.877']
['10', 'ShoheiOhtani', 'DH', 'LAA', '157', '586', '90', '160', '30', '6', '34', '95', '72', '161', '11', '9', '.273', '.356', '.
519', '.875']
['11', 'MookieBetts', 'RF', 'LAD', '142', '572', '117', '154', '40', '3', '35', '82', '55', '104', '12', '2', '.269', '.340', '.
533', '.873']
['12', 'PeteAlonso', '1B', 'NYM', '160', '597', '95', '162', '27', '0', '40', '131', '67', '128', '5', '1', '.271', '.352', '.51
8', '.870']
['13', 'JoséRamírez', '3B', 'CLE', '157', '601', '90', '168', '44', '5', '29', '126', '69', '82', '20', '7', '.280', '.355', '.5
14', '.869']
['14', 'JulioRodríguez', 'CF', 'SEA', '132', '511', '84', '145', '25', '3', '28', '75', '40', '145', '25', '7', '.284', '.345',
'.509', '.854']
['15', 'JuanSoto', 'RF', 'SD', '153', '524', '93', '127', '25', '2', '27', '62', '135', '96', '6', '2', '.242', '.401', '.452',
'.853']
['16', 'NathanielLowe', '1B', 'TEX', '157', '593', '74', '179', '26', '3', '27', '76', '48', '147', '2', '2', '.302', '.358', '.
492', '.850']
['17', 'AndrésGiménez', '2B', 'CLE', '146', '491', '66', '146', '26', '3', '17', '69', '34', '112', '20', '3', '.297', '.371',
'.466', '.837']
['18', 'JeffMcNeil', '2B', 'NYM', '148', '533', '73', '174', '39', '1', '9', '62', '40', '61', '4', '0', '.326', '.382', '.454',
'.836']
['19', 'XanderBogaerts', 'SS', 'BOS', '150', '557', '84', '171', '38', '0', '15', '73', '57', '118', '8', '2', '.307', '.377',
'.456', '.833']
['19', 'CarlosCorrea', 'SS', 'MIN', '136', '522', '70', '152', '24', '1', '22', '64', '61', '121', '0', '1', '.291', '.366', '.4
67', '.833']
['19', 'TaylorWard', 'RF', 'LAA', '135', '495', '73', '139', '22', '2', '23', '65', '60', '120', '5', '3', '.281', '.360', '.47
3', '.833']
['22', 'KyleSchwarber', 'LF', 'PHI', '155', '577', '100', '126', '21', '3', '46', '94', '86', '200', '10', '1', '.218', '.323',
```

```
'.504', '.827']
['23', 'JoséAbreu', '1B', 'CWS', '157', '601', '85', '183', '40', '0', '15', '75', '62', '110', '0', '0', '.304', '.378', '.44
6', '.824']
['23', 'YandyDíaz', '3B', 'TB', '137', '473', '71', '140', '33', '0', '9', '57', '78', '60', '3', '3', '.296', '.401', '.423',
'.824']
['25', 'AlexBregman', '3B', 'HOU', '155', '548', '93', '142', '38', '0', '23', '93', '87', '77', '1', '2', '.259', '.366', '.45
4', '.820']
['25', 'J.T.Realmuto', 'C', 'PHI', '139', '504', '75', '139', '26', '5', '22', '84', '41', '119', '21', '1', '.276', '.342', '.4
78', '.820']
['27', 'VladimirGuerrero', '1B', 'TOR', '160', '638', '90', '175', '35', '0', '32', '97', '58', '116', '8', '3', '.274', '.339',
'.480', '.819']
['28', 'AnthonyRizzo', '1B', 'NYY', '130', '465', '77', '104', '21', '1', '32', '75', '58', '101', '6', '5', '.224', '.338', '.4
80', '.818']
['29', 'StarlingMarte', 'RF', 'NYM', '118', '466', '76', '136', '24', '5', '16', '63', '26', '97', '18', '9', '.292', '.347', '.
468', '.815']
['30', 'GeorgeSpringer', 'CF', 'TOR', '133', '513', '89', '137', '22', '4', '25', '76', '54', '100', '14', '2', '.267', '.342',
'.472', '.814']
['31', 'BrandonDrury', '1B', 'SD', '138', '518', '87', '136', '31', '2', '28', '87', '38', '126', '2', '3', '.263', '.320', '.49
2', '.812']
['32', 'TreaTurner', 'SS', 'LAD', '160', '652', '101', '194', '39', '4', '21', '100', '45', '131', '27', '3', '.298', '.343', '.
466', '.809']
['33', 'WillSmith', 'C', 'LAD', '137', '508', '68', '132', '26', '3', '24', '87', '56', '96', '1', '0', '.260', '.343', '.465',
'.808']
['33', 'KyleTucker', 'RF', 'HOU', '150', '544', '71', '140', '28', '1', '30', '107', '59', '95', '25', '4', '.257', '.330', '.47
8', '.808']
['35', 'TeoscarHernández', 'RF', 'TOR', '131', '499', '71', '133', '35', '1', '25', '77', '34', '152', '6', '3', '.267', '.316',
'.491', '.807']
['35', 'HunterRenfroe', 'RF', 'MIL', '125', '474', '62', '121', '23', '1', '29', '72', '39', '121', '1', '1', '.255', '.315', '.
492', '.807']
['37', 'BryanReynolds', 'CF', 'PIT', '145', '542', '74', '142', '19', '4', '27', '62', '56', '141', '7', '3', '.262', '.345', '.
461', '.806']
['38', 'ChristianWalker', '1B', 'AZ', '160', '583', '84', '141', '25', '2', '36', '94', '69', '131', '2', '2', '.242', '.327',
'.477', '.804']
['39', 'BoBichette', 'SS', 'TOR', '159', '652', '91', '189', '43', '1', '24', '93', '41', '155', '13', '8', '.290', '.333', '.46
9', '.802']
['39', 'MattOlson', '1B', 'ATL', '162', '616', '86', '148', '44', '0', '34', '103', '75', '170', '0', '0', '.240', '.325', '.47
7', '.802']
['41', 'BrandonNimmo', 'CF', 'NYM', '151', '580', '102', '159', '30', '7', '16', '64', '71', '116', '3', '2', '.274', '.367', '.
433', '.800']
['42', 'LuisArraez', '1B', 'MIN', '144', '547', '88', '173', '31', '1', '8', '49', '50', '43', '4', '4', '.316', '.375', '.420',
'.795']
['43', 'RhysHoskins', '1B', 'PHI', '156', '589', '81', '145', '33', '2', '30', '79', '72', '169', '2', '1', '.246', '.332', '.46
2', '.794']
['44', 'EugenioSuárez', '3B', 'SEA', '150', '543', '76', '128', '24', '2', '31', '87', '73', '196', '0', '0', '.236', '.332', '.
```

```
459', '.791']
['45', 'J.D.Martinez', 'DH', 'BOS', '139', '533', '76', '146', '43', '1', '16', '62', '52', '145', '0', '0', '.274', '.341', '.4
48', '.789']
['46', 'FranciscoLindor', 'SS', 'NYM', '161', '630', '98', '170', '25', '5', '26', '107', '59', '133', '16', '6', '.270', '.33
9', '.449', '.788']
['46', 'JustinTurner', '3B', 'LAD', '128', '468', '61', '130', '36', '0', '13', '81', '50', '89', '3', '0', '.278', '.350', '.43
8', '.788']
['48', 'AlejandroKirk', 'C', 'TOR', '139', '470', '59', '134', '19', '0', '14', '63', '63', '58', '0', '0', '.285', '.372', '.41
5', '.787']
['49', 'JoshBell', 'DH', 'SD', '156', '552', '78', '147', '29', '3', '17', '71', '81', '102', '0', '1', '.266', '.362', '.422',
'.784']
['50', 'C.J.Cron', '1B', 'COL', '150', '575', '79', '148', '28', '3', '29', '102', '43', '164', '0', '0', '.257', '.315', '.46
8', '.783']
['51', 'IanHapp', 'LF', 'CHC', '158', '573', '72', '155', '42', '2', '17', '72', '58', '149', '9', '4', '.271', '.342', '.440',
'.782']
['52', 'DansbySwanson', 'SS', 'ATL', '162', '640', '99', '177', '32', '1', '25', '96', '49', '182', '18', '7', '.277', '.329',
'.447', '.776']
['53', 'TyFrance', '1B', 'SEA', '140', '551', '65', '151', '27', '1', '20', '83', '35', '94', '0', '0', '.274', '.338', '.436',
'.774']
['54', 'StevenKwan', 'LF', 'CLE', '147', '563', '89', '168', '25', '7', '6', '52', '62', '60', '19', '5', '.298', '.373', '.40
0', '.773']
['54', 'AnthonySantander', 'RF', 'BAL', '152', '574', '78', '138', '24', '0', '33', '89', '55', '122', '0', '2', '.240', '.318',
'.455', '.773']
['56', 'RandyArozarena', 'LF', 'TB', '153', '586', '72', '154', '41', '3', '20', '89', '46', '156', '32', '12', '.263', '.327',
'.445', '.772']
['56', 'AndrewBenintendi', 'LF', 'NYY', '126', '461', '54', '140', '23', '3', '5', '51', '52', '77', '8', '3', '.304', '.373',
'.399', '.772']
['56', 'CoreySeager', 'SS', 'TEX', '151', '593', '91', '145', '24', '1', '33', '83', '58', '103', '3', '0', '.245', '.317', '.45
5', '.772']
['59', 'MarkCanha', 'LF', 'NYM', '140', '462', '71', '123', '24', '0', '13', '61', '48', '97', '3', '1', '.266', '.367', '.403',
'.770']
['60', 'RowdyTellez', '1B', 'MIL', '153', '529', '67', '116', '23', '0', '35', '89', '62', '121', '2', '1', '.219', '.306', '.46
1', '.767']
['60', 'GioUrshela', '3B', 'MIN', '144', '501', '61', '143', '27', '3', '13', '64', '41', '96', '1', '0', '.285', '.338', '.42
9', '.767']
['62', 'RonaldAcuña', 'RF', 'ATL', '119', '467', '71', '124', '24', '0', '15', '50', '53', '126', '29', '11', '.266', '.351', '.
413', '.764']
['63', 'GleyberTorres', '2B', 'NYY', '140', '526', '73', '135', '28', '1', '24', '76', '39', '129', '10', '5', '.257', '.310',
'.451', '.761']
['64', 'SeanMurphy', 'C', 'OAK', '148', '537', '67', '134', '37', '2', '18', '66', '56', '124', '1', '0', '.250', '.332', '.42
6', '.758']
['65', 'MattChapman', '3B', 'TOR', '155', '538', '83', '123', '27', '1', '27', '76', '68', '170', '2', '2', '.229', '.324', '.43
3', '.757']
['66', 'WillyAdames', 'SS', 'MIL', '139', '563', '83', '134', '31', '0', '31', '98', '49', '166', '8', '3', '.238', '.298', '.45
```

```
8', '.756']
['66', 'AdolisGarcía', 'RF', 'TEX', '156', '605', '88', '151', '34', '5', '27', '101', '40', '183', '25', '6', '.250', '.300',
'.456', '.756']
['68', 'AndrewVaughn', 'RF', 'CWS', '134', '510', '60', '138', '28', '1', '17', '76', '31', '96', '0', '0', '.271', '.321', '.42
9', '.750']
['69', 'SethBrown', '1B', 'OAK', '150', '500', '55', '115', '26', '3', '25', '73', '51', '146', '11', '2', '.230', '.305', '.44
4', '.749']
['70', 'DaultonVarsho', 'RF', 'AZ', '151', '531', '79', '125', '23', '3', '27', '74', '46', '145', '16', '6', '.235', '.302', '.
443', '.745']
['71', 'RyanMcMahon', '3B', 'COL', '153', '529', '67', '130', '23', '3', '20', '67', '60', '158', '7', '3', '.246', '.327', '.41
4', '.741']
['72', 'JoshRojas', '3B', 'AZ', '125', '443', '66', '119', '25', '1', '9', '56', '55', '98', '23', '3', '.269', '.349', '.391',
'.740']
['73', 'ChristianYelich', 'LF', 'MIL', '154', '575', '99', '145', '25', '4', '14', '57', '88', '162', '19', '3', '.252', '.355',
'.383', '.738']
['74', 'NicoHoerner', 'SS', 'CHC', '135', '481', '60', '135', '22', '5', '10', '55', '28', '57', '20', '2', '.281', '.327', '.41
0', '.737']
['75', 'DJLeMahieu', '3B', 'NYY', '125', '467', '74', '122', '18', '0', '12', '46', '67', '71', '4', '3', '.261', '.357', '.37
7', '.734']
['76', 'CharlieBlackmon', 'DH', 'COL', '135', '530', '60', '140', '22', '6', '16', '78', '32', '109', '4', '1', '.264', '.314',
'.419', '.733']
['76', 'BrendanRodgers', '2B', 'COL', '137', '527', '72', '140', '30', '3', '13', '63', '46', '101', '0', '0', '.266', '.325',
'.408', '.733']
['76', 'MarcusSemien', '2B', 'TEX', '161', '657', '101', '163', '31', '5', '26', '83', '53', '120', '25', '8', '.248', '.304',
'.429', '.733']
['76', 'AlexVerdugo', 'LF', 'BOS', '152', '593', '75', '166', '39', '1', '11', '74', '42', '86', '1', '3', '.280', '.328', '.40
5', '.733']
['80', 'KetelMarte', '2B', 'AZ', '137', '492', '68', '118', '42', '2', '12', '52', '55', '101', '5', '1', '.240', '.321', '.40
7', '.728']
['80', 'RyanMountcastle', '1B', 'BAL', '145', '555', '62', '139', '28', '1', '22', '85', '43', '154', '4', '1', '.250', '.305',
'.423', '.728']
['82', 'EduardoEscobar', '3B', 'NYM', '136', '495', '58', '119', '26', '4', '20', '69', '40', '129', '0', '2', '.240', '.295',
'.430', '.725']
['83', 'TommyEdman', '2B', 'STL', '153', '577', '95', '153', '31', '4', '13', '57', '46', '111', '32', '3', '.265', '.324', '.40
0', '.724']
['83', 'RandalGrichuk', 'RF', 'COL', '141', '506', '60', '131', '21', '3', '19', '73', '24', '127', '4', '0', '.259', '.299', '.
425', '.724']
['83', 'PatrickWisdom', '3B', 'CHC', '134', '469', '67', '97', '28', '0', '25', '66', '53', '183', '8', '4', '.207', '.298', '.4
26', '.724']
['86', 'LuisRengifo', '2B', 'LAA', '127', '489', '45', '129', '22', '4', '17', '52', '17', '79', '6', '2', '.264', '.294', '.42
9', '.723']
['87', 'JakeCronenworth', '2B', 'SD', '158', '587', '88', '140', '30', '4', '17', '88', '70', '131', '3', '0', '.239', '.332',
'.390', '.722']
['87', 'ThairoEstrada', '2B', 'SF', '140', '488', '71', '127', '22', '2', '14', '62', '33', '89', '21', '6', '.260', '.322', '.4
```

```
00', '.722']
['87', 'JuricksonProfar', 'LF', 'SD', '152', '575', '82', '140', '36', '2', '15', '58', '73', '103', '5', '1', '.243', '.331',
'.391', '.722']
['87', 'BobbyWitt', 'SS', 'KC', '150', '591', '82', '150', '31', '6', '20', '80', '30', '135', '30', '7', '.254', '.294', '.42
8', '.722']
['91', 'CedricMullins', 'CF', 'BAL', '156', '608', '89', '157', '32', '4', '16', '64', '47', '126', '34', '10', '.258', '.318',
'.403', '.721']
['92', 'AustinHays', 'LF', 'BAL', '145', '535', '66', '134', '35', '2', '16', '60', '34', '114', '2', '4', '.250', '.306', '.41
3', '.719']
['93', 'JeremyPeña', 'SS', 'HOU', '136', '521', '72', '132', '20', '2', '22', '63', '22', '135', '11', '2', '.253', '.289', '.42
6', '.715']
['93', 'AmedRosario', 'SS', 'CLE', '153', '637', '86', '180', '26', '9', '11', '71', '25', '111', '18', '4', '.283', '.312', '.4
03', '.715']
['95', 'AlecBohm', '3B', 'PHI', '152', '586', '79', '164', '24', '3', '13', '72', '31', '110', '2', '3', '.280', '.315', '.398',
'.713']
['95', 'MaxMuncy', '3B', 'LAD', '136', '464', '69', '91', '22', '1', '21', '69', '90', '141', '2', '0', '.196', '.329', '.384',
'.713']
['97', 'WilmerFlores', '2B', 'SF', '151', '525', '72', '120', '28', '1', '19', '71', '59', '103', '0', '0', '.229', '.316', '.39
4', '.710']
['97', 'TreyMancini', 'DH', 'HOU', '143', '519', '56', '124', '23', '1', '18', '63', '53', '135', '0', '0', '.239', '.319', '.39
1', '.710']
['97', 'LukeVoit', 'DH', 'WSH', '135', '500', '55', '113', '22', '0', '22', '69', '55', '179', '1', '1', '.226', '.308', '.402',
'.710']
['100', 'Ha-SeongKim', 'SS', 'SD', '150', '517', '58', '130', '29', '3', '11', '59', '51', '100', '12', '2', '.251', '.325', '.3
83', '.708']
['101', 'ElvisAndrus', 'SS', 'CWS', '149', '535', '66', '133', '32', '0', '17', '58', '39', '92', '18', '4', '.249', '.303', '.4
04', '.707']
['102', 'MJMelendez', 'C', 'KC', '129', '460', '57', '100', '21', '3', '18', '62', '66', '131', '2', '3', '.217', '.313', '.39
3', '.706']
['103', 'LaneThomas', 'LF', 'WSH', '146', '498', '62', '120', '26', '2', '17', '52', '41', '132', '8', '4', '.241', '.301', '.40
4', '.705']
['104', 'KyleFarmer', 'SS', 'CIN', '145', '526', '58', '134', '25', '1', '14', '78', '33', '99', '4', '3', '.255', '.315', '.38
6', '.701']
['105', 'AndrewMcCutchen', 'DH', 'MIL', '134', '515', '66', '122', '25', '0', '17', '69', '57', '124', '8', '6', '.237', '.316',
'.384', '.700']
['106', 'MikeYastrzemski', 'RF', 'SF', '148', '485', '73', '104', '31', '2', '17', '57', '61', '141', '5', '1', '.214', '.305',
'.392', '.697']
['107', 'NickCastellanos', 'RF', 'PHI', '136', '524', '56', '138', '27', '0', '13', '62', '29', '130', '7', '1', '.263', '.305',
'.389', '.694']
['108', 'CarlosSantana', 'DH', 'SEA', '131', '431', '52', '87', '18', '0', '19', '60', '71', '88', '0', '0', '.202', '.316', '.3
76', '.692']
['109', 'JesseWinker', 'LF', 'SEA', '136', '456', '51', '100', '15', '0', '14', '53', '84', '103', '0', '0', '.219', '.344', '.3
44', '.688']
['110', 'MarcellOzuna', 'DH', 'ATL', '124', '470', '56', '106', '19', '0', '23', '56', '31', '122', '2', '1', '.226', '.274', '.
```

```
413', '.687']
['111', 'TommyPham', 'LF', 'BOS', '144', '554', '89', '131', '23', '1', '17', '63', '56', '167', '8', '3', '.236', '.312', '.37
4', '.686']
['112', 'JoshDonaldson', '3B', 'NYY', '132', '478', '59', '106', '28', '0', '15', '62', '54', '148', '2', '2', '.222', '.308',
'.374', '.682']
['113', 'AJPollock', 'LF', 'CWS', '138', '489', '61', '120', '26', '1', '14', '56', '32', '98', '3', '1', '.245', '.292', '.38
9', '.681']
['114', 'J.P.Crawford', 'SS', 'SEA', '145', '518', '57', '126', '24', '3', '6', '42', '68', '80', '3', '2', '.243', '.339', '.33
6', '.675']
['115', 'WhitMerrifield', '2B', 'TOR', '139', '504', '70', '126', '28', '1', '11', '58', '38', '85', '16', '5', '.250', '.298',
'.375', '.673']
['116', 'JavierBáez', 'SS', 'DET', '144', '555', '64', '132', '27', '4', '17', '67', '26', '147', '9', '2', '.238', '.278', '.39
3', '.671']
['117', 'JesúsAguilar', 'DH', 'BAL', '129', '464', '39', '109', '19', '0', '16', '51', '28', '119', '1', '0', '.235', '.281', '.
379', '.660']
['118', "Ke'BryanHayes", '3B', 'PIT', '136', '505', '55', '123', '24', '3', '7', '41', '48', '122', '20', '5', '.244', '.314',
'.345', '.659']
['119', 'CodyBellinger', 'CF', 'LAD', '144', '504', '70', '106', '27', '3', '19', '68', '38', '150', '14', '3', '.210', '.265',
'.389', '.654']
['120', 'NelsonCruz', 'DH', 'WSH', '124', '448', '50', '105', '16', '0', '10', '64', '49', '119', '4', '0', '.234', '.313', '.33
7', '.650']
['121', 'YuliGurriel', '1B', 'HOU', '146', '545', '53', '132', '40', '0', '8', '53', '30', '73', '8', '0', '.242', '.288', '.36
0', '.648']
['122', 'JorgeMateo', 'SS', 'BAL', '150', '494', '63', '109', '25', '7', '13', '50', '27', '147', '35', '9', '.221', '.267', '.3
79', '.646']
['123', 'TonyKemp', '2B', 'OAK', '147', '497', '61', '117', '24', '2', '7', '46', '45', '69', '11', '1', '.235', '.307', '.334',
'.641']
['123', 'IsiahKiner-Falefa', 'SS', 'NYY', '142', '483', '66', '126', '20', '0', '4', '48', '35', '72', '22', '4', '.261', '.31
4', '.327', '.641']
['125', 'CésarHernández', '2B', 'WSH', '147', '560', '64', '139', '28', '4', '1', '34', '45', '114', '10', '4', '.248', '.311',
'.318', '.629']
['126', 'TrentGrisham', 'CF', 'SD', '152', '451', '58', '83', '16', '2', '17', '53', '57', '150', '7', '1', '.184', '.284', '.34
1', '.625']
['127', 'AdamFrazier', '2B', 'SEA', '156', '541', '61', '129', '22', '4', '3', '42', '46', '73', '11', '6', '.238', '.301', '.31
1', '.612']
['128', 'MiguelRojas', 'SS', 'MIA', '140', '471', '34', '111', '19', '2', '6', '36', '26', '61', '9', '3', '.236', '.283', '.32
3', '.606']
['129', 'MylesStraw', 'CF', 'CLE', '152', '535', '72', '118', '22', '3', '0', '32', '54', '87', '21', '1', '.221', '.291', '.27
3', '.564']
['130', 'JonathanSchoop', '2B', 'DET', '131', '481', '48', '97', '23', '1', '11', '38', '19', '107', '5', '0', '.202', '.239',
'.322', '.561']
```

In [3]:
```
## adding two more element in headers
headers.insert(0, 'Rank')
```

```
headers.insert(2, 'Position')

## Create a dataframe with 'headers' as a column and 'body' as the content.
df_mlb = pd.DataFrame(body)
df_mlb.columns = headers
df_mlb = df_mlb.set_index('Rank')

## final data frame (expected size should be 130, 19 - If smaller, run the previous cell again)
df_mlb
```

Out[3]:

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
1	AaronJudge	CF	NYY	157	570	133	177	28	0	62	131	111	175	16	3	.311	.425	.686	1.111
2	YordanAlvarez	DH	HOU	135	470	95	144	29	2	37	97	78	106	1	1	.306	.406	.613	1.019
3	PaulGoldschmidt	1B	STL	151	561	106	178	41	0	35	115	79	141	7	0	.317	.404	.578	.982
4	JoseAltuve	2B	HOU	141	527	103	158	39	0	28	57	66	87	18	1	.300	.387	.533	.920
5	FreddieFreeman	1B	LAD	159	612	117	199	47	2	21	100	84	102	13	3	.325	.407	.511	.918
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
126	TrentGrisham	CF	SD	152	451	58	83	16	2	17	53	57	150	7	1	.184	.284	.341	.625
127	AdamFrazier	2B	SEA	156	541	61	129	22	4	3	42	46	73	11	6	.238	.301	.311	.612
128	MiguelRojas	SS	MIA	140	471	34	111	19	2	6	36	26	61	9	3	.236	.283	.323	.606
129	MylesStraw	CF	CLE	152	535	72	118	22	3	0	32	54	87	21	1	.221	.291	.273	.564
130	JonathanSchoop	2B	DET	131	481	48	97	23	1	11	38	19	107	5	0	.202	.239	.322	.561

130 rows × 19 columns

In [4]:
```
## conneting google drive to colab
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [5]:
```
## save data frame as xlsx file
df_mlb = df_mlb.to_excel('/content/sample_data/MLB2022.xlsx', index=False)
```

ML prediction

- Using the crawled data, we will run a machine learning task
- For simple tutorial, we'll see how to predict the number of homeruns from other stats

In [6]:
```
df_mlb = pd.read_excel('/content/sample_data/MLB2022.xlsx')
X_df = df_mlb.drop('HR', axis=1) ## delete HR data (it will be answer) from data frame
X_df = X_df.drop(['PLAYER', 'Position', 'TEAM'], axis=1) ## delete non-floating number data
X_df
```

Out[6]:

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
0	157	570	133	177	28	0	131	111	175	16	3	0.311	0.425	0.686	1.111
1	135	470	95	144	29	2	97	78	106	1	1	0.306	0.406	0.613	1.019
2	151	561	106	178	41	0	115	79	141	7	0	0.317	0.404	0.578	0.982
3	141	527	103	158	39	0	57	66	87	18	1	0.300	0.387	0.533	0.920
4	159	612	117	199	47	2	100	84	102	13	3	0.325	0.407	0.511	0.918
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
125	152	451	58	83	16	2	53	57	150	7	1	0.184	0.284	0.341	0.625
126	156	541	61	129	22	4	42	46	73	11	6	0.238	0.301	0.311	0.612
127	140	471	34	111	19	2	36	26	61	9	3	0.236	0.283	0.323	0.606
128	152	535	72	118	22	3	32	54	87	21	1	0.221	0.291	0.273	0.564
129	131	481	48	97	23	1	38	19	107	5	0	0.202	0.239	0.322	0.561

130 rows × 15 columns

In [7]:
```
y_df = df_mlb['HR'] ## setting HR data as y (target of prediction)
y_df
```

Out[7]:
```
0      62
1      37
2      35
3      28
4      21
       ..
125    17
126     3
127     6
128     0
129    11
Name: HR, Length: 130, dtype: int64
```

In [8]:
```
from sklearn.model_selection import train_test_split
## define train data and test data
X_train, X_test, y_train, y_test = train_test_split(X_df,y_df, test_size=0.2, random_state=0xc0ffee)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(104, 15) (26, 15) (104,) (26,)
```

In [9]:
```
from sklearn.preprocessing import StandardScaler
## scale the data
scalr = StandardScaler()
X_train = scalr.fit_transform(X_train)
X_test = scalr.transform(X_test)
```

In [10]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
```

In [11]:
```
## train Linear Regression and Ridge Regression model and check the errors
reg1  = LinearRegression()
reg1.fit(X_train, y_train)
pred_train1 = reg1.predict(X_train)
pred_val1 = reg1.predict(X_test)
mse_train1 = mean_squared_error(y_train, pred_train1)
mse_val1 = mean_squared_error(y_test, pred_val1)

reg2 = Ridge()
reg2.fit(X_train, y_train)
pred_train2 = reg2.predict(X_train)
pred_val2 = reg2.predict(X_test)
mse_train2 = mean_squared_error(y_train, pred_train2)
```

```
mse_val2 = mean_squared_error(y_test, pred_val2)

print("1. Linear Regresion\t, train = %.4f, val = %.4f" %(mse_train1, mse_val1))
print("2. Ridge\t\t, train = %.4f, val = %.4f" %(mse_train2, mse_val2))
```

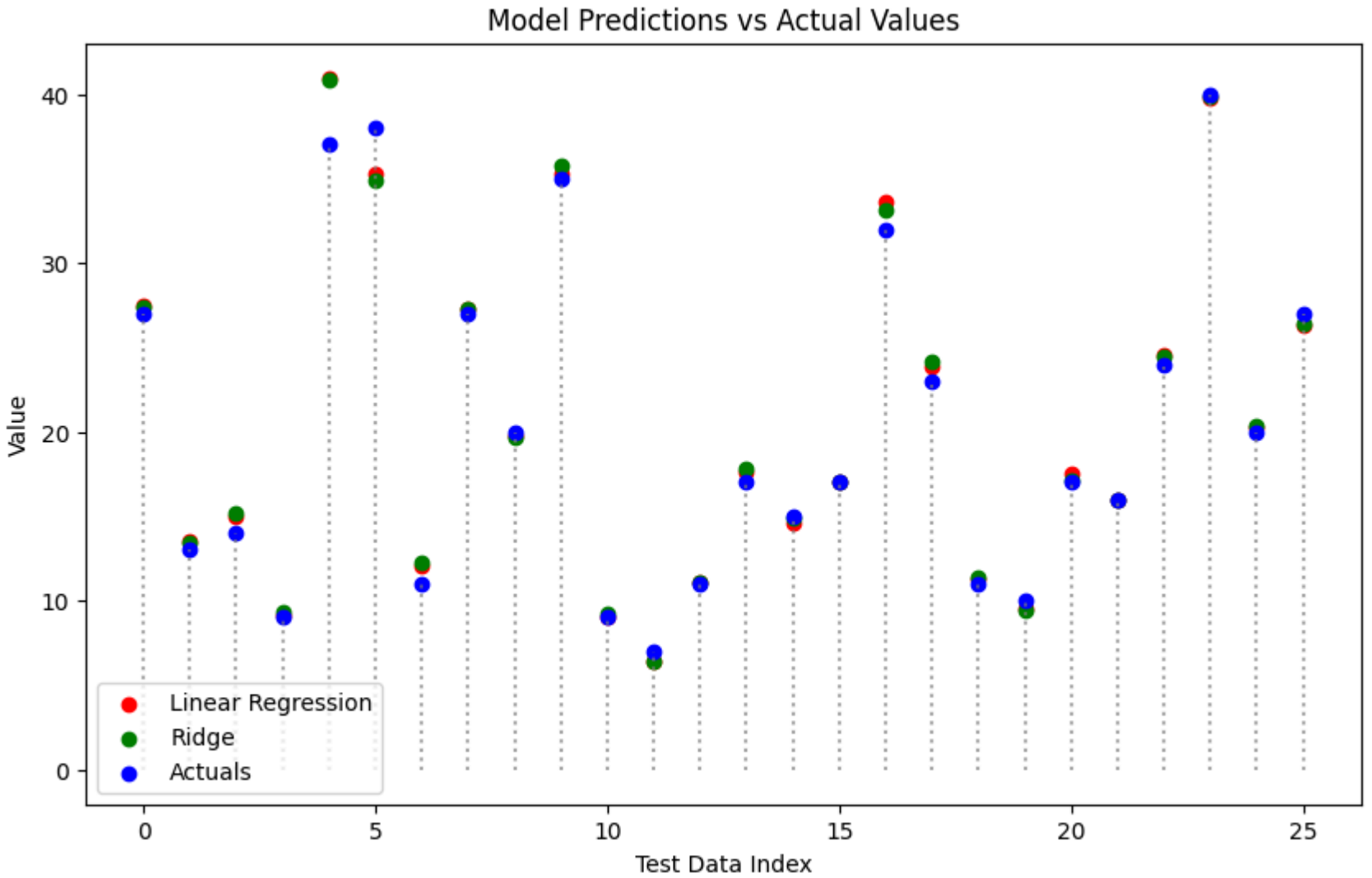
```
1. Linear Regresion     , train = 0.3675, val = 1.2218
2. Ridge                , train = 0.4277, val = 1.2663
```

Data Visualization

Here is the sample code to visualize the prediction results.

In [12]:

```
import matplotlib.pyplot as plt

y_test =  y_test.astype(int).values

# Drawing scatter plots
plt.figure(figsize=(10, 6))
plt.scatter(range(len(pred_val1)), pred_val1, color='r', label='Linear Regression')
plt.scatter(range(len(pred_val2)), pred_val2, color='g', label='Ridge')
plt.scatter(range(len(y_test)), y_test, color='b', label='Actuals')

for i in range(len(y_test)):
    plt.vlines(x=i, ymin=0, ymax=y_test[i], colors='gray', linestyles='dotted', alpha=0.7)

plt.xlabel('Test Data Index')
plt.ylabel('Value')
plt.title('Model Predictions vs Actual Values')
plt.legend()

plt.show()
```



In [13]:
```
# compare this with [MLB 2022 home-run](https://www.mlb.com/stats/home-runs/2022).
df_mlb['HR'] = df_mlb['HR'].astype('int')
df_mlb[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)

#df_mlb1['HR'] = df_mlb1['HR'].astype('int')
#df_mlb1[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
```

Out[13]:

	PLAYER	HR
0	AaronJudge	62
1	KyleSchwarber	46
2	PeteAlonso	40
3	AustinRiley	38
4	YordanAlvarez	37
5	ChristianWalker	36
6	RowdyTellez	35
7	MookieBetts	35
8	PaulGoldschmidt	35
9	ShoheiOhtani	34
10	MattOlson	34
11	CoreySeager	33
12	AnthonySantander	33
13	MannyMachado	32
14	AnthonyRizzo	32
15	VladimirGuerrero	32
16	EugenioSuárez	31
17	WillyAdames	31
18	KyleTucker	30
19	NolanArenado	30

Question for HW1:

Design an ML system to predict the top 20 home run hitters in the MLB 2024 season. The system should predict results similar to the table below. Note that the 2024 MLB season is ongoing and will last until Sep 29, 2024, which means you should devise a better prediction scheme.

It is strongly recommended to use Google Colab (Feel free to copy this page and use it on your own purpose). Include your thoughts alongside your code, using Markdown for formatting. You will submit both the shareable link and a PDF of your notebook after running all your code. (See how to export a Colab notebook to PDF)

Data crawling for current time(Spring Training time).

- The final day of Spring Training is March 26.

In [14]:
```
# If the below code is failed or print only a few players, please run the cell again.

## set driver url to crawler
driver.get('https://www.mlb.com/stats/')

## find element from web
table1 = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[1]/div/table')

header_items = table1.find_elements(By.XPATH, './thead/tr/th')
headers1 = [h.text for h in header_items]

## print properties
print(headers1)

## find element from web
body1 = []
while True:
  time.sleep(2)

  ## crawl from the first page and save it using driver.find_element method
  body_items = table1.find_elements(By.XPATH, './tbody/tr')
  for b in body_items:
    row_title = b.find_elements(By.XPATH, './th')
    row_items = b.find_elements(By.XPATH, './td')
    row = [r.text for r in row_title] + [r.text for r in row_items]

    ## since the first element in each column is in the form '1\nShoheiOhtani\nDH',
    ## we tokenize it and make it an individual element.
    row = row[0].split('\n') + row[1:]
    body1.append(row)
    print(row)
```

```
try:
  next_button = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[2]/div/div/div[2]/button')
  next_button.click()
  #    //*[@id="stats-app-root"]/section/section/div[3]/div[2]/div/div/div[2]/button
except:
  break

try:
  cookie_button = driver.find_element(By.XPATH, '//*[@id="onetrust-accept-btn-handler"]')
  cookie_button.click()
except:
  pass
```

```
['PLAYER', 'TEAM', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR', 'RBI', 'BB', 'SO', 'SB', 'CS', 'AVG', 'OBP', 'SLG', 'OPS']
['1', 'JamesWood', 'RF', 'WSH', '22', '44', '13', '16', '1', '1', '4', '7', '11', '13', '3', '1', '.364', '.509', '.705', '1.21
4']
['2', 'WyattLangford', 'LF', 'TEX', '19', '56', '13', '21', '2', '0', '6', '20', '5', '15', '0', '0', '.375', '.429', '.732',
'1.161']
['3', 'ChristianEncarnacion-Strand', '1B', 'CIN', '15', '48', '13', '16', '2', '3', '4', '12', '3', '6', '0', '0', '.333', '.38
5', '.750', '1.135']
['4', 'MookieBetts', '2B', 'LAD', '13', '34', '10', '15', '3', '0', '1', '3', '5', '6', '0', '1', '.441', '.513', '.618', '1.13
1']
['5', 'LuisMatos', 'CF', 'SF', '20', '51', '14', '16', '6', '0', '4', '11', '3', '8', '1', '1', '.314', '.375', '.667', '1.042']
['6', 'AlexBregman', '3B', 'HOU', '16', '47', '8', '18', '4', '0', '2', '8', '4', '7', '0', '0', '.383', '.431', '.596', '1.02
7']
['7', 'CobyMayo', '3B', 'BAL', '23', '50', '7', '18', '7', '0', '1', '11', '6', '12', '1', '0', '.360', '.448', '.560', '1.008']
['8', 'TreyLipscomb', '2B', 'WSH', '20', '48', '6', '19', '2', '1', '1', '7', '5', '7', '1', '2', '.396', '.453', '.542', '.99
5']
['8', 'JacksonMerrill', 'CF', 'SD', '13', '37', '8', '13', '3', '0', '2', '6', '3', '3', '2', '0', '.351', '.400', '.595', '.99
5']
['10', 'SpencerSteer', 'LF', 'CIN', '16', '47', '8', '15', '3', '1', '3', '9', '5', '10', '3', '0', '.319', '.377', '.617', '.99
4']
['11', 'MichaelToglia', '1B', 'COL', '20', '48', '9', '13', '3', '0', '5', '13', '5', '18', '1', '0', '.271', '.340', '.646', '.
986']
['12', 'AlecBurleson', 'LF', 'STL', '18', '43', '3', '16', '4', '0', '1', '7', '6', '3', '1', '0', '.372', '.440', '.535', '.97
5']
['13', 'MarcusSemien', '2B', 'TEX', '19', '43', '9', '11', '3', '0', '3', '7', '11', '10', '0', '0', '.256', '.429', '.535', '.9
64']
['14', 'FreddieFreeman', '1B', 'LAD', '13', '32', '6', '9', '0', '1', '3', '14', '2', '7', '0', '0', '.281', '.324', '.625', '.9
49']
['15', 'PeteAlonso', '1B', 'NYM', '17', '46', '4', '14', '5', '0', '2', '4', '7', '11', '0', '0', '.304', '.396', '.543', '.93
9']
['16', 'ZackGelof', '2B', 'OAK', '18', '51', '13', '15', '3', '0', '4', '8', '4', '19', '0', '1', '.294', '.345', '.588', '.93
3']
['17', 'LaneThomas', 'RF', 'WSH', '19', '48', '12', '15', '0', '0', '3', '6', '9', '11', '5', '1', '.313', '.424', '.500', '.92
4']
['18', 'EllyDe La Cruz', 'SS', 'CIN', '16', '47', '13', '13', '3', '2', '2', '6', '7', '20', '5', '1', '.277', '.370', '.553',
'.923']
['19', 'JoséFermín', '3B', 'STL', '22', '43', '9', '13', '3', '0', '2', '4', '6', '5', '1', '0', '.302', '.400', '.512', '.912']
['20', 'RichiePalacios', 'LF', 'TB', '18', '48', '8', '15', '2', '0', '3', '7', '4', '10', '2', '1', '.313', '.358', '.542', '.9
00']
['21', 'CeddanneRafaela', 'CF', 'BOS', '20', '54', '8', '15', '6', '0', '3', '8', '4', '12', '4', '1', '.278', '.339', '.556',
'.895']
['22', 'GrahamPauley', '1B', 'SD', '16', '35', '8', '11', '1', '1', '1', '3', '5', '9', '1', '0', '.314', '.400', '.486', '.88
6']
['23', 'LawrenceButler', 'CF', 'OAK', '19', '49', '8', '17', '6', '0', '0', '6', '6', '6', '1', '1', '.347', '.411', '.469', '.8
80']
```

```
['24', 'DaultonVarsho', 'LF', 'TOR', '18', '45', '9', '14', '3', '0', '1', '9', '10', '7', '8', '0', '.311', '.429', '.444', '.8
73']
['25', 'WillBenson', 'LF', 'CIN', '17', '43', '10', '9', '4', '1', '2', '5', '11', '15', '6', '1', '.209', '.382', '.488', '.87
0']
['26', 'DylanCarlson', 'CF', 'STL', '17', '48', '4', '13', '3', '0', '3', '13', '4', '14', '1', '0', '.271', '.327', '.521', '.8
48']
['27', 'GavinSheets', '1B', 'CWS', '23', '58', '6', '16', '4', '0', '3', '5', '5', '13', '2', '1', '.276', '.338', '.500', '.83
8']
['28', 'JoeyMeneses', '1B', 'WSH', '17', '51', '6', '15', '4', '0', '2', '9', '3', '5', '1', '0', '.294', '.345', '.490', '.83
5']
['28', 'TylerNevin', '1B', 'BAL', '22', '55', '6', '18', '2', '0', '2', '7', '2', '10', '0', '0', '.327', '.362', '.473', '.83
5']
['30', 'TylerFitzgerald', '2B', 'SF', '21', '42', '12', '9', '1', '1', '2', '11', '14', '21', '4', '1', '.214', '.404', '.429',
'.833']
['31', 'TylerWade', '3B', 'SD', '14', '34', '9', '10', '1', '1', '1', '7', '3', '6', '3', '0', '.294', '.351', '.471', '.822']
['32', 'EguyRosario', 'X', 'SD', '14', '33', '3', '9', '3', '0', '1', '6', '4', '9', '2', '0', '.273', '.359', '.455', '.814']
['33', 'ElehurisMontero', 'DH', 'COL', '21', '54', '7', '13', '5', '0', '3', '9', '3', '16', '0', '0', '.241', '.300', '.500',
'.800']
['34', 'JacksonChourio', 'CF', 'MIL', '17', '58', '13', '19', '3', '1', '0', '3', '5', '12', '2', '1', '.328', '.381', '.414',
'.795']
['34', 'MarkVientos', 'DH', 'NYM', '19', '56', '10', '13', '2', '0', '5', '9', '2', '19', '0', '0', '.232', '.259', '.536', '.79
5']
['36', 'CJAbrams', 'SS', 'WSH', '16', '49', '5', '14', '3', '0', '2', '6', '1', '6', '4', '1', '.286', '.314', '.469', '.783']
['37', 'BrettBaty', '3B', 'NYM', '17', '48', '9', '12', '1', '0', '3', '6', '4', '10', '0', '1', '.250', '.321', '.458', '.779']
['38', 'EvanCarter', 'CF', 'TEX', '18', '47', '9', '13', '1', '0', '2', '10', '4', '11', '2', '1', '.277', '.352', '.426', '.77
8']
['39', 'EsteuryRuiz', 'LF', 'OAK', '18', '50', '7', '13', '4', '1', '1', '8', '5', '17', '2', '1', '.260', '.327', '.440', '.76
7']
['39', 'JordanWalker', 'RF', 'STL', '17', '44', '8', '11', '2', '1', '1', '8', '7', '14', '2', '2', '.250', '.358', '.409', '.76
7']
['41', 'AndrewVaughn', '1B', 'CWS', '20', '55', '7', '16', '5', '0', '1', '8', '3', '12', '0', '0', '.291', '.322', '.436', '.75
8']
['42', 'CurtisMead', '3B', 'TB', '19', '43', '5', '12', '1', '0', '1', '8', '5', '4', '0', '0', '.279', '.360', '.372', '.732']
['43', 'NolanJones', 'LF', 'COL', '18', '47', '6', '12', '5', '1', '0', '4', '5', '21', '0', '1', '.255', '.327', '.404', '.73
1']
['44', 'AlexanderCanario', 'RF', 'CHC', '22', '50', '9', '12', '4', '1', '0', '2', '10', '12', '3', '0', '.240', '.367', '.360',
'.727']
['45', 'EzequielDuran', 'SS', 'TEX', '20', '58', '8', '17', '1', '1', '1', '4', '3', '9', '3', '1', '.293', '.328', '.397', '.72
5']
['46', 'JoshNaylor', '1B', 'CLE', '16', '45', '3', '14', '2', '0', '0', '7', '5', '7', '3', '1', '.311', '.365', '.356', '.721']
['47', 'AmedRosario', 'RF', 'TB', '18', '50', '8', '13', '3', '0', '2', '4', '0', '8', '3', '0', '.260', '.260', '.440', '.700']
['48', 'GavinLux', 'SS', 'LAD', '11', '32', '6', '9', '0', '0', '0', '3', '3', '5', '0', '0', '.281', '.343', '.281', '.624']
['49', 'DominicFletcher', 'RF', 'CWS', '22', '52', '6', '11', '3', '0', '1', '4', '6', '20', '1', '0', '.212', '.293', '.327',
'.620']
```

```
['50', 'WilyerAbreu', 'RF', 'BOS', '23', '56', '4', '8', '1', '1', '2', '7', '10', '23', '0', '1', '.143', '.273', '.304', '.57
7']
```

In [15]:
```
## adding two more element in headers
headers1.insert(0, 'Rank')
headers1.insert(2, 'Position')

## Create a dataframe with 'headers' as a column and 'body' as the content.
df_mlb_training = pd.DataFrame(body1)
df_mlb_training.columns = headers1
df_mlb_training = df_mlb_training.set_index('Rank')

## final data frame
df_mlb_training
```

Out[15]:

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
1	JamesWood	RF	WSH	22	44	13	16	1	1	4	7	11	13	3	1	.364	.509	.705	1.214
2	WyattLangford	LF	TEX	19	56	13	21	2	0	6	20	5	15	0	0	.375	.429	.732	1.161
3	ChristianEncarnacion-Strand	1B	CIN	15	48	13	16	2	3	4	12	3	6	0	0	.333	.385	.750	1.135
4	MookieBetts	2B	LAD	13	34	10	15	3	0	1	3	5	6	0	1	.441	.513	.618	1.131
5	LuisMatos	CF	SF	20	51	14	16	6	0	4	11	3	8	1	1	.314	.375	.667	1.042
6	AlexBregman	3B	HOU	16	47	8	18	4	0	2	8	4	7	0	0	.383	.431	.596	1.027
7	CobyMayo	3B	BAL	23	50	7	18	7	0	1	11	6	12	1	0	.360	.448	.560	1.008
8	TreyLipscomb	2B	WSH	20	48	6	19	2	1	1	7	5	7	1	2	.396	.453	.542	.995
8	JacksonMerrill	CF	SD	13	37	8	13	3	0	2	6	3	3	2	0	.351	.400	.595	.995
10	SpencerSteer	LF	CIN	16	47	8	15	3	1	3	9	5	10	3	0	.319	.377	.617	.994
11	MichaelToglia	1B	COL	20	48	9	13	3	0	5	13	5	18	1	0	.271	.340	.646	.986
12	AlecBurleson	LF	STL	18	43	3	16	4	0	1	7	6	3	1	0	.372	.440	.535	.975
13	MarcusSemien	2B	TEX	19	43	9	11	3	0	3	7	11	10	0	0	.256	.429	.535	.964
14	FreddieFreeman	1B	LAD	13	32	6	9	0	1	3	14	2	7	0	0	.281	.324	.625	.949
15	PeteAlonso	1B	NYM	17	46	4	14	5	0	2	4	7	11	0	0	.304	.396	.543	.939
16	ZackGelof	2B	OAK	18	51	13	15	3	0	4	8	4	19	0	1	.294	.345	.588	.933
17	LaneThomas	RF	WSH	19	48	12	15	0	0	3	6	9	11	5	1	.313	.424	.500	.924
18	EllyDe La Cruz	SS	CIN	16	47	13	13	3	2	2	6	7	20	5	1	.277	.370	.553	.923
19	JoséFermín	3B	STL	22	43	9	13	3	0	2	4	6	5	1	0	.302	.400	.512	.912
20	RichiePalacios	LF	TB	18	48	8	15	2	0	3	7	4	10	2	1	.313	.358	.542	.900
21	CeddanneRafaela	CF	BOS	20	54	8	15	6	0	3	8	4	12	4	1	.278	.339	.556	.895
22	GrahamPauley	1B	SD	16	35	8	11	1	1	1	3	5	9	1	0	.314	.400	.486	.886
23	LawrenceButler	CF	OAK	19	49	8	17	6	0	0	6	6	6	1	1	.347	.411	.469	.880

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
24	DaultonVarsho	LF	TOR	18	45	9	14	3	0	1	9	10	7	8	0	.311	.429	.444	.873
25	WillBenson	LF	CIN	17	43	10	9	4	1	2	5	11	15	6	1	.209	.382	.488	.870
26	DylanCarlson	CF	STL	17	48	4	13	3	0	3	13	4	14	1	0	.271	.327	.521	.848
27	GavinSheets	1B	CWS	23	58	6	16	4	0	3	5	5	13	2	1	.276	.338	.500	.838
28	JoeyMeneses	1B	WSH	17	51	6	15	4	0	2	9	3	5	1	0	.294	.345	.490	.835
28	TylerNevin	1B	BAL	22	55	6	18	2	0	2	7	2	10	0	0	.327	.362	.473	.835
30	TylerFitzgerald	2B	SF	21	42	12	9	1	1	2	11	14	21	4	1	.214	.404	.429	.833
31	TylerWade	3B	SD	14	34	9	10	1	1	1	7	3	6	3	0	.294	.351	.471	.822
32	EguyRosario	X	SD	14	33	3	9	3	0	1	6	4	9	2	0	.273	.359	.455	.814
33	ElehurisMontero	DH	COL	21	54	7	13	5	0	3	9	3	16	0	0	.241	.300	.500	.800
34	JacksonChourio	CF	MIL	17	58	13	19	3	1	0	3	5	12	2	1	.328	.381	.414	.795
34	MarkVientos	DH	NYM	19	56	10	13	2	0	5	9	2	19	0	0	.232	.259	.536	.795
36	CJAbrams	SS	WSH	16	49	5	14	3	0	2	6	1	6	4	1	.286	.314	.469	.783
37	BrettBaty	3B	NYM	17	48	9	12	1	0	3	6	4	10	0	1	.250	.321	.458	.779
38	EvanCarter	CF	TEX	18	47	9	13	1	0	2	10	4	11	2	1	.277	.352	.426	.778
39	EsteuryRuiz	LF	OAK	18	50	7	13	4	1	1	8	5	17	2	1	.260	.327	.440	.767
39	JordanWalker	RF	STL	17	44	8	11	2	1	1	8	7	14	2	2	.250	.358	.409	.767
41	AndrewVaughn	1B	CWS	20	55	7	16	5	0	1	8	3	12	0	0	.291	.322	.436	.758
42	CurtisMead	3B	TB	19	43	5	12	1	0	1	8	5	4	0	0	.279	.360	.372	.732
43	NolanJones	LF	COL	18	47	6	12	5	1	0	4	5	21	0	1	.255	.327	.404	.731
44	AlexanderCanario	RF	CHC	22	50	9	12	4	1	0	2	10	12	3	0	.240	.367	.360	.727
45	EzequielDuran	SS	TEX	20	58	8	17	1	1	1	4	3	9	3	1	.293	.328	.397	.725
46	JoshNaylor	1B	CLE	16	45	3	14	2	0	0	7	5	7	3	1	.311	.365	.356	.721

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
47	AmedRosario	RF	TB	18	50	8	13	3	0	2	4	0	8	3	0	.260	.260	.440	.700
48	GavinLux	SS	LAD	11	32	6	9	0	0	0	3	3	5	0	0	.281	.343	.281	.624
49	DominicFletcher	RF	CWS	22	52	6	11	3	0	1	4	6	20	1	0	.212	.293	.327	.620
50	WilyerAbreu	RF	BOS	23	56	4	8	1	1	2	7	10	23	0	1	.143	.273	.304	.577

In [16]:
```
## drive already mounted at /content/drive
## conneting google drive to colab
#from google.colab import drive
#drive.mount('/content/drive')
```

In [17]:
```
## save data frame as xlsx file
df_mlb_training = df_mlb_training.to_excel('/content/sample_data/MLB2024_Training.xlsx', index=False)
```

ML prediction for Spring Training time

- Using the crawled data, we will run a machine learning task
- For simple tutorial, we'll see how to predict the number of homeruns from other stats

In [18]:
```
df_mlb_training = pd.read_excel('/content/sample_data/MLB2024_Training.xlsx')
X_df_mlb_training = df_mlb_training.drop('HR', axis=1) ## delete HR data (it will be answer) from data frame
X_df_mlb_training = X_df_mlb_training.drop(['PLAYER', 'Position', 'TEAM'], axis=1) ## delete non-floating number data
X_df_mlb_training
```

Out[18]:

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
0	22	44	13	16	1	1	7	11	13	3	1	0.364	0.509	0.705	1.214
1	19	56	13	21	2	0	20	5	15	0	0	0.375	0.429	0.732	1.161
2	15	48	13	16	2	3	12	3	6	0	0	0.333	0.385	0.750	1.135
3	13	34	10	15	3	0	3	5	6	0	1	0.441	0.513	0.618	1.131
4	20	51	14	16	6	0	11	3	8	1	1	0.314	0.375	0.667	1.042
5	16	47	8	18	4	0	8	4	7	0	0	0.383	0.431	0.596	1.027
6	23	50	7	18	7	0	11	6	12	1	0	0.360	0.448	0.560	1.008
7	20	48	6	19	2	1	7	5	7	1	2	0.396	0.453	0.542	0.995
8	13	37	8	13	3	0	6	3	3	2	0	0.351	0.400	0.595	0.995
9	16	47	8	15	3	1	9	5	10	3	0	0.319	0.377	0.617	0.994
10	20	48	9	13	3	0	13	5	18	1	0	0.271	0.340	0.646	0.986
11	18	43	3	16	4	0	7	6	3	1	0	0.372	0.440	0.535	0.975
12	19	43	9	11	3	0	7	11	10	0	0	0.256	0.429	0.535	0.964
13	13	32	6	9	0	1	14	2	7	0	0	0.281	0.324	0.625	0.949
14	17	46	4	14	5	0	4	7	11	0	0	0.304	0.396	0.543	0.939
15	18	51	13	15	3	0	8	4	19	0	1	0.294	0.345	0.588	0.933
16	19	48	12	15	0	0	6	9	11	5	1	0.313	0.424	0.500	0.924
17	16	47	13	13	3	2	6	7	20	5	1	0.277	0.370	0.553	0.923
18	22	43	9	13	3	0	4	6	5	1	0	0.302	0.400	0.512	0.912
19	18	48	8	15	2	0	7	4	10	2	1	0.313	0.358	0.542	0.900
20	20	54	8	15	6	0	8	4	12	4	1	0.278	0.339	0.556	0.895
21	16	35	8	11	1	1	3	5	9	1	0	0.314	0.400	0.486	0.886
22	19	49	8	17	6	0	6	6	6	1	1	0.347	0.411	0.469	0.880
23	18	45	9	14	3	0	9	10	7	8	0	0.311	0.429	0.444	0.873

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
24	17	43	10	9	4	1	5	11	15	6	1	0.209	0.382	0.488	0.870
25	17	48	4	13	3	0	13	4	14	1	0	0.271	0.327	0.521	0.848
26	23	58	6	16	4	0	5	5	13	2	1	0.276	0.338	0.500	0.838
27	17	51	6	15	4	0	9	3	5	1	0	0.294	0.345	0.490	0.835
28	22	55	6	18	2	0	7	2	10	0	0	0.327	0.362	0.473	0.835
29	21	42	12	9	1	1	11	14	21	4	1	0.214	0.404	0.429	0.833
30	14	34	9	10	1	1	7	3	6	3	0	0.294	0.351	0.471	0.822
31	14	33	3	9	3	0	6	4	9	2	0	0.273	0.359	0.455	0.814
32	21	54	7	13	5	0	9	3	16	0	0	0.241	0.300	0.500	0.800
33	17	58	13	19	3	1	3	5	12	2	1	0.328	0.381	0.414	0.795
34	19	56	10	13	2	0	9	2	19	0	0	0.232	0.259	0.536	0.795
35	16	49	5	14	3	0	6	1	6	4	1	0.286	0.314	0.469	0.783
36	17	48	9	12	1	0	6	4	10	0	1	0.250	0.321	0.458	0.779
37	18	47	9	13	1	0	10	4	11	2	1	0.277	0.352	0.426	0.778
38	18	50	7	13	4	1	8	5	17	2	1	0.260	0.327	0.440	0.767
39	17	44	8	11	2	1	8	7	14	2	2	0.250	0.358	0.409	0.767
40	20	55	7	16	5	0	8	3	12	0	0	0.291	0.322	0.436	0.758
41	19	43	5	12	1	0	8	5	4	0	0	0.279	0.360	0.372	0.732
42	18	47	6	12	5	1	4	5	21	0	1	0.255	0.327	0.404	0.731
43	22	50	9	12	4	1	2	10	12	3	0	0.240	0.367	0.360	0.727
44	20	58	8	17	1	1	4	3	9	3	1	0.293	0.328	0.397	0.725
45	16	45	3	14	2	0	7	5	7	3	1	0.311	0.365	0.356	0.721
46	18	50	8	13	3	0	4	0	8	3	0	0.260	0.260	0.440	0.700
47	11	32	6	9	0	0	3	3	5	0	0	0.281	0.343	0.281	0.624

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
48	22	52	6	11	3	0	4	6	20	1	0	0.212	0.293	0.327	0.620
49	23	56	4	8	1	1	7	10	23	0	1	0.143	0.273	0.304	0.577

In [19]:
```
y_df_mlb_training = df_mlb_training['HR']## setting HR data as y (target of prediction)
y_df_mlb_training
```

Out[19]:

0	4
1	6
2	4
3	1
4	4
5	2
6	1
7	1
8	2
9	3
10	5
11	1
12	3
13	3
14	2
15	4
16	3
17	2
18	2
19	3
20	3
21	1
22	0
23	1
24	2
25	3
26	3
27	2
28	2
29	2
30	1
31	1
32	3
33	0
34	5
35	2
36	3
37	2
38	1
39	1
40	1
41	1
42	0
43	0

```
44    1
45    0
46    2
47    0
48    1
49    2
Name: HR, dtype: int64
```

In [20]:
```
from sklearn.model_selection import train_test_split
## define train data and test data
X_train1, X_test1, y_train1, y_test1 = train_test_split(X_df_mlb_training,y_df_mlb_training, test_size=0.2, random_state=0xc0ffee
print(X_train1.shape, X_test1.shape, y_train1.shape, y_test1.shape)
```

```
(40, 15) (10, 15) (40,) (10,)
```

In [21]:
```
from sklearn.preprocessing import StandardScaler
## scale the data
scalr = StandardScaler()
X_train1 = scalr.fit_transform(X_train1)
X_test1 = scalr.transform(X_test1)
```

In [22]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
```

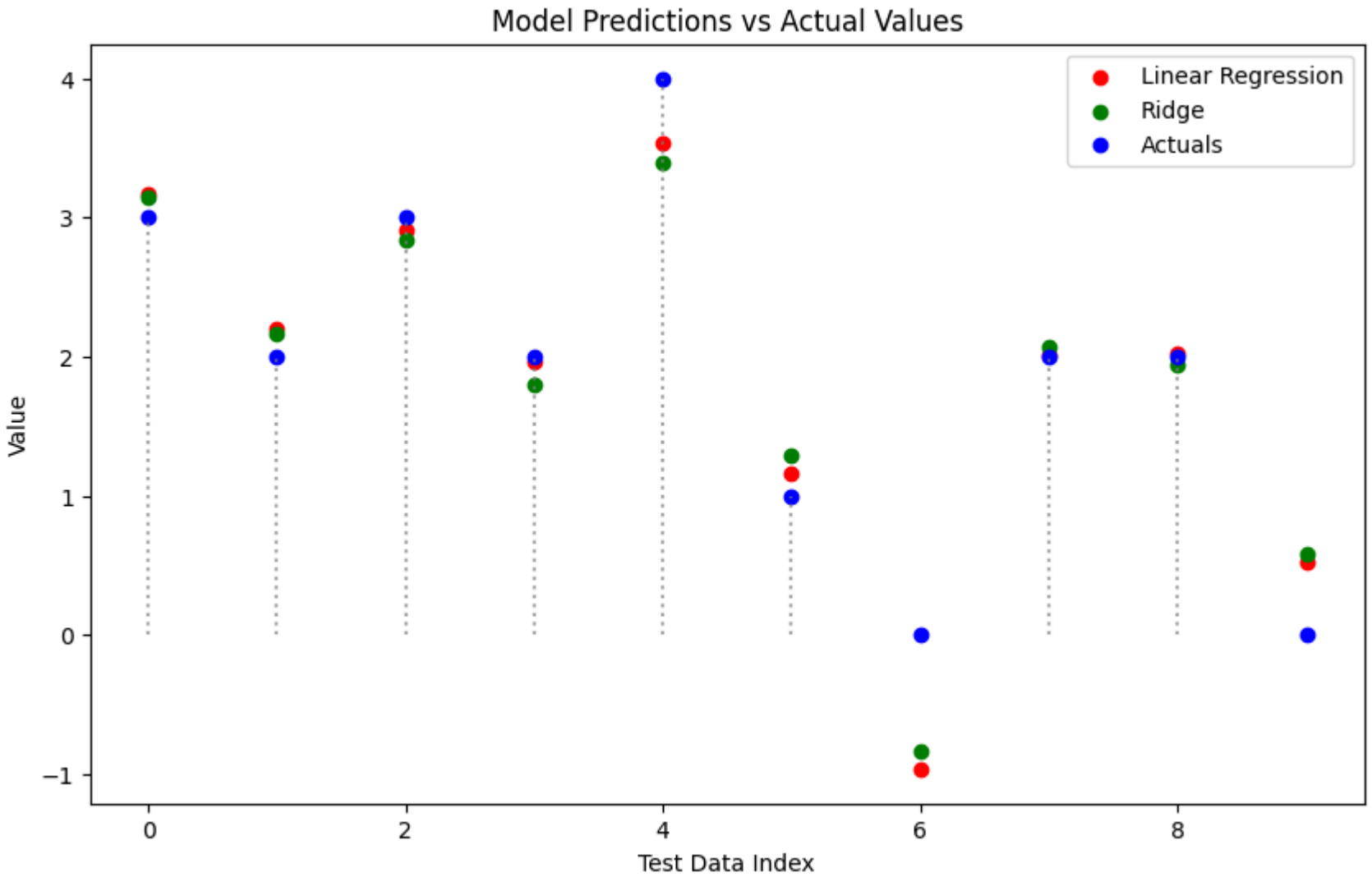
In [23]:
```
## train Linear Regression and Ridge Regression model and check the errors
reg1  = LinearRegression()
reg1.fit(X_train1, y_train1)
pred_train1 = reg1.predict(X_train1)
pred_val1 = reg1.predict(X_test1)
mse_train1 = mean_squared_error(y_train1, pred_train1)
mse_val1 = mean_squared_error(y_test1, pred_val1)

reg2 = Ridge()
reg2.fit(X_train1, y_train1)
pred_train2 = reg2.predict(X_train1)
pred_val2 = reg2.predict(X_test1)
mse_train2 = mean_squared_error(y_train1, pred_train2)
mse_val2 = mean_squared_error(y_test1, pred_val2)

print("1. Linear Regresion\t, train = %.4f, val = %.4f" %(mse_train1, mse_val1))
print("2. Ridge\t\t, train = %.4f, val = %.4f" %(mse_train2, mse_val2))
```

```
1. Linear Regresion     , train = 0.0207, val = 0.1526
2. Ridge                , train = 0.0269, val = 0.1614
```

Data Visualization for Spring Training time

Here is the sample code to visualize the prediction results.

In [24]:
```python
import matplotlib.pyplot as plt

y_test1 =  y_test1.astype(int).values

# Drawing scatter plots
plt.figure(figsize=(10, 6))
plt.scatter(range(len(pred_val1)), pred_val1, color='r', label='Linear Regression')
plt.scatter(range(len(pred_val2)), pred_val2, color='g', label='Ridge')
plt.scatter(range(len(y_test1)), y_test1, color='b', label='Actuals')

for i in range(len(y_test1)):
    plt.vlines(x=i, ymin=0, ymax=y_test1[i], colors='gray', linestyles='dotted', alpha=0.7)

plt.xlabel('Test Data Index')
plt.ylabel('Value')
plt.title('Model Predictions vs Actual Values')
plt.legend()

plt.show()
```



In [25]:
```
df_mlb_training['HR'] = df_mlb_training['HR'].astype('int')
df_mlb_training[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
```

Out[25]:

	PLAYER	HR
0	WyattLangford	6
1	MarkVientos	5
2	MichaelToglia	5
3	JamesWood	4
4	ZackGelof	4
5	ChristianEncarnacion-Strand	4
6	LuisMatos	4
7	BrettBaty	3
8	ElehurisMontero	3
9	GavinSheets	3
10	CeddanneRafaela	3
11	RichiePalacios	3
12	LaneThomas	3
13	DylanCarlson	3
14	MarcusSemien	3
15	SpencerSteer	3
16	FreddieFreeman	3
17	PeteAlonso	2
18	JoeyMeneses	2
19	AmedRosario	2

Prediction by using data from Spring Training time

In [26]:
```
features = ['OPS']# On-base plus slugging
X = df_mlb_training[features]
```

```
y = df_mlb_training['HR']  # Home runs

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model selection and training
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Prediction on unseen data (2024 season) by using data from Spring Training time
y_pred = model.predict(X_test)

# Top 10 predicted home run hitters
top_10_idx = y_pred.argsort()[:10]
top_10_players = df_mlb_training.iloc[top_10_idx]['PLAYER'].to_list()

print("Top 10 predicted home run hitters for 2024:")
for player in top_10_players:
  print(player)
```

```
Top 10 predicted home run hitters for 2024:
MookieBetts
AlexBregman
WyattLangford
ChristianEncarnacion-Strand
CobyMayo
TreyLipscomb
JamesWood
SpencerSteer
JacksonMerrill
LuisMatos
```

MLB 2023

Data crawling for Regular Season 2023

In [28]:
```
## set driver url to crawler

driver.get('https://www.mlb.com/stats/2023')

## find element from web
```

```
table2 = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[1]/div/table')

header_items = table2.find_elements(By.XPATH, './thead/tr/th')
headers2 = [h.text for h in header_items]

## print properties
print(headers2)

## find element from web
body2 = []
while True:
  time.sleep(2)

  ## crawl from the first page and save it using driver.find_element method
  body_items = table2.find_elements(By.XPATH, './tbody/tr')
  for b in body_items:
    row_title = b.find_elements(By.XPATH, './th')
    row_items = b.find_elements(By.XPATH, './td')
    row = [r.text for r in row_title] + [r.text for r in row_items]

    ## since the first element in each column is in the form '1\nShoheiOhtani\nDH',
    ## we tokenize it and make it an individual element.
    row = row[0].split('\n') + row[1:]
    body2.append(row)
    print(row)
  try:
    cookie_button = driver.find_element(By.XPATH, '//*[@id="onetrust-accept-btn-handler"]')
    cookie_button.click()
  except:
    pass
  try:
    next_button = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[2]/div/div/div[last()]/butt
    next_button.click()
  except:
    break
```

```
['PLAYER', 'TEAM', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR', 'RBI', 'BB', 'SO', 'SB', 'CS', 'AVG', 'OBP', 'SLG', 'OPS']
['1', 'ShoheiOhtani', 'DH', 'LAA', '135', '497', '102', '151', '26', '8', '44', '95', '91', '143', '20', '6', '.304', '.412', '.
654', '1.066']
['2', 'CoreySeager', 'SS', 'TEX', '119', '477', '88', '156', '42', '0', '33', '96', '49', '88', '2', '1', '.327', '.390', '.62
3', '1.013']
['3', 'RonaldAcuña', 'RF', 'ATL', '159', '643', '149', '217', '35', '4', '41', '106', '80', '84', '73', '14', '.337', '.416', '.
596', '1.012']
['4', 'MattOlson', '1B', 'ATL', '162', '608', '127', '172', '27', '3', '54', '139', '104', '167', '1', '0', '.283', '.389', '.60
4', '.993']
['5', 'MookieBetts', 'RF', 'LAD', '152', '584', '126', '179', '40', '1', '39', '107', '96', '107', '14', '3', '.307', '.408', '.
579', '.987']
['6', 'FreddieFreeman', '1B', 'LAD', '161', '637', '131', '211', '59', '2', '29', '102', '72', '121', '23', '1', '.331', '.410',
'.567', '.977']
['7', 'YandyDíaz', '1B', 'TB', '137', '525', '95', '173', '35', '0', '22', '78', '65', '94', '0', '1', '.330', '.410', '.522',
'.932']
['8', 'JuanSoto', 'LF', 'SD', '162', '568', '97', '156', '32', '1', '35', '109', '132', '129', '12', '5', '.275', '.410', '.51
9', '.929']
['9', 'MarcellOzuna', 'DH', 'ATL', '144', '530', '84', '145', '29', '1', '40', '100', '57', '134', '0', '0', '.274', '.346', '.5
58', '.904']
['10', 'BryceHarper', 'DH', 'PHI', '126', '457', '84', '134', '29', '1', '21', '72', '80', '119', '11', '3', '.293', '.401', '.4
99', '.900']
['11', 'KyleTucker', 'RF', 'HOU', '157', '574', '97', '163', '37', '5', '29', '112', '80', '92', '30', '5', '.284', '.369', '.51
7', '.886']
['12', 'CodyBellinger', 'CF', 'CHC', '130', '499', '95', '153', '29', '1', '26', '97', '40', '87', '20', '6', '.307', '.356', '.
525', '.881']
['13', 'CorbinCarroll', 'LF', 'AZ', '155', '565', '116', '161', '30', '10', '25', '76', '57', '125', '54', '5', '.285', '.362',
'.506', '.868']
['14', 'LuisArraez', '2B', 'MIA', '147', '574', '71', '203', '30', '3', '10', '69', '35', '34', '3', '2', '.354', '.393', '.46
9', '.862']
['15', 'AustinRiley', '3B', 'ATL', '159', '636', '117', '179', '32', '3', '37', '97', '59', '172', '3', '1', '.281', '.345', '.5
16', '.861']
['16', 'TristonCasas', '1B', 'BOS', '132', '429', '66', '113', '21', '2', '24', '65', '70', '126', '0', '0', '.263', '.367', '.4
90', '.857']
['16', 'LuisRobert', 'CF', 'CWS', '145', '546', '90', '144', '36', '1', '38', '80', '30', '172', '20', '4', '.264', '.315', '.54
2', '.857']
['18', 'JorgeSoler', 'DH', 'MIA', '137', '504', '77', '126', '24', '0', '36', '75', '66', '141', '1', '0', '.250', '.341', '.51
2', '.853']
['19', 'RafaelDevers', '3B', 'BOS', '153', '580', '90', '157', '34', '0', '33', '100', '62', '126', '5', '1', '.271', '.351', '.
500', '.851']
['20', 'OzzieAlbies', '2B', 'ATL', '148', '596', '96', '167', '30', '5', '33', '109', '46', '107', '13', '1', '.280', '.336', '.
513', '.849']
['21', 'KetelMarte', '2B', 'AZ', '150', '569', '94', '157', '26', '9', '25', '82', '71', '109', '8', '2', '.276', '.358', '.48
5', '.843']
['22', 'SeiyaSuzuki', 'RF', 'CHC', '138', '515', '75', '147', '31', '6', '20', '74', '59', '130', '6', '7', '.285', '.357', '.48
```

```
5', '.842']
['23', 'IsaacParedes', '3B', 'TB', '143', '492', '71', '123', '24', '0', '31', '98', '58', '104', '1', '0', '.250', '.352', '.48
8', '.840']
['24', 'AdolisGarcía', 'RF', 'TEX', '148', '555', '108', '136', '29', '0', '39', '107', '65', '175', '9', '1', '.245', '.328',
'.508', '.836']
['25', 'JoséRamírez', '3B', 'CLE', '156', '611', '87', '172', '36', '5', '24', '80', '73', '73', '28', '6', '.282', '.356', '.47
5', '.831']
['26', 'ChristianWalker', '1B', 'AZ', '157', '582', '86', '150', '36', '2', '33', '103', '62', '127', '11', '0', '.258', '.333',
'.497', '.830']
['27', 'BrandonNimmo', 'CF', 'NYM', '152', '592', '89', '162', '30', '6', '24', '68', '74', '146', '3', '3', '.274', '.363', '.4
66', '.829']
['28', 'JakeBurger', '3B', 'MIA', '141', '492', '71', '123', '28', '1', '34', '80', '32', '149', '1', '1', '.250', '.309', '.51
8', '.827']
['29', 'MarcusSemien', '2B', 'TEX', '162', '670', '122', '185', '40', '4', '29', '100', '72', '110', '14', '3', '.276', '.348',
'.478', '.826']
['30', 'WilliamContreras', 'C', 'MIL', '141', '540', '86', '156', '38', '1', '17', '78', '63', '126', '6', '1', '.289', '.367',
'.457', '.824']
['31', 'PeteAlonso', '1B', 'NYM', '154', '568', '92', '123', '21', '2', '46', '118', '65', '151', '4', '1', '.217', '.318', '.50
4', '.822']
['32', 'SpencerSteer', '1B', 'CIN', '156', '582', '74', '158', '37', '3', '23', '86', '68', '139', '15', '3', '.271', '.356', '.
464', '.820']
['33', 'TJFriedl', 'CF', 'CIN', '138', '488', '73', '136', '22', '8', '18', '66', '47', '90', '27', '6', '.279', '.352', '.467',
'.819']
['34', 'J.P.Crawford', 'SS', 'SEA', '145', '534', '94', '142', '35', '0', '19', '65', '94', '125', '2', '0', '.266', '.380', '.4
38', '.818']
['34', 'JulioRodríguez', 'CF', 'SEA', '155', '654', '102', '180', '37', '2', '32', '103', '47', '175', '37', '10', '.275', '.33
3', '.485', '.818']
['36', 'BrentRooker', 'DH', 'OAK', '137', '463', '61', '114', '20', '1', '30', '69', '49', '172', '4', '0', '.246', '.329', '.48
8', '.817']
['36', 'KyleSchwarber', 'LF', 'PHI', '160', '585', '108', '115', '19', '1', '47', '104', '126', '215', '0', '2', '.197', '.343',
'.474', '.817']
['36', 'ChristianYelich', 'LF', 'MIL', '144', '550', '106', '153', '34', '1', '19', '76', '78', '140', '28', '3', '.278', '.37
0', '.447', '.817']
['39', 'BoBichette', 'SS', 'TOR', '135', '571', '69', '175', '30', '3', '20', '73', '27', '115', '5', '3', '.306', '.339', '.47
5', '.814']
['39', 'GunnarHenderson', '3B', 'BAL', '150', '560', '100', '143', '29', '9', '28', '82', '56', '159', '10', '3', '.255', '.32
5', '.489', '.814']
['39', 'BobbyWitt', 'SS', 'KC', '158', '641', '97', '177', '28', '11', '30', '96', '40', '121', '49', '15', '.276', '.319', '.49
5', '.814']
['42', 'PaulGoldschmidt', '1B', 'STL', '154', '593', '89', '159', '31', '0', '25', '80', '87', '161', '11', '2', '.268', '.363',
'.447', '.810']
['43', 'AdleyRutschman', 'C', 'BAL', '154', '588', '84', '163', '31', '1', '20', '80', '92', '101', '1', '2', '.277', '.374', '.
435', '.809']
['44', 'MichaelHarris', 'CF', 'ATL', '138', '505', '76', '148', '33', '3', '18', '57', '25', '101', '20', '4', '.293', '.331',
```

```
'.477', '.808']
['44', 'MaxMuncy', '3B', 'LAD', '135', '482', '95', '102', '17', '1', '36', '105', '85', '153', '1', '2', '.212', '.333', '.47
5', '.808']
['46', 'JeimerCandelario', '1B', 'CHC', '140', '505', '77', '127', '39', '3', '22', '70', '53', '127', '8', '1', '.251', '.336',
'.471', '.807']
['47', 'FranciscoLindor', 'SS', 'NYM', '160', '602', '108', '153', '33', '2', '31', '98', '66', '137', '31', '4', '.254', '.33
6', '.470', '.806']
['48', 'AlexBregman', '3B', 'HOU', '161', '622', '103', '163', '28', '4', '25', '98', '92', '87', '3', '1', '.262', '.363', '.44
1', '.804']
['49', 'BrandonDrury', '2B', 'LAA', '125', '485', '61', '127', '30', '3', '26', '83', '25', '136', '0', '2', '.262', '.306', '.4
97', '.803']
['50', 'GleyberTorres', '2B', 'NYY', '158', '596', '90', '163', '28', '2', '25', '68', '67', '98', '13', '6', '.273', '.347', '.
453', '.800']
['50', 'JustinTurner', 'DH', 'BOS', '146', '558', '86', '154', '31', '0', '23', '96', '51', '110', '4', '1', '.276', '.345', '.4
55', '.800']
['52', 'AnthonySantander', 'RF', 'BAL', '153', '591', '81', '152', '41', '1', '28', '95', '55', '152', '5', '1', '.257', '.325',
'.472', '.797']
['52', 'WillSmith', 'C', 'LAD', '126', '464', '80', '121', '21', '2', '19', '76', '63', '89', '3', '0', '.261', '.359', '.438',
'.797']
['54', 'JackSuwinski', 'CF', 'PIT', '144', '447', '63', '100', '21', '2', '26', '74', '75', '172', '13', '2', '.224', '.339', '.
454', '.793']
['55', 'IanHapp', 'LF', 'CHC', '158', '580', '86', '144', '35', '4', '21', '84', '99', '153', '14', '3', '.248', '.360', '.431',
'.791']
['56', 'XanderBogaerts', 'SS', 'SD', '155', '596', '83', '170', '31', '2', '19', '58', '56', '110', '19', '2', '.285', '.350',
'.440', '.790']
['56', 'JamesOutman', 'CF', 'LAD', '151', '483', '86', '120', '16', '3', '23', '70', '68', '181', '16', '3', '.248', '.353', '.4
37', '.790']
['56', 'BryanReynolds', 'LF', 'PIT', '145', '574', '85', '151', '31', '5', '24', '84', '53', '138', '12', '1', '.263', '.330',
'.460', '.790']
['56', 'LaMonteWade', '1B', 'SF', '135', '429', '64', '110', '14', '2', '17', '45', '76', '95', '2', '0', '.256', '.373', '.41
7', '.790']
['60', 'RandyArozarena', 'LF', 'TB', '151', '551', '95', '140', '19', '3', '23', '83', '80', '156', '22', '10', '.254', '.364',
'.425', '.789']
['60', 'VladimirGuerrero', '1B', 'TOR', '156', '602', '78', '159', '30', '0', '26', '94', '67', '100', '5', '3', '.264', '.345',
'.444', '.789']
['62', 'NickCastellanos', 'RF', 'PHI', '157', '626', '79', '170', '37', '2', '29', '106', '36', '185', '11', '2', '.272', '.31
1', '.476', '.787']
['63', 'LarsNootbaar', 'CF', 'STL', '117', '426', '74', '111', '23', '1', '14', '46', '72', '99', '11', '1', '.261', '.367', '.4
18', '.785']
['64', 'LaneThomas', 'RF', 'WSH', '157', '628', '101', '168', '36', '3', '28', '86', '36', '176', '20', '5', '.268', '.315', '.4
68', '.783']
['64', 'MasatakaYoshida', 'LF', 'BOS', '140', '537', '71', '155', '33', '3', '15', '72', '34', '81', '8', '0', '.289', '.338',
'.445', '.783']
['66', 'JoshJung', '3B', 'TEX', '122', '478', '75', '127', '25', '1', '23', '70', '30', '151', '1', '3', '.266', '.315', '.467',
```

```
'.782']
['67', 'MannyMachado', '3B', 'SD', '138', '543', '75', '140', '21', '0', '30', '91', '50', '109', '3', '2', '.258', '.319', '.46
2', '.781']
['68', 'TreaTurner', 'SS', 'PHI', '155', '639', '102', '170', '35', '5', '26', '76', '45', '150', '30', '0', '.266', '.320', '.4
59', '.779']
['69', 'NolanArenado', '3B', 'STL', '144', '560', '71', '149', '26', '2', '26', '93', '41', '101', '3', '3', '.266', '.315', '.4
59', '.774']
['69', 'NathanielLowe', '1B', 'TEX', '161', '623', '89', '163', '38', '3', '17', '82', '93', '165', '1', '0', '.262', '.360', '.
414', '.774']
['71', 'LourdesGurriel', 'LF', 'AZ', '145', '551', '65', '144', '35', '2', '24', '82', '33', '103', '5', '0', '.261', '.309', '.
463', '.772']
['72', 'FernandoTatis', 'RF', 'SD', '141', '575', '91', '148', '33', '1', '25', '78', '53', '141', '29', '4', '.257', '.322', '.
449', '.771']
['73', 'AustinHays', 'LF', 'BAL', '144', '520', '76', '143', '36', '2', '16', '67', '38', '141', '5', '1', '.275', '.325', '.44
4', '.769']
['74', 'AlecBohm', '3B', 'PHI', '145', '558', '74', '153', '31', '0', '20', '97', '42', '94', '4', '1', '.274', '.327', '.437',
'.764']
['75', "Ke'BryanHayes", '3B', 'PIT', '124', '494', '65', '134', '31', '7', '15', '61', '28', '104', '10', '6', '.271', '.309',
'.453', '.762']
['75', 'CalRaleigh', 'C', 'SEA', '145', '513', '78', '119', '23', '1', '30', '75', '54', '158', '0', '0', '.232', '.306', '.45
6', '.762']
['75', 'J.T.Realmuto', 'C', 'PHI', '135', '489', '70', '123', '28', '5', '20', '63', '35', '138', '16', '5', '.252', '.310', '.4
52', '.762']
['78', 'SpencerTorkelson', '1B', 'DET', '159', '606', '88', '141', '34', '1', '31', '94', '67', '171', '3', '0', '.233', '.313',
'.446', '.759']
['79', 'MarkCanha', 'DH', 'MIL', '139', '435', '51', '114', '25', '1', '11', '62', '49', '79', '11', '1', '.262', '.355', '.40
0', '.755']
['79', 'EddieRosario', 'LF', 'ATL', '142', '478', '64', '122', '24', '3', '21', '74', '34', '122', '3', '4', '.255', '.305', '.4
50', '.755']
['81', 'MattChapman', '3B', 'TOR', '140', '509', '66', '122', '39', '2', '17', '54', '62', '165', '4', '2', '.240', '.330', '.42
4', '.754']
['82', 'RyanMcMahon', '3B', 'COL', '152', '555', '80', '133', '31', '3', '23', '70', '68', '198', '5', '5', '.240', '.322', '.43
1', '.753']
['83', 'Ha-SeongKim', '2B', 'SD', '152', '538', '84', '140', '23', '0', '17', '60', '75', '124', '38', '9', '.260', '.351', '.39
8', '.749']
['84', 'BrysonStott', '2B', 'PHI', '151', '585', '78', '164', '32', '2', '15', '62', '39', '100', '31', '3', '.280', '.329', '.4
19', '.748']
['85', 'CarlosSantana', '1B', 'MIL', '146', '550', '78', '132', '33', '1', '23', '86', '65', '104', '6', '0', '.240', '.318', '.
429', '.747']
['86', 'JonathanIndia', '2B', 'CIN', '119', '454', '78', '111', '23', '0', '17', '61', '52', '109', '14', '2', '.244', '.338',
'.407', '.745']
['86', 'AlexVerdugo', 'RF', 'BOS', '142', '546', '81', '144', '37', '5', '13', '54', '45', '93', '5', '3', '.264', '.324', '.42
1', '.745']
['88', 'JoshBell', '1B', 'MIA', '150', '547', '52', '135', '28', '0', '22', '74', '63', '134', '0', '1', '.247', '.325', '.419',
```

```
'.744']
['88', 'DansbySwanson', 'SS', 'CHC', '147', '565', '81', '138', '25', '3', '22', '80', '66', '154', '9', '1', '.244', '.328', '.
416', '.744']
['90', 'AndrewVaughn', '1B', 'CWS', '152', '566', '67', '146', '30', '2', '21', '80', '36', '129', '0', '0', '.258', '.314', '.4
29', '.743']
['91', 'OrlandoArcia', 'SS', 'ATL', '139', '488', '66', '129', '25', '0', '17', '65', '39', '102', '1', '0', '.264', '.321', '.4
20', '.741']
['92', 'TeoscarHernández', 'RF', 'SEA', '160', '625', '70', '161', '29', '2', '26', '93', '38', '211', '7', '2', '.258', '.305',
'.435', '.740']
['93', 'J.D.Davis', '3B', 'SF', '144', '480', '61', '119', '23', '1', '18', '69', '52', '152', '1', '0', '.248', '.325', '.413',
'.738']
['94', 'LeodyTaveras', 'CF', 'TEX', '143', '511', '67', '136', '31', '3', '14', '67', '35', '117', '14', '4', '.266', '.312', '.
421', '.733']
['95', 'GeorgeSpringer', 'RF', 'TOR', '154', '613', '87', '158', '25', '1', '21', '72', '60', '125', '20', '5', '.258', '.327',
'.405', '.732']
['96', 'ThairoEstrada', '2B', 'SF', '120', '495', '63', '134', '26', '2', '14', '49', '22', '120', '23', '7', '.271', '.315', '.
416', '.731']
['97', 'NicoHoerner', '2B', 'CHC', '150', '619', '98', '175', '27', '4', '9', '68', '49', '83', '43', '7', '.283', '.346', '.38
3', '.729']
['98', 'EliasDíaz', 'C', 'COL', '141', '486', '48', '130', '25', '1', '14', '72', '34', '118', '1', '0', '.267', '.316', '.409',
'.725']
['99', 'JoeyMeneses', 'DH', 'WSH', '154', '611', '71', '168', '36', '1', '13', '89', '38', '130', '0', '0', '.275', '.321', '.40
1', '.722']
['100', 'WillyAdames', 'SS', 'MIL', '149', '553', '73', '120', '29', '2', '24', '80', '71', '165', '5', '3', '.217', '.310', '.4
07', '.717']
['100', 'DJLeMahieu', '3B', 'NYY', '136', '497', '55', '121', '22', '3', '15', '44', '60', '125', '2', '2', '.243', '.327', '.39
0', '.717']
['100', 'KeibertRuiz', 'C', 'WSH', '136', '523', '55', '136', '24', '0', '18', '67', '31', '58', '1', '1', '.260', '.308', '.40
9', '.717']
['100', 'MattVierling', 'RF', 'DET', '134', '479', '63', '125', '21', '5', '10', '44', '44', '112', '6', '6', '.261', '.329', '.
388', '.717']
['104', 'BryanDe La Cruz', 'LF', 'MIA', '153', '579', '60', '149', '32', '0', '19', '78', '40', '142', '4', '1', '.257', '.304',
'.411', '.715']
['105', 'MJMelendez', 'RF', 'KC', '148', '533', '65', '125', '29', '5', '16', '56', '62', '170', '6', '4', '.235', '.316', '.39
8', '.714']
['105', 'SalvadorPerez', 'C', 'KC', '140', '538', '59', '137', '21', '0', '23', '80', '19', '135', '0', '0', '.255', '.292', '.4
22', '.714']
['105', 'EugenioSuárez', '3B', 'SEA', '162', '598', '68', '139', '29', '0', '22', '96', '70', '214', '2', '1', '.232', '.323',
'.391', '.714']
['108', 'AndrésGiménez', '2B', 'CLE', '153', '557', '76', '140', '27', '5', '15', '62', '32', '112', '30', '6', '.251', '.314',
'.399', '.713']
['108', 'HunterRenfroe', 'DH', 'CIN', '140', '498', '60', '116', '31', '0', '20', '60', '44', '125', '0', '0', '.233', '.297',
'.416', '.713']
['110', 'CJAbrams', 'SS', 'WSH', '151', '563', '83', '138', '28', '6', '18', '64', '32', '118', '47', '4', '.245', '.300', '.41
```

```
2', '.712']
['111', 'CarlosCorrea', 'SS', 'MIN', '135', '514', '60', '118', '29', '2', '18', '65', '59', '131', '0', '0', '.230', '.312', '.
399', '.711']
['111', 'JeffMcNeil', '2B', 'NYM', '156', '585', '75', '158', '25', '4', '10', '55', '39', '65', '10', '0', '.270', '.333', '.37
8', '.711']
['113', 'StevenKwan', 'LF', 'CLE', '158', '638', '93', '171', '36', '7', '5', '54', '70', '75', '21', '3', '.268', '.340', '.37
0', '.710']
['114', 'TommyEdman', '2B', 'STL', '137', '479', '69', '119', '25', '4', '13', '47', '35', '84', '27', '4', '.248', '.307', '.39
9', '.706']
['115', 'JeremyPeña', 'SS', 'HOU', '150', '577', '81', '152', '32', '3', '10', '52', '43', '129', '13', '9', '.263', '.324', '.3
81', '.705']
['116', 'TyFrance', '1B', 'SEA', '158', '587', '79', '147', '32', '0', '12', '58', '43', '117', '1', '0', '.250', '.337', '.36
6', '.703']
['117', 'WhitMerrifield', '2B', 'TOR', '145', '547', '66', '149', '27', '0', '11', '67', '36', '101', '26', '10', '.272', '.31
8', '.382', '.700']
['118', 'TylerStephenson', 'C', 'CIN', '142', '465', '59', '113', '20', '2', '13', '56', '47', '135', '0', '1', '.243', '.317',
'.378', '.695']
['118', 'EzequielTovar', 'SS', 'COL', '153', '581', '79', '147', '37', '4', '15', '73', '25', '166', '11', '5', '.253', '.287',
'.408', '.695']
['120', 'DominicSmith', '1B', 'WSH', '153', '527', '57', '134', '21', '1', '12', '46', '47', '91', '1', '1', '.254', '.326', '.3
66', '.692']
['121', 'JakeCronenworth', '1B', 'SD', '127', '458', '54', '105', '24', '7', '10', '48', '46', '97', '6', '1', '.229', '.312',
'.378', '.690']
['122', 'JuricksonProfar', 'LF', 'SD', '125', '459', '55', '111', '27', '2', '9', '46', '50', '90', '1', '0', '.242', '.321', '.
368', '.689']
['123', 'AmedRosario', '2B', 'LAD', '142', '510', '70', '134', '25', '8', '6', '58', '29', '99', '15', '2', '.263', '.305', '.37
8', '.683']
['124', 'AndrewBenintendi', 'LF', 'CWS', '151', '562', '72', '147', '34', '2', '5', '45', '52', '89', '13', '2', '.262', '.326',
'.356', '.682']
['125', 'MaikelGarcia', '3B', 'KC', '123', '464', '59', '126', '20', '4', '4', '50', '38', '115', '23', '7', '.272', '.323', '.3
58', '.681']
['126', 'JoséAbreu', '1B', 'HOU', '141', '540', '62', '128', '23', '1', '18', '90', '42', '130', '0', '1', '.237', '.296', '.38
3', '.679']
['127', 'DaultonVarsho', 'LF', 'TOR', '158', '527', '65', '116', '23', '3', '20', '61', '45', '135', '16', '7', '.220', '.285',
'.389', '.674']
['128', 'TrentGrisham', 'CF', 'SD', '153', '469', '67', '93', '31', '1', '13', '50', '75', '154', '15', '3', '.198', '.315', '.3
52', '.667']
['129', 'AnthonyVolpe', 'SS', 'NYY', '159', '541', '62', '113', '23', '4', '21', '60', '52', '167', '24', '5', '.209', '.283',
'.383', '.666']
['130', 'ZachMcKinstry', '3B', 'DET', '148', '464', '60', '107', '21', '4', '9', '35', '44', '113', '16', '6', '.231', '.302',
'.351', '.653']
['131', 'EnriqueHernández', '2B', 'LAD', '140', '465', '57', '110', '23', '0', '11', '61', '34', '97', '4', '1', '.237', '.289',
'.357', '.646']
['132', 'MylesStraw', 'CF', 'CLE', '147', '462', '52', '110', '18', '3', '1', '29', '42', '97', '20', '6', '.238', '.301', '.29
```

```
7', '.598']
['133', 'JavierBáez', 'SS', 'DET', '136', '510', '58', '113', '18', '4', '9', '59', '24', '125', '12', '0', '.222', '.267', '.32
5', '.592']
['134', 'TimAnderson', 'SS', 'CWS', '123', '493', '52', '121', '18', '2', '1', '25', '26', '122', '13', '2', '.245', '.286', '.2
96', '.582']
```

In [29]:
```
## adding two more element in headers
headers2.insert(0, 'Rank')
headers2.insert(2, 'Position')

## Create a dataframe with 'headers' as a column and 'body' as the content.
df_mlb_2023_regular = pd.DataFrame(body2)
df_mlb_2023_regular.columns = headers2
df_mlb_2023_regular = df_mlb_2023_regular.set_index('Rank')

## final data frame (expected size should be 130, 19 - If smaller, run the previous cell again)
df_mlb_2023_regular
```

Out[29]:

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
1	ShoheiOhtani	DH	LAA	135	497	102	151	26	8	44	95	91	143	20	6	.304	.412	.654	1.066
2	CoreySeager	SS	TEX	119	477	88	156	42	0	33	96	49	88	2	1	.327	.390	.623	1.013
3	RonaldAcuña	RF	ATL	159	643	149	217	35	4	41	106	80	84	73	14	.337	.416	.596	1.012
4	MattOlson	1B	ATL	162	608	127	172	27	3	54	139	104	167	1	0	.283	.389	.604	.993
5	MookieBetts	RF	LAD	152	584	126	179	40	1	39	107	96	107	14	3	.307	.408	.579	.987
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
130	ZachMcKinstry	3B	DET	148	464	60	107	21	4	9	35	44	113	16	6	.231	.302	.351	.653
131	EnriqueHernández	2B	LAD	140	465	57	110	23	0	11	61	34	97	4	1	.237	.289	.357	.646
132	MylesStraw	CF	CLE	147	462	52	110	18	3	1	29	42	97	20	6	.238	.301	.297	.598
133	JavierBáez	SS	DET	136	510	58	113	18	4	9	59	24	125	12	0	.222	.267	.325	.592
134	TimAnderson	SS	CWS	123	493	52	121	18	2	1	25	26	122	13	2	.245	.286	.296	.582

134 rows × 19 columns

In [30]:
```
## drive already mounted at /content/drive
## conneting google drive to colab
#from google.colab import drive
#drive.mount('/content/drive')
```

In [31]:
```
## save data frame as xlsx file
df_mlb_2023_regular = df_mlb_2023_regular.to_excel('/content/sample_data/MLB2023_Regular.xlsx', index=False)
```

ML prediction for Regular Season 2023

- Using the crawled data, we will run a machine learning task
- For simple tutorial, we'll see how to predict the number of homeruns from other stats

In [32]:
```
df_mlb_2023_regular = pd.read_excel('/content/sample_data/MLB2023_Regular.xlsx')
X_df_mlb_2023_regular = df_mlb_2023_regular.drop('HR', axis=1) ## delete HR data (it will be answer) from data frame
X_df_mlb_2023_regular = X_df_mlb_2023_regular.drop(['PLAYER', 'Position', 'TEAM'], axis=1) ## delete non-floating number data
X_df_mlb_2023_regular
```

Out[32]:

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
0	135	497	102	151	26	8	95	91	143	20	6	0.304	0.412	0.654	1.066
1	119	477	88	156	42	0	96	49	88	2	1	0.327	0.390	0.623	1.013
2	159	643	149	217	35	4	106	80	84	73	14	0.337	0.416	0.596	1.012
3	162	608	127	172	27	3	139	104	167	1	0	0.283	0.389	0.604	0.993
4	152	584	126	179	40	1	107	96	107	14	3	0.307	0.408	0.579	0.987
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
129	148	464	60	107	21	4	35	44	113	16	6	0.231	0.302	0.351	0.653
130	140	465	57	110	23	0	61	34	97	4	1	0.237	0.289	0.357	0.646
131	147	462	52	110	18	3	29	42	97	20	6	0.238	0.301	0.297	0.598
132	136	510	58	113	18	4	59	24	125	12	0	0.222	0.267	0.325	0.592
133	123	493	52	121	18	2	25	26	122	13	2	0.245	0.286	0.296	0.582

134 rows × 15 columns

In [33]:
```
y_df_mlb_2023_regular = df_mlb_2023_regular['HR']## setting HR data as y (target of prediction)
y_df_mlb_2023_regular
```

Out[33]:
```
0      44
1      33
2      41
3      54
4      39
       ..
129     9
130    11
131     1
132     9
133     1
Name: HR, Length: 134, dtype: int64
```

In [34]:
```
from sklearn.model_selection import train_test_split
## define train data and test data
```

```
X_train2, X_test2, y_train2, y_test2 = train_test_split(X_df_mlb_2023_regular,y_df_mlb_2023_regular, test_size=0.2, random_state=
print(X_train2.shape, X_test2.shape, y_train2.shape, y_test2.shape)
```

```
(107, 15) (27, 15) (107,) (27,)
```

In [35]:
```
from sklearn.preprocessing import StandardScaler
## scale the data
scalr = StandardScaler()
X_train2 = scalr.fit_transform(X_train2)
X_test2 = scalr.transform(X_test2)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
```

In [37]:
```
## train Linear Regression and Ridge Regression model and check the errors
reg1  = LinearRegression()
reg1.fit(X_train2, y_train2)
pred_train1 = reg1.predict(X_train2)
pred_val1 = reg1.predict(X_test2)
mse_train1 = mean_squared_error(y_train2, pred_train1)
mse_val1 = mean_squared_error(y_test2, pred_val1)

reg2 = Ridge()
reg2.fit(X_train2, y_train2)
pred_train2 = reg2.predict(X_train2)
pred_val2 = reg2.predict(X_test2)
mse_train2 = mean_squared_error(y_train2, pred_train2)
mse_val2 = mean_squared_error(y_test2, pred_val2)

print("1. Linear Regresion\t, train = %.4f, val = %.4f" %(mse_train1, mse_val1))
print("2. Ridge\t\t, train = %.4f, val = %.4f" %(mse_train2, mse_val2))
```

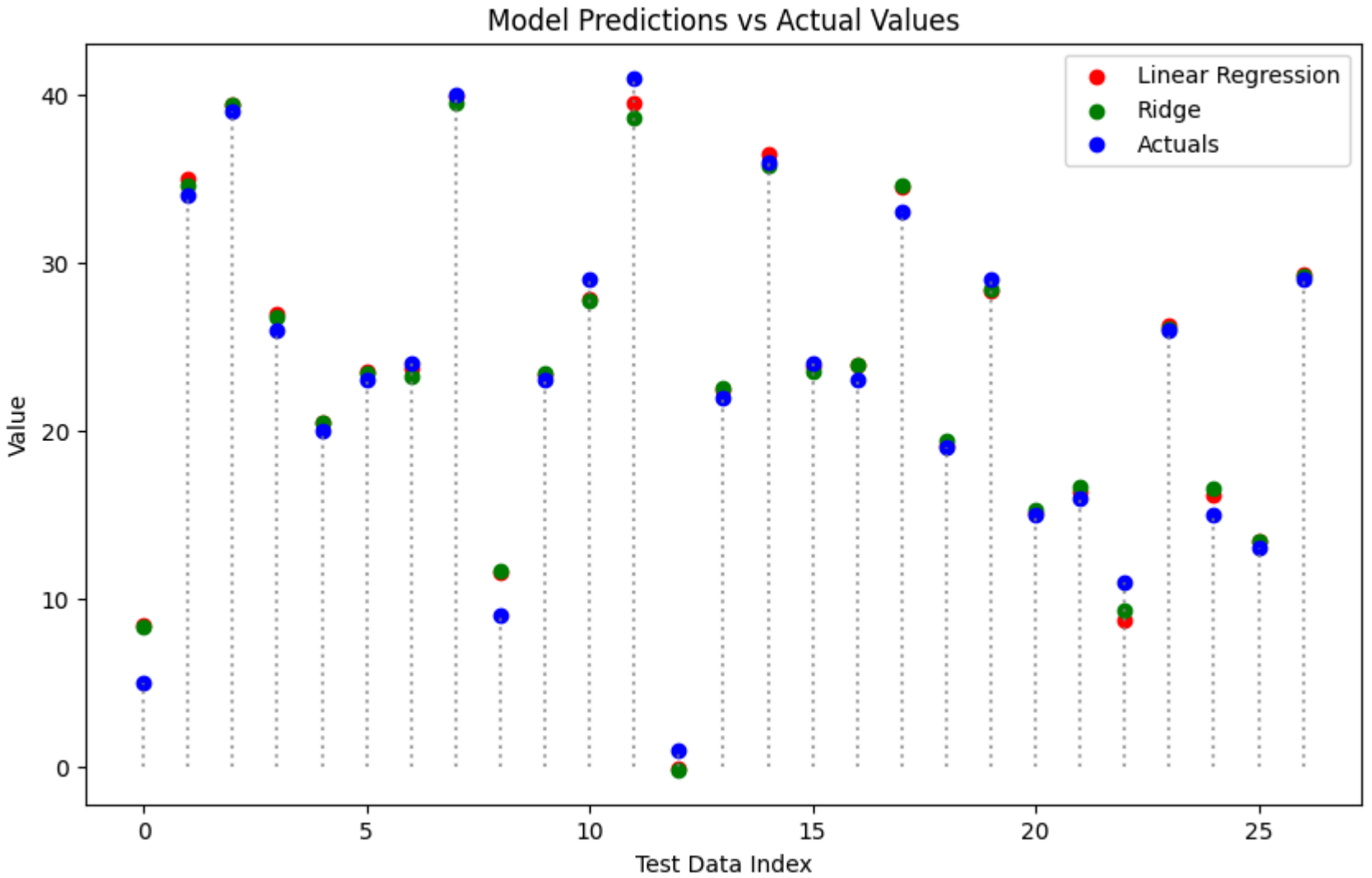
```
1. Linear Regresion     , train = 0.5532, val = 1.3544
2. Ridge                , train = 0.6128, val = 1.4670
```

Data Visualization for Regular Season 2023

Here is the sample code to visualize the prediction results.

In [38]:
```
import matplotlib.pyplot as plt
```

```
y_test2 = y_test2.astype(int).values

# Drawing scatter plots
plt.figure(figsize=(10, 6))
plt.scatter(range(len(pred_val1)), pred_val1, color='r', label='Linear Regression')
plt.scatter(range(len(pred_val2)), pred_val2, color='g', label='Ridge')
plt.scatter(range(len(y_test2)), y_test2, color='b', label='Actuals')

for i in range(len(y_test2)):
    plt.vlines(x=i, ymin=0, ymax=y_test2[i], colors='gray', linestyles='dotted', alpha=0.7)

plt.xlabel('Test Data Index')
plt.ylabel('Value')
plt.title('Model Predictions vs Actual Values')
plt.legend()

plt.show()
```



In [39]:
```
df_mlb_2023_regular['HR'] = df_mlb_2023_regular['HR'].astype('int')
df_mlb_2023_regular[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
```

Out[39]:

	PLAYER	HR
0	MattOlson	54
1	KyleSchwarber	47
2	PeteAlonso	46
3	ShoheiOhtani	44
4	RonaldAcuña	41
5	MarcellOzuna	40
6	MookieBetts	39
7	AdolisGarcía	39
8	LuisRobert	38
9	AustinRiley	37
10	MaxMuncy	36
11	JorgeSoler	36
12	JuanSoto	35
13	JakeBurger	34
14	ChristianWalker	33
15	RafaelDevers	33
16	OzzieAlbies	33
17	CoreySeager	33
18	JulioRodríguez	32
19	FranciscoLindor	31

Compare MLB 2022 and MLB 2023

In [40]:
```
df_mlb['HR'] = df_mlb['HR'].astype('int') # MLB 2022
df_mlb_2023_regular['HR'] = df_mlb_2023_regular['HR'].astype('int') # MLB 2023
```

```
# Sort top 20 players with highest HR in each dataframe
top_20_mlb = df_mlb[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
top_20_mlb_2023_regular = df_mlb_2023_regular[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20

# Concatenate dataframes
comparison_df = pd.concat([top_20_mlb, top_20_mlb_2023_regular], axis=1, keys=['MLB 2022', 'MLB 2023'])

# Display the comparison dataframe
comparison_df
#print(comparison_df)
```

Out[40]:

	MLB 2022		MLB 2023	
	PLAYER	HR	PLAYER	HR
0	AaronJudge	62	MattOlson	54
1	KyleSchwarber	46	KyleSchwarber	47
2	PeteAlonso	40	PeteAlonso	46
3	AustinRiley	38	ShoheiOhtani	44
4	YordanAlvarez	37	RonaldAcuña	41
5	ChristianWalker	36	MarcellOzuna	40
6	RowdyTellez	35	MookieBetts	39
7	MookieBetts	35	AdolisGarcía	39
8	PaulGoldschmidt	35	LuisRobert	38
9	ShoheiOhtani	34	AustinRiley	37
10	MattOlson	34	MaxMuncy	36
11	CoreySeager	33	JorgeSoler	36
12	AnthonySantander	33	JuanSoto	35
13	MannyMachado	32	JakeBurger	34
14	AnthonyRizzo	32	ChristianWalker	33
15	VladimirGuerrero	32	RafaelDevers	33
16	EugenioSuárez	31	OzzieAlbies	33
17	WillyAdames	31	CoreySeager	33
18	KyleTucker	30	JulioRodríguez	32
19	NolanArenado	30	FranciscoLindor	31

In [41]:
```
# Extract top 20 players from df_mlb and df_mlb1
top_20_mlb = df_mlb[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
top_20_mlb_2023_regular = df_mlb_2023_regular[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20
```

```
# Find same players between df_mlb and df_mlb1
common_players = pd.merge(top_20_mlb, top_20_mlb_2023_regular, on='PLAYER', suffixes=('_mlb_2022', '_mlb_2023'), how='inner')

#print(common_players)
common_players
```

Out[41]:

	PLAYER	HR_mlb_2022	HR_mlb_2023
0	KyleSchwarber	46	47
1	PeteAlonso	40	46
2	AustinRiley	38	37
3	ChristianWalker	36	33
4	MookieBetts	35	39
5	ShoheiOhtani	34	44
6	MattOlson	34	54
7	CoreySeager	33	33

Prediction by using data from 2023 Regular Season

In [42]:

```
df_mlb_2023_regular['wOBA'] = df_mlb_2023_regular['OBP'] + df_mlb_2023_regular['SLG']  # On-base plus slugging

features = ['wOBA']
X = df_mlb_2023_regular[features]
y = df_mlb_2023_regular['HR']  # Home Runs

# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model selection and training
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Prediction for 2024 season
y_pred = model.predict(X_test)
```

```
# Top 20 predicted home run hitters (assuming y_pred represents predictions)
top_20_idx = y_pred.argsort()[:20]  # Get indices of top 20 predictions
top_20_players = df_mlb_2023_regular.iloc[top_20_idx]['PLAYER'].to_list()

print("Top 20 predicted home run hitters for 2024:")
for player in top_20_players:
  print(player)
```

```
Top 20 predicted home run hitters for 2024:
ChristianWalker
JuanSoto
OzzieAlbies
KetelMarte
LuisRobert
MarcellOzuna
BryceHarper
YandyDíaz
ShoheiOhtani
MookieBetts
KyleTucker
TristonCasas
RonaldAcuña
FreddieFreeman
AustinRiley
CoreySeager
CodyBellinger
JoséRamírez
RafaelDevers
IsaacParedes
```

- Using more features

In [43]:
```
def create_features(data):
  data['wOBA'] = data['OBP'] + data['SLG']  # On-base plus slugging
  data['ISO'] = data['SLG'] - data['AVG']  # Isolated slugging
  data['PA'] = data['H'] + data['BB'] + data['AB']  # Plate Appearance.
  data['K%'] = (data['SO'] / data['PA']) * 100  # Strikeout rate (potential for outs)
  data['BB%'] = (data['BB'] / data['PA']) * 100  # Walk rate (on-base opportunities)
  return data

df_mlb_predict_2023 = create_features(df_mlb_2023_regular.copy())  # Avoid modifying original DataFrame
```

```
# Filter relevant features
features = ['wOBA', 'ISO', 'K%', 'BB%']
X = df_mlb_predict_2023[features]
Y = df_mlb_predict_2023['HR']  # Home runs

# Train-test split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

# Model selection and training
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, Y_train)

# Prediction for 2024 season
Y_pred = model.predict(X_test)

# Top 20 predicted home run leaders (assuming y_pred represents predictions)
top_20_idx = Y_pred.argsort()[:20]  # Get top 20 predictions
top_20_players = df_mlb_predict_2023.iloc[top_20_idx]['PLAYER'].to_list()

print("Top 20 predicted home run leaders for 2024:")
for player in top_20_players:
  print(player)
```

```
Top 20 predicted home run leaders for 2024:
LuisRobert
BryceHarper
RafaelDevers
ShoheiOhtani
OzzieAlbies
AustinRiley
MookieBetts
MarcellOzuna
JuanSoto
RonaldAcuña
ChristianWalker
KetelMarte
SeiyaSuzuki
TristonCasas
FreddieFreeman
LuisArraez
CoreySeager
CodyBellinger
JoséRamírez
MattOlson
```

- Addtional code

In [44]:
```python
df_mlb_predict1_2023 = create_features(df_mlb_2023_regular.copy())

# Features
X = df_mlb_predict1_2023[["OPS", "ISO"]]
y = df_mlb_predict1_2023["HR"]

from sklearn.feature_selection import SelectKBest, f_classif
selector = SelectKBest(f_classif, k=2)  # You can choose important features
X_selected = selector.fit_transform(X, y)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.2)

# Model selection
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
# Prediction for 2024 season
y_pred = model.predict(X_test)

# Top 20 predicted home run leaders (assuming y_pred represents predictions)
top_20_idx = y_pred.argsort()[:20]  # Get top 20 predictions
top_20_players = df_mlb_predict1_2023.iloc[top_20_idx]['PLAYER'].to_list()

print("Top 20 predicted home run leaders for 2024:")
for player in top_20_players:
  print(player)
```

```
Top 20 predicted home run leaders for 2024:
JuanSoto
ShoheiOhtani
MookieBetts
CorbinCarroll
JorgeSoler
LuisRobert
RonaldAcuña
JoséRamírez
KyleTucker
KetelMarte
MarcellOzuna
TristonCasas
BryceHarper
RafaelDevers
FreddieFreeman
SeiyaSuzuki
MattOlson
IsaacParedes
AustinRiley
LuisArraez
```

Predict home runs for players

- I've already compiled the Top 20 Predicted Home Run Leaders for 2024. Now let's look at their HR statistics.

In [45]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import SelectKBest, f_classif
```

- Predict HR stats for one player

In [46]:
```
# Load data from your Excel file
df_mlb_2023_regular_test = pd.read_excel("/content/sample_data/MLB2023_Regular.xlsx")

# Feature engineering
df_mlb_2023_regular_test["wOBA"] = df_mlb_2023_regular_test["OBP"] + df_mlb_2023_regular_test["SLG"]
df_mlb_2023_regular_test["ISO"] = df_mlb_2023_regular_test["SLG"] - df_mlb_2023_regular_test["AVG"]
df_mlb_2023_regular_test["PA"] = df_mlb_2023_regular_test["H"] + df_mlb_2023_regular_test["BB"] + df_mlb_2023_regular_test["AB"]
df_mlb_2023_regular_test["K%"] = (df_mlb_2023_regular_test["SO"] / df_mlb_2023_regular_test["PA"]) * 100  # Strikeout Rate
df_mlb_2023_regular_test["BB%"] = (df_mlb_2023_regular_test["BB"] / df_mlb_2023_regular_test["PA"]) * 100  # Walk Rate


# Feature selection
X = df_mlb_2023_regular_test[["wOBA", "ISO", "K%", "BB%", "PA"]]  # Add advanced features here
y = df_mlb_2023_regular_test["HR"]
selector = SelectKBest(f_classif, k=5)  # Choose number of features
X_selected = selector.fit_transform(X, y)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_selected, y, test_size=0.2)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Model selection and hyperparameter tuning
model = RandomForestRegressor(n_estimators=100, random_state=42)

model.fit(X_train, y_train)

# Prediction
y_pred = model.predict(X_test)

# Evaluate model performance (additional)
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Predict HR stats for one player
new_player_name = "ChristianWalker"  # Replace with desired player name
```

```
new_player_data = df_mlb_2023_regular_test[df_mlb_2023_regular_test["PLAYER"] == new_player_name]  # Select player data
new_player_features = new_player_data[["wOBA", "ISO", "K%", "BB%", "PA"]]
new_player_features_selected = selector.transform(new_player_features)
new_player_features_scaled = scaler.transform(new_player_features_selected)  # If using standardization
predicted_hr = model.predict(new_player_features_scaled)[0]  # Assuming single prediction

print(f"Predicted HR for {new_player_name}: {predicted_hr:.2f}")
```

```
Mean Squared Error: 14.971540740740737
Predicted HR for ChristianWalker: 31.51
```

- Prediction for a list of new players

In [47]:

```
# Prediction for a list of new players
new_player_names = ["ChristianWalker", "ShoheiOhtani", "AustinRiley", "BrandonNimmo", "JorgeSoler",
                    "MarcellOzuna", "KyleTucker", "SeiyaSuzuki", "JuanSoto", "CodyBellinger",
                    "FreddieFreeman", "KetelMarte", "RafaelDevers", "MookieBetts", "OzzieAlbies",
                    "LuisRobert", "TristonCasas", "MattOlson", "RonaldAcuña", "JoséRamírez"]  # List of player names

predicted_hr = []
for player_name in new_player_names:
  new_player_data = df_mlb_2023_regular_test[df_mlb_2023_regular_test["PLAYER"] == player_name]  # Select player data
  if not new_player_data.empty:  # Check if player data exists
      new_player_features = new_player_data[["wOBA", "ISO", "K%", "BB%", "PA"]]
      new_player_features_selected = selector.transform(new_player_features)
      new_player_features_scaled = scaler.transform(new_player_features_selected)  # If using standardization
      predicted_hr.append(model.predict(new_player_features_scaled)[0])  # Assuming single prediction
  else:
      print(f"Player data not found for {player_name}")
      predicted_hr.append(None)

sorted_results = sorted(zip(predicted_hr, new_player_names), reverse=True)
predicted_hr, new_player_names = zip(*sorted_results)  # Sorted results

print("Predicted HRs:")
for i, name in enumerate(new_player_names):
  if predicted_hr[i] is not None:  # Check if prediction exists
    print(f"{i+1}. {name}: {predicted_hr[i]:.2f}")  # Print name and predicted HR
  else:
    print(f"{i+1}. {name}: Data not found")  # Inform about missing data
```

```
Predicted HRs:
1. MattOlson: 50.57
2. ShoheiOhtani: 46.08
3. MarcellOzuna: 41.42
4. RonaldAcuña: 40.70
5. LuisRobert: 39.39
6. MookieBetts: 39.21
7. JorgeSoler: 37.44
8. JuanSoto: 33.51
9. ChristianWalker: 31.51
10. OzzieAlbies: 31.35
11. AustinRiley: 30.96
12. RafaelDevers: 30.11
13. FreddieFreeman: 29.63
14. KyleTucker: 29.06
15. KetelMarte: 27.25
16. CodyBellinger: 26.52
17. TristonCasas: 24.63
18. BrandonNimmo: 24.12
19. JoséRamírez: 24.09
20. SeiyaSuzuki: 21.36
```

In [48]:

```
# Create a DataFrame
results_df = pd.DataFrame({"Player": new_player_names, "Predicted HR": predicted_hr})

mlb_2023_regular = df_mlb_2023_regular[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)

# Concatenate dataframes
concatenate_df = pd.concat([results_df, mlb_2023_regular], axis=1, keys=['Predicted MLB results', 'MLB 2023'])
concatenate_df
```

Out[48]:

	Predicted MLB results		MLB 2023	
	Player	Predicted HR	PLAYER	HR
0	MattOlson	50.57	MattOlson	54
1	ShoheiOhtani	46.08	KyleSchwarber	47
2	MarcellOzuna	41.42	PeteAlonso	46
3	RonaldAcuña	40.70	ShoheiOhtani	44
4	LuisRobert	39.39	RonaldAcuña	41
5	MookieBetts	39.21	MarcellOzuna	40
6	JorgeSoler	37.44	MookieBetts	39
7	JuanSoto	33.51	AdolisGarcía	39
8	ChristianWalker	31.51	LuisRobert	38
9	OzzieAlbies	31.35	AustinRiley	37
10	AustinRiley	30.96	MaxMuncy	36
11	RafaelDevers	30.11	JorgeSoler	36
12	FreddieFreeman	29.63	JuanSoto	35
13	KyleTucker	29.06	JakeBurger	34
14	KetelMarte	27.25	ChristianWalker	33
15	CodyBellinger	26.52	RafaelDevers	33
16	TristonCasas	24.63	OzzieAlbies	33
17	BrandonNimmo	24.12	CoreySeager	33
18	JoséRamírez	24.09	JulioRodríguez	32
19	SeiyaSuzuki	21.36	FranciscoLindor	31

Addtional training of MLB 2024 Regular Season

There's not enough information there, so few games have been played. But let there be information about MLB 2024 Regular Season.

- Data crawling for Regular Season 2024

In [49]:
```
## set driver url to crawler

driver.get('https://www.mlb.com/stats/regular-season')

## find element from web
table3 = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[1]/div/table')

header_items = table3.find_elements(By.XPATH, './thead/tr/th')
headers3 = [h.text for h in header_items]

## print properties
print(headers3)

## find element from web
body3 = []
while True:
  time.sleep(2)

  ## crawl from the first page and save it using driver.find_element method
  body_items = table3.find_elements(By.XPATH, './tbody/tr')
  for b in body_items:
    row_title = b.find_elements(By.XPATH, './th')
    row_items = b.find_elements(By.XPATH, './td')
    row = [r.text for r in row_title] + [r.text for r in row_items]

    ## we tokenize it and make it an individual element.
    row = row[0].split('\n') + row[1:]
    body3.append(row)
    print(row)
  try:
    cookie_button = driver.find_element(By.XPATH, '//*[@id="onetrust-accept-btn-handler"]')
    cookie_button.click()
  except:
    pass
  try:
    next_button = driver.find_element(By.XPATH, '//*[@id="stats-app-root"]/section/section/div[3]/div[2]/div/div/div[last()]/butt
    next_button.click()
```

```
  except:
    break
```

```
['PLAYER', 'TEAM', 'G', 'AB', 'R', 'H', '2B', '3B', 'HR', 'RBI', 'BB', 'SO', 'SB', 'CS', 'AVG', 'OBP', 'SLG', 'OPS']
['1', 'MookieBetts', 'SS', 'LAD', '2', '9', '2', '6', '1', '0', '1', '7', '2', '0', '0', '0', '.667', '.727', '1.111', '1.838']
['2', 'TylerWade', '3B', 'SD', '2', '5', '2', '3', '0', '0', '0', '1', '1', '1', '0', '0', '.600', '.667', '.600', '1.267']
['3', 'JakeCronenworth', '1B', 'SD', '2', '8', '2', '4', '0', '1', '0', '4', '0', '2', '0', '0', '.500', '.500', '.750', '1.25
0']
['4', 'WillSmith', 'C', 'LAD', '2', '10', '1', '5', '1', '0', '0', '2', '1', '2', '0', '0', '.500', '.545', '.600', '1.145']
['5', 'MannyMachado', 'DH', 'SD', '2', '7', '3', '1', '0', '0', '1', '3', '3', '1', '0', '0', '.143', '.400', '.571', '.971']
['6', 'XanderBogaerts', '2B', 'SD', '2', '9', '3', '4', '0', '0', '0', '3', '1', '2', '0', '0', '.444', '.500', '.444', '.944']
['7', 'MaxMuncy', '3B', 'LAD', '2', '9', '2', '3', '1', '0', '0', '0', '2', '5', '0', '0', '.333', '.455', '.444', '.899']
['8', 'LuisCampusano', 'C', 'SD', '2', '10', '2', '3', '2', '0', '0', '2', '0', '0', '0', '0', '.300', '.300', '.500', '.800']
['9', 'FreddieFreeman', '1B', 'LAD', '2', '6', '0', '1', '0', '0', '0', '0', '4', '3', '0', '0', '.167', '.545', '.167', '.712']
['10', 'FernandoTatis', 'RF', 'SD', '2', '8', '2', '2', '0', '0', '0', '1', '1', '1', '0', '0', '.250', '.400', '.250', '.650']
['11', 'JuricksonProfar', 'LF', 'SD', '2', '5', '0', '1', '0', '0', '0', '0', '2', '2', '0', '0', '.200', '.429', '.200', '.62
9']
['12', 'JacksonMerrill', 'CF', 'SD', '2', '8', '2', '2', '1', '0', '0', '0', '0', '3', '0', '0', '.250', '.250', '.375', '.625']
['13', 'JasonHeyward', 'RF', 'LAD', '2', '6', '3', '2', '0', '0', '0', '2', '0', '0', '0', '0', '.333', '.286', '.333', '.619']
['14', 'ShoheiOhtani', 'DH', 'LAD', '2', '10', '1', '3', '0', '0', '0', '2', '0', '0', '1', '0', '.300', '.273', '.300', '.573']
['15', 'TeoscarHernández', 'LF', 'LAD', '2', '8', '2', '1', '0', '0', '0', '0', '2', '4', '0', '0', '.125', '.300', '.125', '.42
5']
['16', 'GavinLux', '2B', 'LAD', '2', '10', '3', '2', '0', '0', '0', '0', '0', '1', '0', '0', '.200', '.200', '.200', '.400']
['16', 'JamesOutman', 'CF', 'LAD', '2', '6', '2', '0', '0', '0', '0', '1', '3', '0', '0', '0', '.000', '.400', '.000', '.400']
['18', 'Ha-SeongKim', 'SS', 'SD', '2', '7', '0', '0', '0', '0', '0', '1', '2', '0', '1', '0', '.000', '.200', '.000', '.200']
```

In [50]:

```
## adding two more element in headers
headers3.insert(0, 'Rank')
headers3.insert(2, 'Position')

## Create a dataframe with 'headers' as a column and 'body' as the content.
df_mlb_2024_regular = pd.DataFrame(body3)
df_mlb_2024_regular.columns = headers3
df_mlb_2024_regular = df_mlb_2024_regular.set_index('Rank')

## final data frame (expected size should be 130, 19 - If smaller, run the previous cell again)
df_mlb_2024_regular
```

Out[50]:

Rank	PLAYER	Position	TEAM	G	AB	R	H	2B	3B	HR	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
1	MookieBetts	SS	LAD	2	9	2	6	1	0	1	7	2	0	0	0	.667	.727	1.111	1.838
2	TylerWade	3B	SD	2	5	2	3	0	0	0	1	1	1	0	0	.600	.667	.600	1.267
3	JakeCronenworth	1B	SD	2	8	2	4	0	1	0	4	0	2	0	0	.500	.500	.750	1.250
4	WillSmith	C	LAD	2	10	1	5	1	0	0	2	1	2	0	0	.500	.545	.600	1.145
5	MannyMachado	DH	SD	2	7	3	1	0	0	1	3	3	1	0	0	.143	.400	.571	.971
6	XanderBogaerts	2B	SD	2	9	3	4	0	0	0	3	1	2	0	0	.444	.500	.444	.944
7	MaxMuncy	3B	LAD	2	9	2	3	1	0	0	0	2	5	0	0	.333	.455	.444	.899
8	LuisCampusano	C	SD	2	10	2	3	2	0	0	2	0	0	0	0	.300	.300	.500	.800
9	FreddieFreeman	1B	LAD	2	6	0	1	0	0	0	0	4	3	0	0	.167	.545	.167	.712
10	FernandoTatis	RF	SD	2	8	2	2	0	0	0	1	1	1	0	0	.250	.400	.250	.650
11	JuricksonProfar	LF	SD	2	5	0	1	0	0	0	0	2	2	0	0	.200	.429	.200	.629
12	JacksonMerrill	CF	SD	2	8	2	2	1	0	0	0	0	3	0	0	.250	.250	.375	.625
13	JasonHeyward	RF	LAD	2	6	3	2	0	0	0	2	0	0	0	0	.333	.286	.333	.619
14	ShoheiOhtani	DH	LAD	2	10	1	3	0	0	0	2	0	0	1	0	.300	.273	.300	.573
15	TeoscarHernández	LF	LAD	2	8	2	1	0	0	0	0	2	4	0	0	.125	.300	.125	.425
16	GavinLux	2B	LAD	2	10	3	2	0	0	0	0	0	1	0	0	.200	.200	.200	.400
16	JamesOutman	CF	LAD	2	6	2	0	0	0	0	1	3	0	0	0	.000	.400	.000	.400
18	Ha-SeongKim	SS	SD	2	7	0	0	0	0	0	1	2	0	1	0	.000	.200	.000	.200

In [51]:
```
## drive already mounted at /content/drive
## conneting google drive to colab
#from google.colab import drive
#drive.mount('/content/drive')
```

In [52]:
```
## save data frame as xlsx file
df_mlb_2024_regular = df_mlb_2024_regular.to_excel('/content/sample_data/MLB2024_Regular.xlsx', index=False)
```

ML prediction for Regular Season 2024

- Using the crawled data, we will run a machine learning task
- For simple tutorial, we'll see how to predict the number of homeruns from other stats

In [53]:
```
# The correct one is commetted! (df_mlb_2024_regular = pd.read_excel('/content/sample_data/MLB2024_Regular.xlsx'))
#df_mlb_2024_regular = pd.read_excel('/content/sample_data/MLB2024_Regular.xlsx')
df_mlb_2024_regular = pd.read_excel('/content/sample_data/MLB2023_Regular.xlsx')
X_df_mlb_2024_regular = df_mlb_2024_regular.drop('HR', axis=1) ## delete HR data (it will be answer) from data frame
X_df_mlb_2024_regular = X_df_mlb_2024_regular.drop(['PLAYER', 'Position', 'TEAM'], axis=1) ## delete non-floating number data
X_df_mlb_2024_regular
```

Out[53]:

	G	AB	R	H	2B	3B	RBI	BB	SO	SB	CS	AVG	OBP	SLG	OPS
0	135	497	102	151	26	8	95	91	143	20	6	0.304	0.412	0.654	1.066
1	119	477	88	156	42	0	96	49	88	2	1	0.327	0.390	0.623	1.013
2	159	643	149	217	35	4	106	80	84	73	14	0.337	0.416	0.596	1.012
3	162	608	127	172	27	3	139	104	167	1	0	0.283	0.389	0.604	0.993
4	152	584	126	179	40	1	107	96	107	14	3	0.307	0.408	0.579	0.987
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
129	148	464	60	107	21	4	35	44	113	16	6	0.231	0.302	0.351	0.653
130	140	465	57	110	23	0	61	34	97	4	1	0.237	0.289	0.357	0.646
131	147	462	52	110	18	3	29	42	97	20	6	0.238	0.301	0.297	0.598
132	136	510	58	113	18	4	59	24	125	12	0	0.222	0.267	0.325	0.592
133	123	493	52	121	18	2	25	26	122	13	2	0.245	0.286	0.296	0.582

134 rows × 15 columns

In [54]:
```
y_df_mlb_2024_regular = df_mlb_2024_regular['HR']## setting HR data as y (target of prediction)
y_df_mlb_2024_regular
```

Out[54]:
```
0      44
1      33
2      41
3      54
4      39
       ..
129     9
130    11
131     1
132     9
133     1
Name: HR, Length: 134, dtype: int64
```

In [55]:
```
from sklearn.model_selection import train_test_split
## define train data and test data
X_train3, X_test3, y_train3, y_test3 = train_test_split(X_df_mlb_2023_regular,y_df_mlb_2024_regular, test_size=0.2, random_state=
print(X_train3.shape, X_test3.shape, y_train3.shape, y_test3.shape)
```

```
(107, 15) (27, 15) (107,) (27,)
```

In [56]:
```
from sklearn.preprocessing import StandardScaler
## scale the data
scalr = StandardScaler()
X_train3 = scalr.fit_transform(X_train3)
X_test3 = scalr.transform(X_test3)
```

In [57]:
```
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.metrics import mean_squared_error
```

In [58]:
```
## train Linear Regression and Ridge Regression model and check the errors
reg1  = LinearRegression()
reg1.fit(X_train3, y_train3)
pred_train1 = reg1.predict(X_train3)
pred_val1 = reg1.predict(X_test3)
mse_train1 = mean_squared_error(y_train3, pred_train1)
mse_val1 = mean_squared_error(y_test3, pred_val1)

reg2 = Ridge()
reg2.fit(X_train3, y_train3)
pred_train2 = reg2.predict(X_train3)
pred_val2 = reg2.predict(X_test3)
mse_train2 = mean_squared_error(y_train3, pred_train2)
```

```
mse_val2 = mean_squared_error(y_test3, pred_val2)

print("1. Linear Regresion\t, train = %.4f, val = %.4f" %(mse_train1, mse_val1))
print("2. Ridge\t\t, train = %.4f, val = %.4f" %(mse_train2, mse_val2))
```

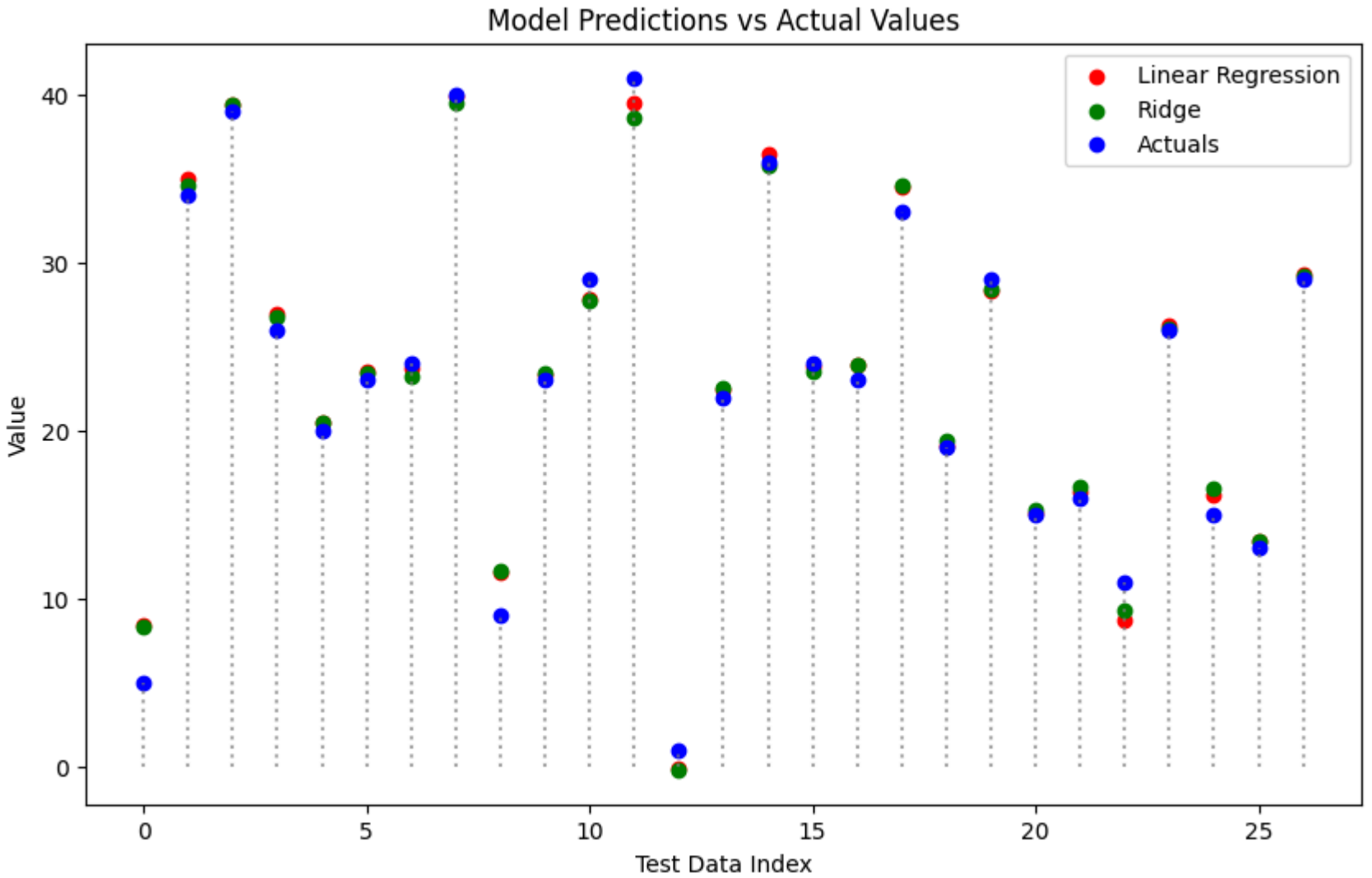
```
1. Linear Regresion     , train = 0.5532, val = 1.3544
2. Ridge                , train = 0.6128, val = 1.4670
```

Data Visualization for Regular Season 2023

Here is the sample code to visualize the prediction results.

In [59]:

```
import matplotlib.pyplot as plt

y_test3 = y_test3.astype(int).values

# Drawing scatter plots
plt.figure(figsize=(10, 6))
plt.scatter(range(len(pred_val1)), pred_val1, color='r', label='Linear Regression')
plt.scatter(range(len(pred_val2)), pred_val2, color='g', label='Ridge')
plt.scatter(range(len(y_test3)), y_test3, color='b', label='Actuals')

for i in range(len(y_test3)):
    plt.vlines(x=i, ymin=0, ymax=y_test3[i], colors='gray', linestyles='dotted', alpha=0.7)

plt.xlabel('Test Data Index')
plt.ylabel('Value')
plt.title('Model Predictions vs Actual Values')
plt.legend()

plt.show()
```



In [60]:
```
df_mlb_2024_regular['HR'] = df_mlb_2024_regular['HR'].astype('int')
df_mlb_2024_regular[['PLAYER', 'HR']].sort_values('HR', ascending=False).reset_index(drop=True).head(20)
```

Out[60]:

	PLAYER	HR
0	MattOlson	54
1	KyleSchwarber	47
2	PeteAlonso	46
3	ShoheiOhtani	44
4	RonaldAcuña	41
5	MarcellOzuna	40
6	MookieBetts	39
7	AdolisGarcía	39
8	LuisRobert	38
9	AustinRiley	37
10	MaxMuncy	36
11	JorgeSoler	36
12	JuanSoto	35
13	JakeBurger	34
14	ChristianWalker	33
15	RafaelDevers	33
16	OzzieAlbies	33
17	CoreySeager	33
18	JulioRodríguez	32
19	FranciscoLindor	31