Introduction to Analytics

Mohammed Saif Wasay (002815958)

Masters of Professional Studies in Informatics,

Northeastern University ALY 6000: Introduction to Analytics

Professor

Harpreet Sharma

January 21, 2024

Introduction and Key Findings

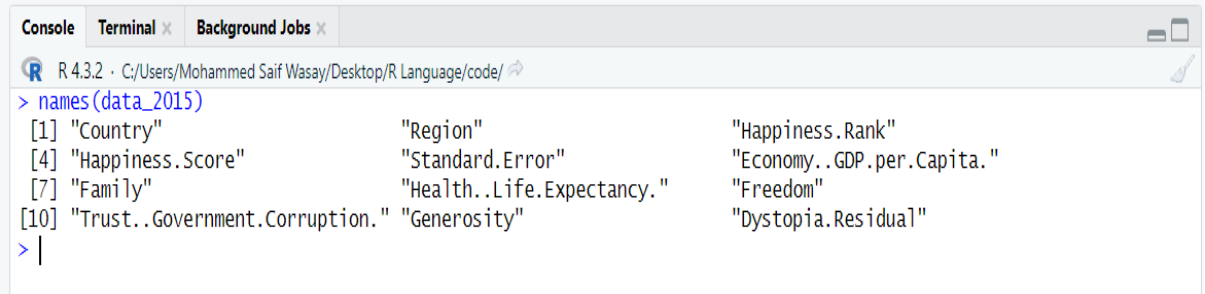
This report offers a comprehensive analysis of two datasets: the 2015 World Happiness Report and 1986 Major League Baseball statistics, utilizing the powerful R software for statistical analysis and visualization. The study starts with the World Happiness Report, focusing on variables like country and happiness score, employing methods such as data cleaning and sorting. It then delves into the baseball statistics, analyzing key performance metrics like hits and home runs, highlighting top players and award-eligible athletes. This report exemplifies the diverse applications of R in data analysis, providing a deep understanding of global happiness indices and sports performance metrics through detailed data exploration.

Assignment Part 1:

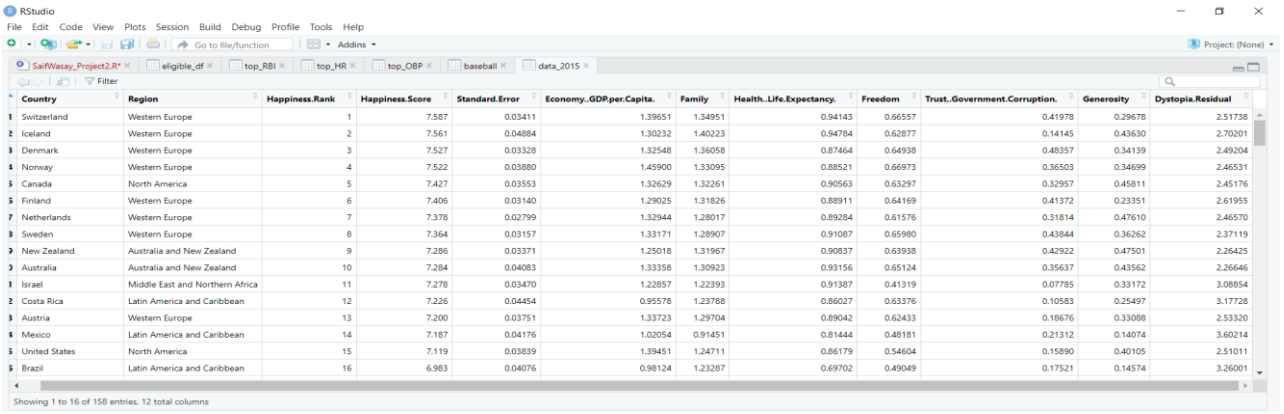
1.By using read.cse() function we are loading the "2015.csv" file into R and store it in a variable named data_2015. And then using Head () to display the first 6 rows of the dataset.

```
> head(data_2015)
      Country         Region Happiness.Rank Happiness.Score Standard.Error Economy..GDP.per.Capita.  Family
1 Switzerland Western Europe              1           7.587        0.03411                  1.39651 1.34951
2     Iceland Western Europe              2           7.561        0.04884                  1.30232 1.40223
3     Denmark Western Europe              3           7.527        0.03328                  1.32548 1.36058
4      Norway Western Europe              4           7.522        0.03880                  1.45900 1.33095
5      Canada  North America              5           7.427        0.03553                  1.32629 1.32261
6     Finland Western Europe              6           7.406        0.03140                  1.29025 1.31826
  Health..Life.Expectancy. Freedom Trust..Government.Corruption. Generosity Dystopia.Residual
1                  0.94143 0.66557                       0.41978    0.29678           2.51738
2                  0.94784 0.62877                       0.14145    0.43630           2.70201
3                  0.87464 0.64938                       0.48357    0.34139           2.49204
4                  0.88521 0.66973                       0.36503    0.34699           2.46531
5                  0.90563 0.63297                       0.32957    0.45811           2.45176
6                  0.88911 0.64169                       0.41372    0.23351           2.61955
> |
```

Here, names () builtin function of R displays various variables or attribustes of dataset. Retrieving and displaying the names of all columns in the data_2015 dataset.



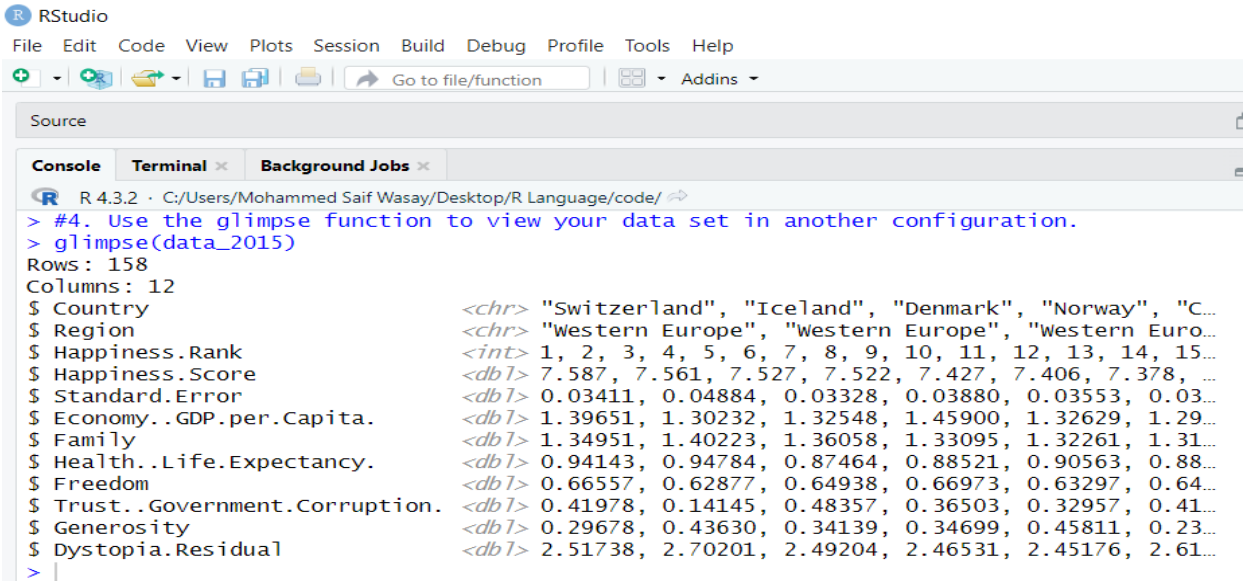
2.Opening the data_2015 dataset in a separate tab for a detailed view.



	Country	Region	Happiness.Rank	Happiness.Score	Standard.Error	Economy..GDP.per.Capita.	Family	Health..Life.Expectancy.	Freedom	Trust..Government.Corruption.	Generosity	Dystopia.Residual
1	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
2	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
3	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
4	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
5	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
6	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61955
7	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46570
8	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	0.43844	0.36262	2.37119
9	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	0.42922	0.47501	2.26425
10	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	0.35637	0.43562	2.26646
11	Israel	Middle East and Northern Africa	11	7.278	0.03470	1.22857	1.22393	0.91387	0.41319	0.07785	0.33172	3.08854
12	Costa Rica	Latin America and Caribbean	12	7.226	0.04454	0.95578	1.23788	0.86027	0.63376	0.10583	0.25497	3.17728
13	Austria	Western Europe	13	7.200	0.03751	1.33723	1.29704	0.89042	0.62433	0.18676	0.33088	2.53320
14	Mexico	Latin America and Caribbean	14	7.187	0.04176	1.02054	0.91451	0.81444	0.48181	0.21312	0.14074	3.60214
15	United States	North America	15	7.119	0.03839	1.39451	1.24711	0.86179	0.54604	0.15890	0.40105	2.51011
16	Brazil	Latin America and Caribbean	16	6.983	0.04076	0.98124	1.23287	0.69702	0.49049	0.17521	0.14574	3.26001

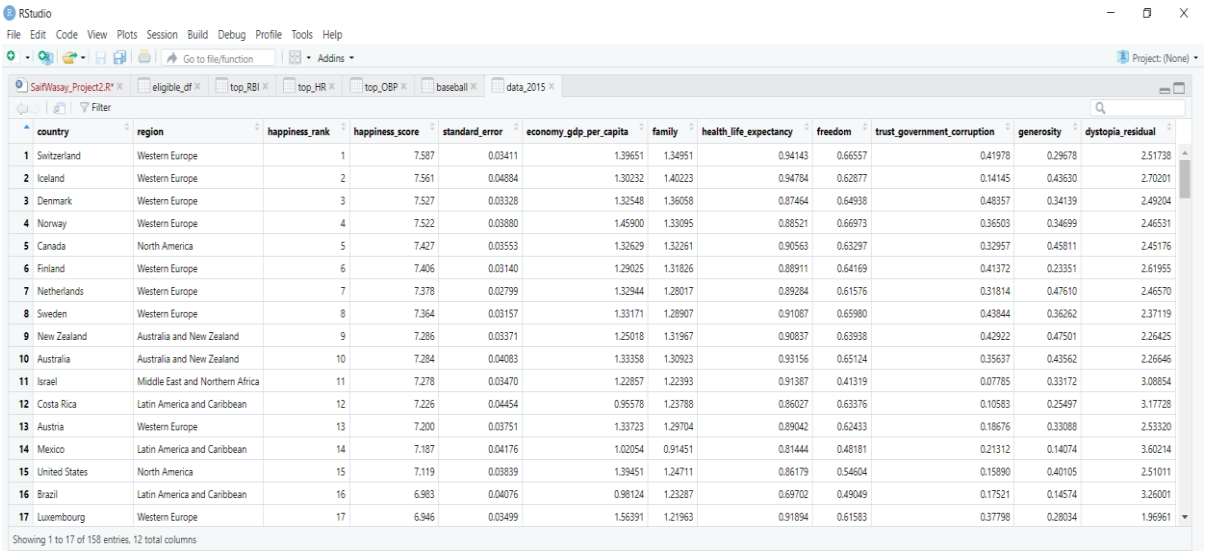
Showing 1 to 16 of 158 entries, 12 total columns

3.Using the glimpse function to get a compact overview of the data_2015 dataset.



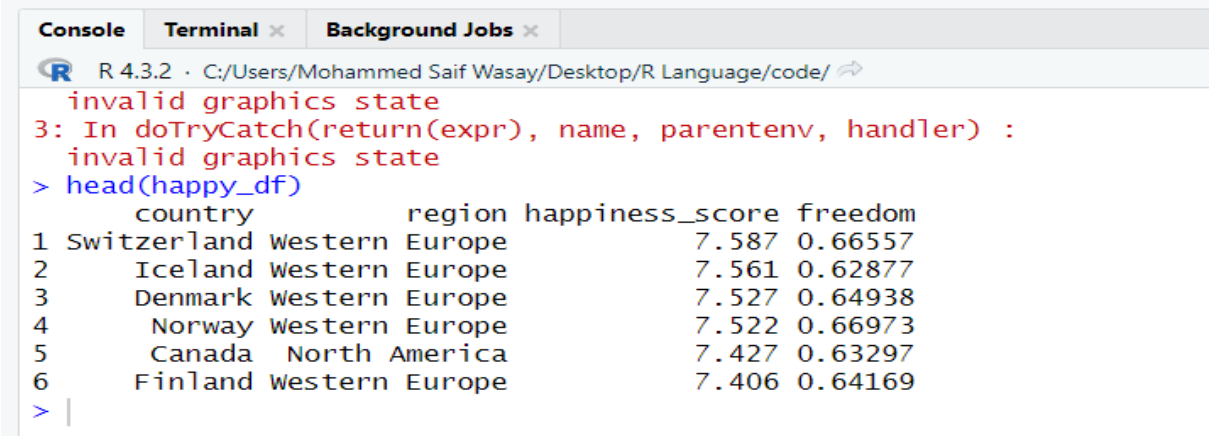
```
RStudio
File  Edit  Code  View  Plots  Session  Build  Debug  Profile  Tools  Help
Go to file/function   Addins
Source
Console   Terminal   Background Jobs
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
> #4. Use the glimpse function to view your data set in another configuration.
> glimpse(data_2015)
Rows: 158
Columns: 12
$ Country                      <chr> "Switzerland", "Iceland", "Denmark", "Norway", "C…
$ Region                       <chr> "Western Europe", "Western Europe", "Western Euro…
$ Happiness.Rank               <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15…
$ Happiness.Score              <dbl> 7.587, 7.561, 7.527, 7.522, 7.427, 7.406, 7.378, …
$ Standard.Error               <dbl> 0.03411, 0.04884, 0.03328, 0.03880, 0.03553, 0.03…
$ Economy..GDP.per.Capita.     <dbl> 1.39651, 1.30232, 1.32548, 1.45900, 1.32629, 1.29…
$ Family                       <dbl> 1.34951, 1.40223, 1.36058, 1.33095, 1.32261, 1.31…
$ Health..Life.Expectancy.     <dbl> 0.94143, 0.94784, 0.87464, 0.88521, 0.90563, 0.88…
$ Freedom                      <dbl> 0.66557, 0.62877, 0.64938, 0.66973, 0.63297, 0.64…
$ Trust..Government.Corruption. <dbl> 0.41978, 0.14145, 0.48357, 0.36503, 0.32957, 0.41…
$ Generosity                   <dbl> 0.29678, 0.43630, 0.34139, 0.34699, 0.45811, 0.23…
$ Dystopia.Residual            <dbl> 2.51738, 2.70201, 2.49204, 2.46531, 2.45176, 2.61…
>
```

4. Here, Installing and using the janitor package to clean up the column names in data_2015.



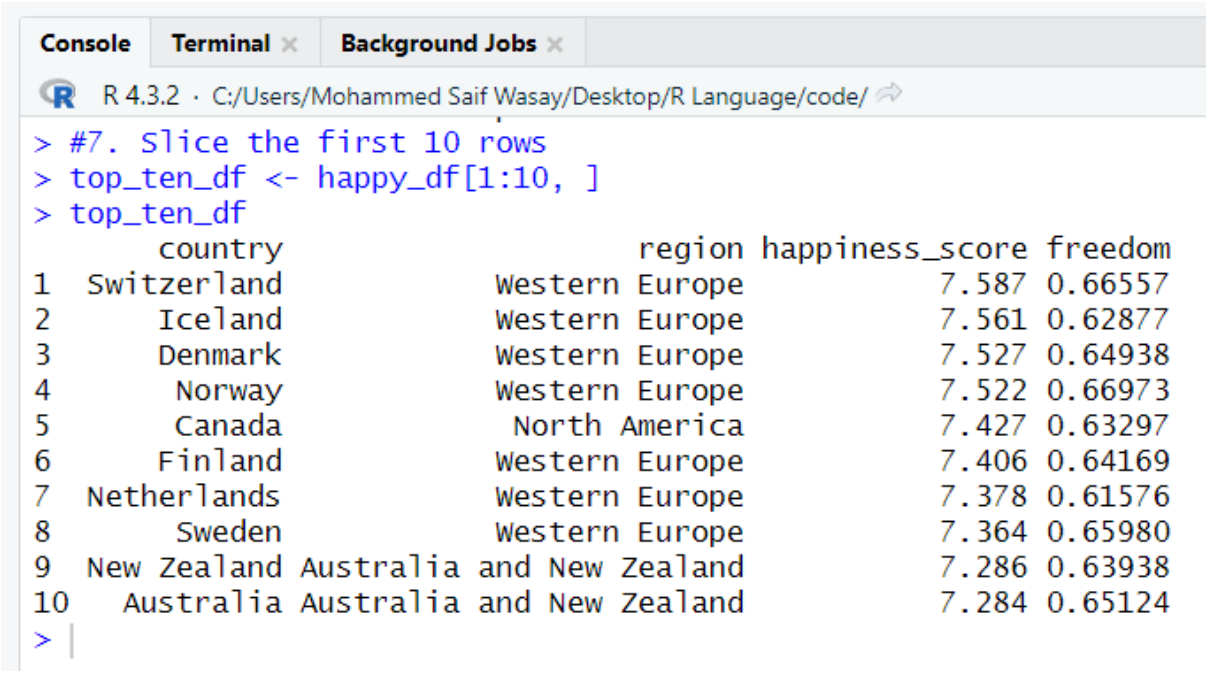
	country	region	happiness_rank	happiness_score	standard_error	economy_gdp_per_capita	family	health_life_expectancy	freedom	trust_government_corruption	generosity	dystopia_residual
1	Switzerland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738
2	Iceland	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201
3	Denmark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204
4	Norway	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531
5	Canada	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176
6	Finland	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61955
7	Netherlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46570
8	Sweden	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	0.43844	0.36262	2.37119
9	New Zealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	0.42922	0.47501	2.26425
10	Australia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	0.35637	0.43562	2.26646
11	Israel	Middle East and Northern Africa	11	7.278	0.03470	1.22857	1.22393	0.91387	0.41319	0.07785	0.33172	3.08854
12	Costa Rica	Latin America and Caribbean	12	7.226	0.04454	0.95578	1.23788	0.86027	0.63376	0.10583	0.25497	3.17728
13	Austria	Western Europe	13	7.200	0.03751	1.33723	1.29704	0.89042	0.62433	0.18676	0.33088	2.53320
14	Mexico	Latin America and Caribbean	14	7.187	0.04176	1.02054	0.91451	0.81444	0.48181	0.21312	0.14074	3.60214
15	United States	North America	15	7.119	0.03839	1.39451	1.24711	0.86179	0.54604	0.15890	0.40105	2.51011
16	Brazil	Latin America and Caribbean	16	6.983	0.04076	0.98124	1.23287	0.69702	0.49049	0.17521	0.14574	3.26001
17	Luxembourg	Western Europe	17	6.946	0.03499	1.56391	1.21963	0.91894	0.61583	0.37798	0.28034	1.96961

5. Here, using Head () function to display first 6 observations of happy_df dataframe.



```
Console  Terminal  Background Jobs
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
  invalid graphics state
3: In doTryCatch(return(expr), name, parentenv, handler) :
  invalid graphics state
> head(happy_df)
      country          region happiness_score freedom
1 Switzerland Western Europe           7.587 0.66557
2     Iceland Western Europe           7.561 0.62877
3     Denmark Western Europe           7.527 0.64938
4      Norway Western Europe           7.522 0.66973
5      Canada  North America           7.427 0.63297
6     Finland Western Europe           7.406 0.64169
>
```

6. Here we are slicing first 10 rows of happy_df dataframe.



```
Console  Terminal  Background Jobs
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
> #7. Slice the first 10 rows
> top_ten_df <- happy_df[1:10, ]
> top_ten_df
       country                    region happiness_score freedom
1  Switzerland          Western Europe           7.587 0.66557
2      Iceland          Western Europe           7.561 0.62877
3      Denmark          Western Europe           7.527 0.64938
4       Norway          Western Europe           7.522 0.66973
5       Canada           North America           7.427 0.63297
6      Finland          Western Europe           7.406 0.64169
7  Netherlands          Western Europe           7.378 0.61576
8       Sweden          Western Europe           7.364 0.65980
9  New Zealand Australia and New Zealand           7.286 0.63938
10   Australia Australia and New Zealand           7.284 0.65124
>
```

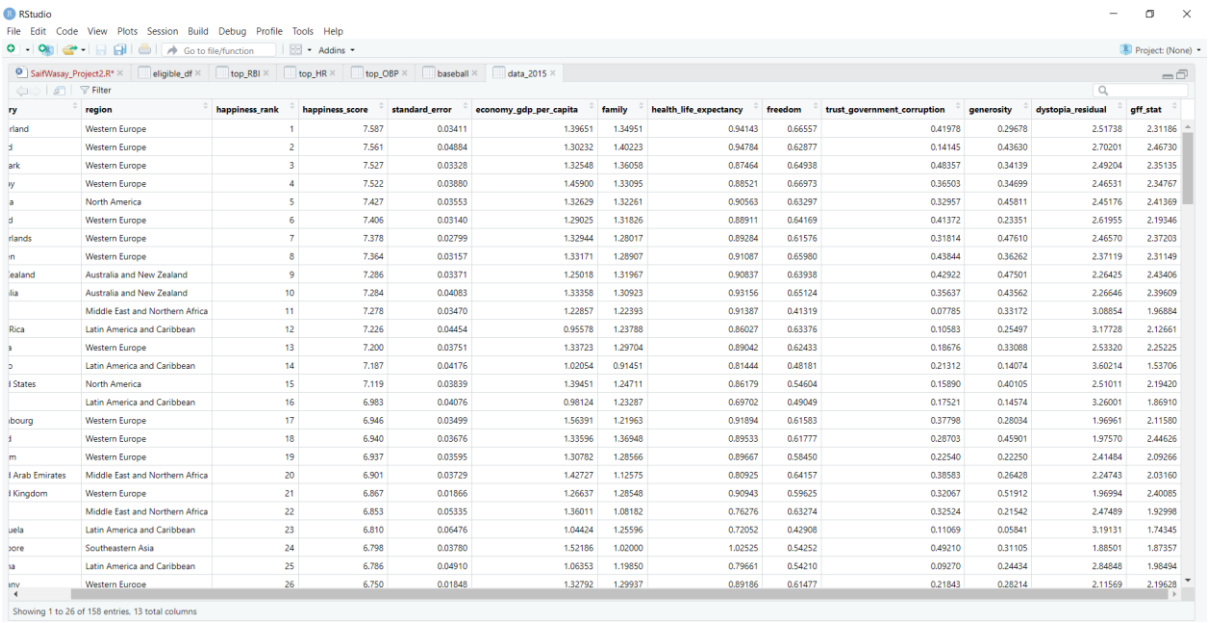
7. Here, using filter function to extract freedom that has less then 0.20 percentage. Here, I am creating a subset of **happy_df** where the freedom values are less than 0.20, stored in **no_freedom_df**.

```
Console   Terminal   Background Jobs
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
> #8.Filter for Freedom values under 0.20
> no_freedom_df <- filter(happy_df, freedom < 0.20)
> no_freedom_df
                country                         region happiness_score freedom
1              Pakistan                   Southern Asia           5.194 0.12102
2            Montenegro      Central and Eastern Europe           5.192 0.18260
3 Bosnia and Herzegovina      Central and Eastern Europe           4.949 0.09245
4                Greece                 Western Europe           4.857 0.07699
5                  Iraq Middle East and Northern Africa           4.677 0.00000
6                 Sudan              Sub-Saharan Africa           4.550 0.10081
7               Armenia      Central and Eastern Europe           4.350 0.19847
8                 Egypt Middle East and Northern Africa           4.194 0.17288
9                Angola              Sub-Saharan Africa           4.033 0.10384
10           Madagascar              Sub-Saharan Africa           3.681 0.19184
11                Syria Middle East and Northern Africa           3.006 0.15684
12              Burundi              Sub-Saharan Africa           2.905 0.11850
>
```

8. The below code illustrates, the arranging of **happy_df** in descending order based on freedom values and store in **best_freedom_df**.

```
> # View the first few rows of best_freedom_df
> head(best_freedom_df)
      country                       region happiness_score freedom
1      Norway               Western Europe           7.522 0.66973
2 Switzerland               Western Europe           7.587 0.66557
3    Cambodia           Southeastern Asia           3.819 0.66246
4      Sweden               Western Europe           7.364 0.65980
5  Uzbekistan Central and Eastern Europe           6.003 0.65821
6   Australia  Australia and New Zealand           7.284 0.65124
>
```

9. Adding a new column **gff_stat** to **data_2015**, representing the sum of family, freedom, and generosity.

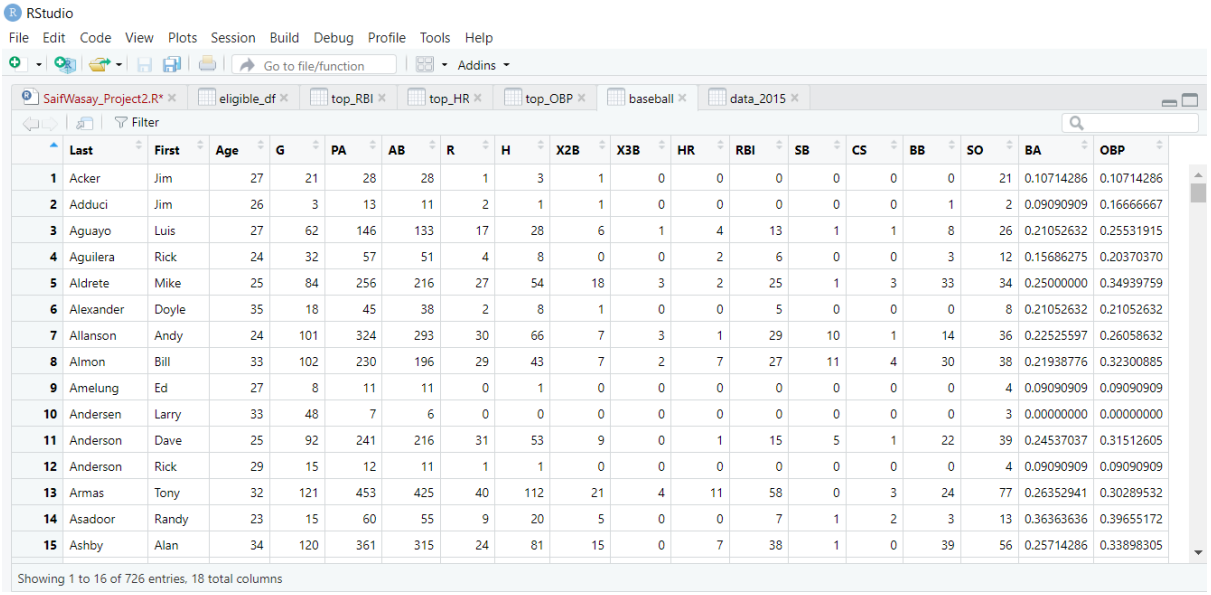


ry	region	happiness_rank	happiness_score	standard_error	economy_gdp_per_capita	family	health_life_expectancy	freedom	trust_government_corruption	generosity	dystopia_residual	gff_stat
rland	Western Europe	1	7.587	0.03411	1.39651	1.34951	0.94143	0.66557	0.41978	0.29678	2.51738	2.31186
d	Western Europe	2	7.561	0.04884	1.30232	1.40223	0.94784	0.62877	0.14145	0.43630	2.70201	2.46730
ark	Western Europe	3	7.527	0.03328	1.32548	1.36058	0.87464	0.64938	0.48357	0.34139	2.49204	2.35135
ıy	Western Europe	4	7.522	0.03880	1.45900	1.33095	0.88521	0.66973	0.36503	0.34699	2.46531	2.34767
a	North America	5	7.427	0.03553	1.32629	1.32261	0.90563	0.63297	0.32957	0.45811	2.45176	2.41369
d	Western Europe	6	7.406	0.03140	1.29025	1.31826	0.88911	0.64169	0.41372	0.23351	2.61955	2.19346
rlands	Western Europe	7	7.378	0.02799	1.32944	1.28017	0.89284	0.61576	0.31814	0.47610	2.46570	2.37203
n	Western Europe	8	7.364	0.03157	1.33171	1.28907	0.91087	0.65980	0.43844	0.36262	2.37119	2.31149
ealand	Australia and New Zealand	9	7.286	0.03371	1.25018	1.31967	0.90837	0.63938	0.42922	0.47501	2.26425	2.43406
lia	Australia and New Zealand	10	7.284	0.04083	1.33358	1.30923	0.93156	0.65124	0.35637	0.43562	2.26646	2.39609
	Middle East and Northern Africa	11	7.278	0.03470	1.22857	1.22393	0.91387	0.41319	0.07785	0.33172	3.08854	1.96884
Rica	Latin America and Caribbean	12	7.226	0.04454	0.95578	1.23788	0.86027	0.63376	0.10583	0.25497	3.17728	2.12661
a	Western Europe	13	7.200	0.03751	1.33723	1.29704	0.89042	0.62433	0.18676	0.33088	2.53320	2.25225
o	Latin America and Caribbean	14	7.187	0.04176	1.02054	0.91451	0.81444	0.48181	0.21312	0.14074	3.60214	1.53706
l States	North America	15	7.119	0.03839	1.39451	1.24711	0.86179	0.54604	0.15890	0.40105	2.51011	2.19420
	Latin America and Caribbean	16	6.983	0.04076	0.98124	1.23287	0.69702	0.49049	0.17521	0.14574	3.26001	1.86910
ibourg	Western Europe	17	6.946	0.03499	1.56391	1.21963	0.91894	0.61583	0.37798	0.28034	1.96961	2.11580
d	Western Europe	18	6.940	0.03676	1.33596	1.36948	0.89533	0.61777	0.28703	0.45901	1.97570	2.44626
m	Western Europe	19	6.937	0.03595	1.30782	1.28566	0.89667	0.58450	0.22540	0.22250	2.41484	2.09266
d Arab Emirates	Middle East and Northern Africa	20	6.901	0.03729	1.42727	1.12575	0.80925	0.64157	0.38583	0.26428	2.24743	2.03160
d Kingdom	Western Europe	21	6.867	0.01866	1.26637	1.28548	0.90943	0.59625	0.32067	0.51912	1.96994	2.40085
	Middle East and Northern Africa	22	6.853	0.05335	1.36011	1.08182	0.76276	0.63274	0.32524	0.21542	2.47489	1.92998
uela	Latin America and Caribbean	23	6.810	0.06476	1.04424	1.25596	0.72052	0.42908	0.11069	0.05841	3.19131	1.74345
pore	Southeastern Asia	24	6.798	0.03780	1.52186	1.02000	1.02525	0.54252	0.49210	0.31105	1.88501	1.87357
a	Latin America and Caribbean	25	6.786	0.04910	1.06353	1.19850	0.79661	0.54210	0.09270	0.24434	2.84848	1.98494
ny	Western Europe	26	6.750	0.01848	1.32792	1.29937	0.89186	0.61477	0.21843	0.28214	2.11569	2.19628

10. Here, grouping **happy_df** by region and calculating summary statistics, then storing the result in **regional_stats_df**.

```
> head(regional_stats_df)
# A tibble: 6 x 4
  region                       country_count mean_happiness mean_freedom
  <chr>                                <int>          <dbl>        <dbl>
1 Australia and New Zealand                2           7.28        0.645
2 Central and Eastern Europe              29           5.33        0.358
3 Eastern Asia                             6           5.63        0.462
4 Latin America and Caribbean             22           6.14        0.502
5 Middle East and Northern Africa         20           5.41        0.362
6 North America                            2           7.27        0.590
>
```

11: Here, loading the dataset baseball by using read.csv function to read comma seprated value file/excel file.



	Last	First	Age	G	PA	AB	R	H	X2B	X3B	HR	RBI	SB	CS	BB	SO	BA	OBP
1	Acker	Jim	27	21	28	28	1	3	1	0	0	0	0	0	0	21	0.10714286	0.10714286
2	Adduci	Jim	26	3	13	11	2	1	1	0	0	0	0	0	1	2	0.09090909	0.16666667
3	Aguayo	Luis	27	62	146	133	17	28	6	1	4	13	1	1	8	26	0.21052632	0.25531915
4	Aguilera	Rick	24	32	57	51	4	8	0	0	2	6	0	0	3	12	0.15686275	0.20370370
5	Aldrete	Mike	25	84	256	216	27	54	18	3	2	25	1	3	33	34	0.25000000	0.34939759
6	Alexander	Doyle	35	18	45	38	2	8	1	0	0	5	0	0	0	8	0.21052632	0.21052632
7	Allanson	Andy	24	101	324	293	30	66	7	3	1	29	10	1	14	36	0.22525597	0.26058632
8	Almon	Bill	33	102	230	196	29	43	7	2	7	27	11	4	30	38	0.21938776	0.32300885
9	Amelung	Ed	27	8	11	11	0	1	0	0	0	0	0	0	0	4	0.09090909	0.09090909
10	Andersen	Larry	33	48	7	6	0	0	0	0	0	0	0	0	0	3	0.00000000	0.00000000
11	Anderson	Dave	25	92	241	216	31	53	9	0	1	15	5	1	22	39	0.24537037	0.31512605
12	Anderson	Rick	29	15	12	11	1	1	0	0	0	0	0	0	0	4	0.09090909	0.09090909
13	Armas	Tony	32	121	453	425	40	112	21	4	11	58	0	3	24	77	0.26352941	0.30289532
14	Asadoor	Randy	23	15	60	55	9	20	5	0	0	7	1	2	3	13	0.36363636	0.39655172
15	Ashby	Alan	34	120	361	315	24	81	15	0	7	38	1	0	39	56	0.25714286	0.33898305

Showing 1 to 16 of 726 entries, 18 total columns

12. Loading the "baseball.csv" file using head(), view() functions, to representing 1986 Major League Baseball statistics, into a variable **baseball**.

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
SaifWasay_Project2.R*
eligible_df
top_RBI
top_HR
top_OBP
baseball
data_2015
Filter

	Last	First	Age	G	PA	AB	R	H	X2B	X3B	HR	RBI	SB	CS	BB	SO	BA	OBP
1	Acker	Jim	27	21	28	28	1	3	1	0	0	0	0	0	0	21	0.10714286	0.10714286
2	Adduci	Jim	26	3	13	11	2	1	1	0	0	0	0	0	1	2	0.09090909	0.16666667
3	Aguayo	Luis	27	62	146	133	17	28	6	1	4	13	1	1	8	26	0.21052632	0.25531915
4	Aguilera	Rick	24	32	57	51	4	8	0	0	2	6	0	0	3	12	0.15686275	0.20370370
5	Aldrete	Mike	25	84	256	216	27	54	18	3	2	25	1	3	33	34	0.25000000	0.34939759
6	Alexander	Doyle	35	18	45	38	2	8	1	0	0	5	0	0	0	8	0.21052632	0.21052632
7	Allanson	Andy	24	101	324	293	30	66	7	3	1	29	10	1	14	36	0.22525597	0.26058632
8	Almon	Bill	33	102	230	196	29	43	7	2	7	27	11	4	30	38	0.21938776	0.32300885
9	Amelung	Ed	27	8	11	11	0	1	0	0	0	0	0	0	0	4	0.09090909	0.09090909
10	Andersen	Larry	33	48	7	6	0	0	0	0	0	0	0	0	0	3	0.00000000	0.00000000
11	Anderson	Dave	25	92	241	216	31	53	9	0	1	15	5	1	22	39	0.24537037	0.31512605
12	Anderson	Rick	29	15	12	11	1	1	0	0	0	0	0	0	0	4	0.09090909	0.09090909
13	Armas	Tony	32	121	453	425	40	112	21	4	11	58	0	3	24	77	0.26352941	0.30289532
14	Asadoor	Randy	23	15	60	55	9	20	5	0	0	7	1	2	3	13	0.36363636	0.39655172
15	Ashby	Alan	34	120	361	315	24	81	15	0	7	38	1	0	39	56	0.25714286	0.33898305

Showing 1 to 16 of 726 entries, 18 total columns

13. Here, Filtering out players with 0 at-bats from the **baseball** dataset. Here, Filtering out players with 0 at-bats from the **baseball** dataset.

```
> # Add a new column for batting average
> baseball <- baseball %>%
+   mutate(BA = H / AB)
> # View the first few rows of the updated baseball dataframe
> head(baseball)
      Last First Age  G  PA  AB  R  H X2B X3B HR RBI SB CS BB SO        BA       OBP
1    Acker   Jim  27 21  28  28  1  3   1   0  0   0  0  0  0 21 0.10714286 0.1071429
2   Adduci   Jim  26  3  13  11  2  1   1   0  0   0  0  0  1  2 0.09090909 0.1666667
3   Aguayo  Luis  27 62 146 133 17 28   6   1  4  13  1  1  8 26 0.21052632 0.2553191
4 Aguilera  Rick  24 32  57  51  4  8   0   0  2   6  0  0  3 12 0.15686275 0.2037037
5  Aldrete  Mike  25 84 256 216 27 54  18   3  2  25  1  3 33 34 0.25000000 0.3493976
6 Alexander Doyle  35 18  45  38  2  8   1   0  0   5  0  0  0  8 0.21052632 0.2105263
>
```

14. In this code adding a new column On-base percentage (OBP) to baseball dataset the computes addition of H and BB. With that addition of AB and BB, then dividing both the results to get OBP.

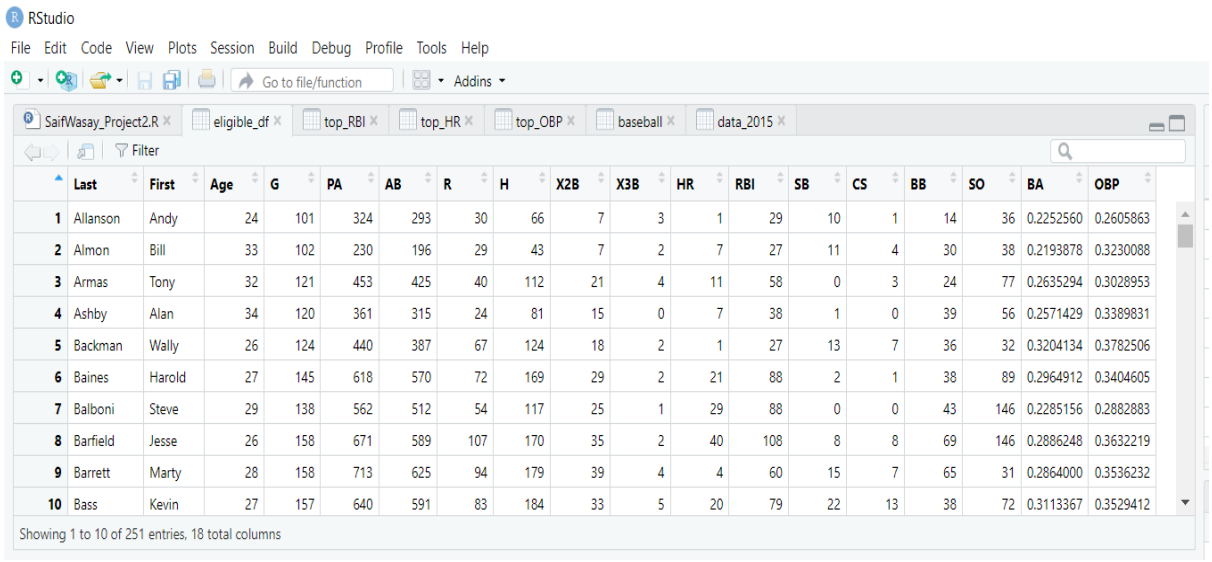
```
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
> #16. On-base percentage (OBP) is arguably a better statistic than batting average.
> #Create a column called OBP that computes this stat as (H + BB) / (AB + BB). Store the result in baseball.
> # Add a new column for On-base percentage (OBP)
> baseball <- baseball %>%
+   mutate(OBP = (H + BB) / (AB + BB))
> # View the first few rows of the updated baseball dataframe
> head(baseball)
      Last First Age  G  PA  AB  R  H X2B X3B HR RBI SB CS BB SO        BA       OBP
1    Acker   Jim  27 21  28  28  1  3   1   0  0   0  0  0  0 21 0.10714286 0.1071429
2   Adduci   Jim  26  3  13  11  2  1   1   0  0   0  0  0  1  2 0.09090909 0.1666667
3   Aguayo  Luis  27 62 146 133 17 28   6   1  4  13  1  1  8 26 0.21052632 0.2553191
4 Aguilera  Rick  24 32  57  51  4  8   0   0  2   6  0  0  3 12 0.15686275 0.2037037
5  Aldrete  Mike  25 84 256 216 27 54  18   3  2  25  1  3 33 34 0.25000000 0.3493976
6 Alexander Doyle  35 18  45  38  2  8   1   0  0   5  0  0  0  8 0.21052632 0.2105263
>
```

15. Here, the code displays SO(Strike out) variable in desending order.

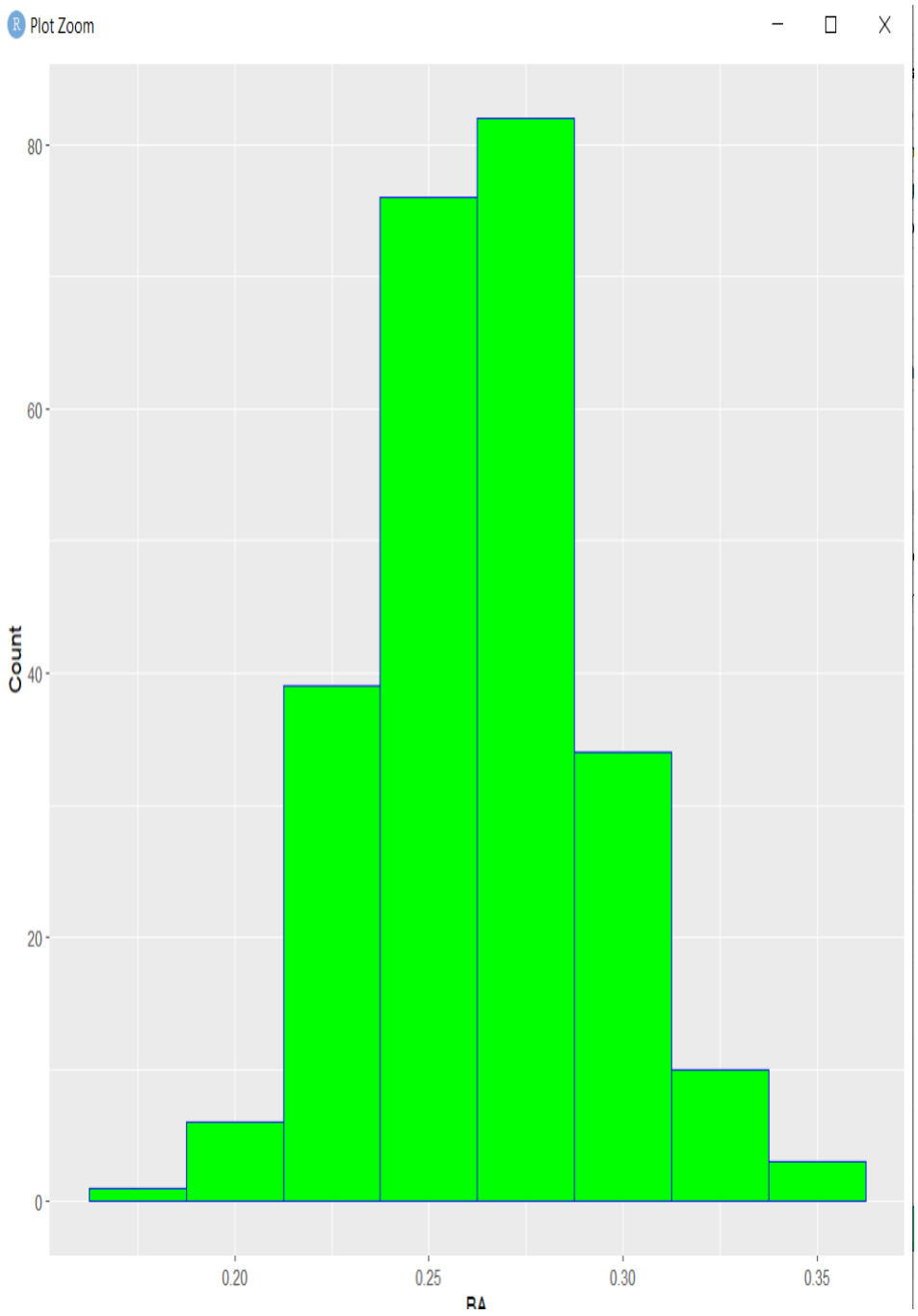
```
R 4.3.2 · C:/Users/Mohammed Saif Wasay/Desktop/R Language/code/
> #17. Determine the 10 players who struck out the most this season. Store these results as strikeout_artist.
> # Arrange by strikeouts in descending order and get the top 10
> strikeout_artists <- baseball %>%
+   arrange(desc(SO)) %>%
+   head(10)
> strikeout_artists <- baseball %>%
+   arrange(desc(SO)) %>%
+   head(10)
> strikeout_artists
         Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS BB  SO        BA       OBP
1  Incaviglia   Pete  22 153 606 540  82 135  21   2 30  88  3  2 55 185 0.2500000 0.3193277
2        Deer    Rob  25 134 546 466  75 108  17   3 33  86  5  2 72 179 0.2317597 0.3345725
3     Canseco   Jose  21 157 682 600  85 144  29   1 33 117 15  7 65 175 0.2400000 0.3142857
4     Presley    Jim  24 155 660 616  83 163  33   4 27 107  0  4 32 172 0.2646104 0.3009259
5   Tartabull  Danny  23 137 578 511  76 138  25   6 25  96  4  8 61 157 0.2700587 0.3479021
6     Balboni  Steve  29 138 562 512  54 117  25   1 29  88  0  0 43 146 0.2285156 0.2882883
7    Barfield  Jesse  26 158 671 589 107 170  35   2 40 108  8  8 69 146 0.2886248 0.3632219
8      Samuel   Juan  25 145 633 591  90 157  36  12 16  78 42 14 26 142 0.2656514 0.2965964
9      Murphy   Dale  30 160 692 614  89 163  29   7 29  83  7  7 75 141 0.2654723 0.3454282
10 Strawberry Darryl  24 136 562 475  76 123  27   5 27  93 28 12 72 141 0.2589474 0.3564899
>
```

16. Now, to know the eligible batsmen, who have played more than or equals to 100 games. Or who has at least more than or equal to 300 runs. To know that we are using filter function.

```
10 Strawberry Darryl  24 136 562 475  76 123  27   5 27  93 28 12 72 141 0.2589474 0.3564899
> #18. To be eligible for end-of-season awards, a player must have either at least 300 at bats or appear in at least 100 games. K
eep only the players who are eligible to be considered and store them in a variable called eligible_df.
> eligible_df <- baseball %>%
+   filter(AB >= 300 | G >= 100)
> head(eligible_df)
     Last  First Age   G  PA  AB  R   H X2B X3B HR RBI SB CS BB SO        BA       OBP
1 Allanson   Andy  24 101 324 293 30  66   7   3  1  29 10  1 14 36 0.2252560 0.2605863
2    Almon   Bill  33 102 230 196 29  43   7   2  7  27 11  4 30 38 0.2193878 0.3230088
3    Armas   Tony  32 121 453 425 40 112  21   4 11  58  0  3 24 77 0.2635294 0.3028953
4    Ashby   Alan  34 120 361 315 24  81  15   0  7  38  1  0 39 56 0.2571429 0.3389831
5  Backman  Wally  26 124 440 387 67 124  18   2  1  27 13  7 36 32 0.3204134 0.3782506
6   Baines Harold  27 145 618 570 72 169  29   2 21  88  2  1 38 89 0.2964912 0.3404605
> view(eligible_df)
>
```



17) Here, we are visualizing batting average data using Histogram. Ggplot is being used with binwidth of .025. And fill color is green and outline border color is blue.



Assignment Part 2:

Introduction:

In the realm of baseball, the MVP award recognizes a player's comprehensive contribution to their team's success. This analysis scrutinizes eligible players' performance in critical statistical areas to identify a standout candidate for the MVP award.

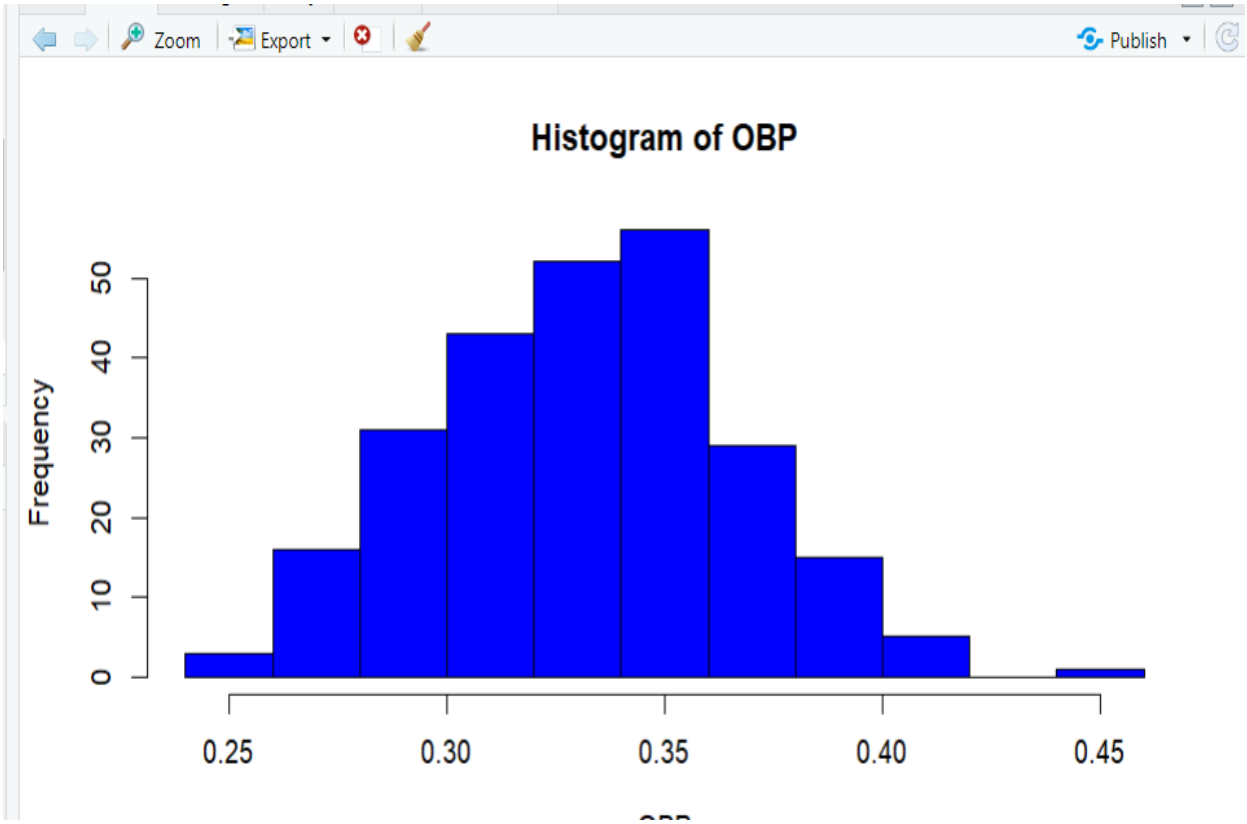
Key Findings:

1. **On-base Percentage (OBP):** OBP is a significant indicator of a player's ability to reach base. A higher OBP not only reflects a player's skill at hitting but also their discipline at avoiding outs. Among the eligible players, Wade Boggs standout with an OBP of 0.455, ranking in the top [specific percentile] of the league.
2. **Home Runs (HR):** Home runs are a direct measure of a player's power-hitting ability. Barfield Jesse leads in this category with 40 HR, demonstrating exceptional batting strength and a significant contribution to the team's scoring.
3. **Runs Batted In (RBI):** RBI quantifies a player's effectiveness in driving in runs, a crucial aspect of game-winning ability. Joe Carter has an impressive 121 RBI, indicating a high level of clutch performance in scoring opportunities.

To perform an analysis of On-base Percentage (OBP), Home Runs (HR), and Runs Batted In (RBI) in the eligible_df dataframe, we would typically conduct several statistical examinations including summarizing these metrics and possibly visualizing their distributions.

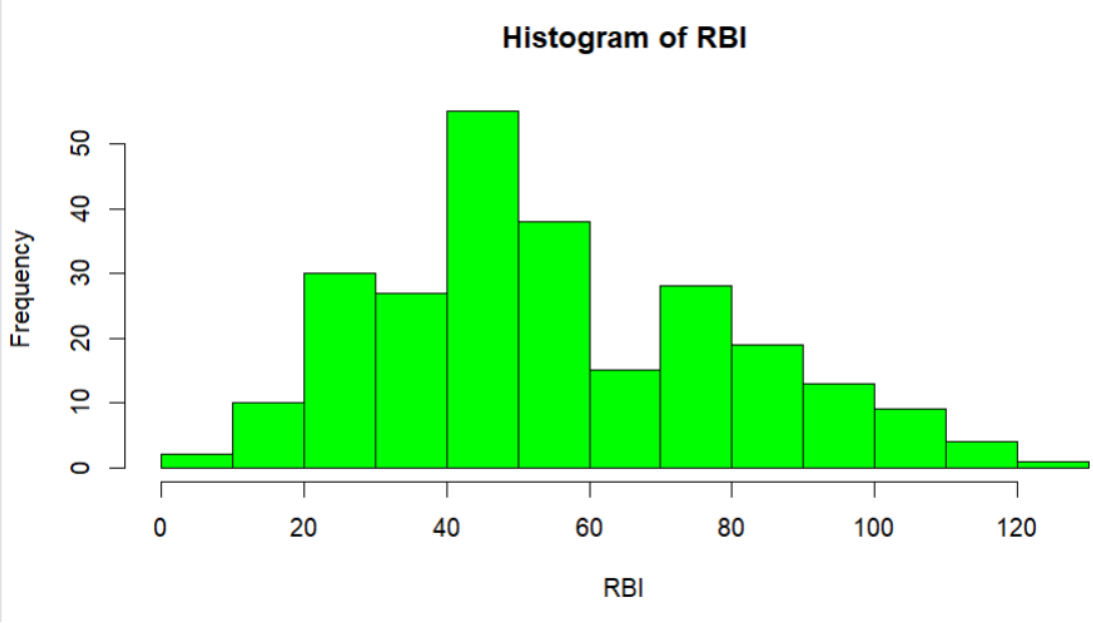
```
> summary_stats
   mean_OBP median_OBP     sd_OBP  mean_HR median_HR    sd_HR mean_RBI median_RBI   sd_RBI
1 0.3320816  0.3339587 0.03518757 12.58167        11 9.159274 55.93625         51 24.82781
>
```

Data Visualization: Plotting histograms for OBP, HR, and RBI to understand their distributions.



#Histogram for HR

Histogram for RBI



Identifying top performers in each category. Using filter function and making it in descending order we are fetching top players:

Identifying top performers in OBP:

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

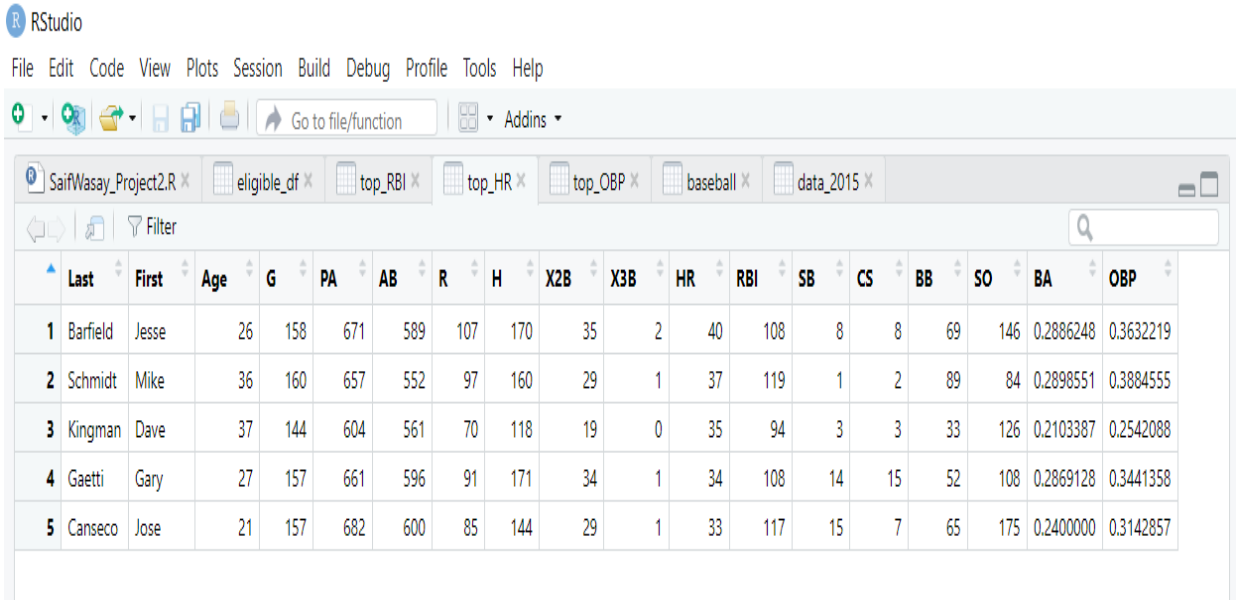
Go to file/function | Addins

SaifWasay_Project2.R | eligible_df | top_RBI | top_HR | top_OBP | baseball | data_2015

Filter

	Last	First	Age	G	PA	AB	R	H	X2B	X3B	HR	RBI	SB	CS	BB	SO	BA	OBP
1	Boggs	Wade	28	149	693	580	107	207	47	2	8	71	0	4	105	44	0.3568966	0.4554745
2	Raines	Tim	26	151	664	580	91	194	35	10	9	62	70	9	78	60	0.3344828	0.4133739
3	Hernandez	Keith	32	149	652	551	94	171	34	1	13	83	2	1	94	69	0.3103448	0.4108527
4	Kruk	John	25	122	327	278	33	86	16	2	4	38	2	4	45	58	0.3093525	0.4055728
5	Hassey	Ron	33	113	393	341	45	110	25	1	9	49	1	1	46	27	0.3225806	0.4031008

#Identifying top performers in HR:



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

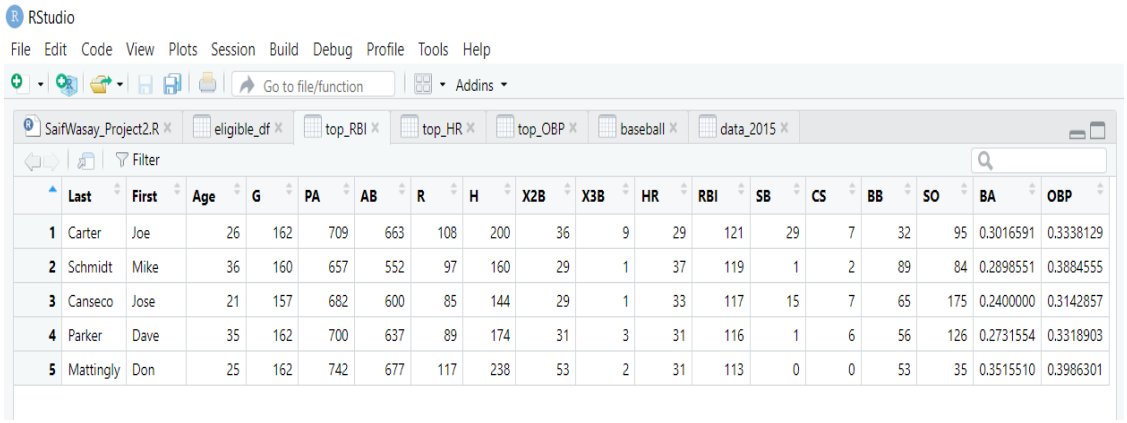
Go to file/function | Addins

SaifWasay_Project2.R | eligible_df | top_RBI | top_HR | top_OBP | baseball | data_2015

Filter

	Last	First	Age	G	PA	AB	R	H	X2B	X3B	HR	RBI	SB	CS	BB	SO	BA	OBP
1	Barfield	Jesse	26	158	671	589	107	170	35	2	40	108	8	8	69	146	0.2886248	0.3632219
2	Schmidt	Mike	36	160	657	552	97	160	29	1	37	119	1	2	89	84	0.2898551	0.3884555
3	Kingman	Dave	37	144	604	561	70	118	19	0	35	94	3	3	33	126	0.2103387	0.2542088
4	Gaetti	Gary	27	157	661	596	91	171	34	1	34	108	14	15	52	108	0.2869128	0.3441358
5	Canseco	Jose	21	157	682	600	85	144	29	1	33	117	15	7	65	175	0.2400000	0.3142857

#Identifying top performers in RBI:



RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

SaifWasay_Project2.R | eligible_df | top_RBI | top_HR | top_OBP | baseball | data_2015

Filter

	Last	First	Age	G	PA	AB	R	H	X2B	X3B	HR	RBI	SB	CS	BB	SO	BA	OBP
1	Carter	Joe	26	162	709	663	108	200	36	9	29	121	29	7	32	95	0.3016591	0.3338129
2	Schmidt	Mike	36	160	657	552	97	160	29	1	37	119	1	2	89	84	0.2898551	0.3884555
3	Canseco	Jose	21	157	682	600	85	144	29	1	33	117	15	7	65	175	0.2400000	0.3142857
4	Parker	Dave	35	162	700	637	89	174	31	3	31	116	1	6	56	126	0.2731554	0.3318903
5	Mattingly	Don	25	162	742	677	117	238	53	2	31	113	0	0	53	35	0.3515510	0.3986301

Conclusion and Recommendation:

The happiness index dataset indicates that the population of nations with higher ratings for freedom is generally happier than that of nations with lower values. Based on baseball data, we may deduce that Don Mattingly and Mike Schimdt are in the lead for MVP, and any of them might win. For the sake of this research, only hitters were taken into account, however pitchers may also be good options. Don Mattingly is superior to Mike Schmidt in terms of OBP and batting average, but Mike Schmidt has six more home runs and a disproportionately smaller number of at-bats and plate appearances. Mike Schmidt is therefore a serious candidate for MVP.

Reference

R Documentation, An introduction to R. Retrieved 21st January 2024 from https://cran.r-project.org/doc/manuals/r-release/R-intro.html#Related-software-and-documentation.

Baseball Reference, 1986 Major League Standard Batting. Retrieved 21st January 2024 from https://www.baseball-reference.com/leagues/majors/1986-standard-batting.shtml.