Yuvi

June 14, 2019

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        %matplotlib inline
        import warnings
        warnings.filterwarnings('ignore')
```

Yuvraj Singh - http://www.espncricinfo.com/india/content/player/36084.html

```
In [2]: df = pd.read_csv("yuv.csv")
        df['Date']= pd.to_datetime(df.Date)
```

```
In [3]: df.head()
```

```
Out[3]:   Bat1 Bat2 Runs  BF      SR  4s 6s          Opposition         Ground  \
        0  DNB    -    -   -       -   -  -         ODI v Kenya  Nairobi (Gym)
        1   84    -   84  80  105.00  12  0     ODI v Australia  Nairobi (Gym)
        2   41    -   41  35  117.14   6  0  ODI v South Africa  Nairobi (Gym)
        3   18    -   18  19   94.73   2  0   ODI v New Zealand  Nairobi (Gym)
        4    7    -    7  15   46.66   0  0     ODI v Sri Lanka        Sharjah

                Date       Game#
        0 2000-10-03  ODI # 1630
        1 2000-10-07  ODI # 1633
        2 2000-10-13  ODI # 1638
        3 2000-10-15  ODI # 1639
        4 2000-10-20  ODI # 1640
```

```
In [4]: new = df["Opposition"].str.split("v", n = 1, expand = True)
        new.head()
```

```
Out[4]:       0               1
        0  ODI           Kenya
        1  ODI       Australia
        2  ODI    South Africa
        3  ODI     New Zealand
        4  ODI       Sri Lanka
```

```
In [5]: df["Match_Type"]= new[0].str.strip()
        df.drop(columns =["Opposition"], inplace = True)
        df["Opposition"]= new[1].str.strip()
        df.head()
```

```
Out[5]:   Bat1 Bat2 Runs  BF      SR  4s  6s         Ground        Date       Game#  \
        0  DNB    -    -   -       -   -   -  Nairobi (Gym) 2000-10-03  ODI # 1630
        1   84    -   84  80  105.00  12   0  Nairobi (Gym) 2000-10-07  ODI # 1633
        2   41    -   41  35  117.14   6   0  Nairobi (Gym) 2000-10-13  ODI # 1638
        3   18    -   18  19   94.73   2   0  Nairobi (Gym) 2000-10-15  ODI # 1639
        4    7    -    7  15   46.66   0   0        Sharjah 2000-10-20  ODI # 1640

          Match_Type    Opposition
        0        ODI         Kenya
        1        ODI     Australia
        2        ODI  South Africa
        3        ODI   New Zealand
        4        ODI     Sri Lanka
```

```
In [6]: ODI_data = df[df['Match_Type']=='ODI']
        TEST_data = df[df['Match_Type']=='Test']
        T20_data = df[df['Match_Type']=='T20I']
```

1 ODI Batting DATA Analysis

```
In [7]: no_of_ducks = ODI_data[(ODI_data['Runs'] =='0') & (ODI_data['Bat1'] !='0*')]
```

```
In [8]: no_of_ducks.head()
```

```
Out[8]:    Bat1 Bat2 Runs BF    SR 4s 6s          Ground        Date       Game#  \
        24    0    -    0  1  0.00  0  0  Port Elizabeth 2001-10-17  ODI # 1761
        27    0    -    0  2  0.00  0  0          Durban 2001-10-26  ODI # 1766
        53    0    -    0  2  0.00  0  0          Napier 2002-12-29  ODI # 1927
        60    0    -    0  8  0.00  0  0       Centurion 2003-02-15  ODI # 1951
        77    0    -    0  3  0.00  0  0         Cuttack 2003-11-06  ODI # 2056

           Match_Type    Opposition
        24        ODI         Kenya
        27        ODI  South Africa
        53        ODI   New Zealand
        60        ODI     Australia
        77        ODI   New Zealand
```

```
In [9]: total_duck = len(no_of_ducks)
```

```
In [10]: total_duck
```

```
Out[10]: 18
```

#1. Total duck - 18

```
In [11]: no_of_ducks['Opposition'].value_counts().reset_index(name='counts').head(5)
```

```
Out[11]:          index  counts
         0     Australia       4
         1   New Zealand       3
         2     Sri Lanka       3
         3  South Africa       2
         4       England       2
```

```
In [12]: no_of_ducks['Ground'].value_counts().reset_index(name='counts').head(5)
```

```
Out[12]:           index  counts
         0         Mohali       2
         1   Kuala Lumpur       2
         2      Centurion       1
         3        Karachi       1
         4  Colombo (RPS)       1
```

```
In [13]: zero_at_home_ground= ["Mohali", "Nagpur", "Cuttack", "Dharamsala"]

         info_duck = dict(no_of_ducks['Ground'].value_counts())
         home_count =0
         for h in zero_at_home_ground:
             home_count += info_duck[h]

         away = len(no_of_ducks) - home_count

         print "Home -{} , per - {} %".format(str(home_count), str(float(home_count)/len(ODI_da
         print "Away -{} , per - {} %".format(str(away), str(float(away)/len(ODI_data)*100))
```

```
Home -5 , per - 1.64473684211 %
Away -13 , per - 4.27631578947 %
```

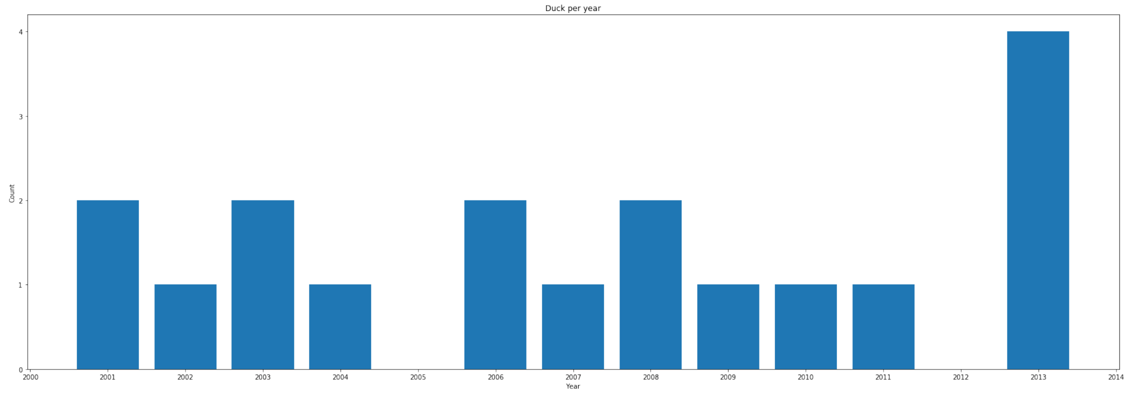
```
In [14]: no_of_ducks["BF"] = no_of_ducks["BF"].apply(pd.to_numeric)
```

```
In [15]: print len(no_of_ducks[no_of_ducks["BF"] ==1])
         print no_of_ducks["BF"].min(), no_of_ducks["BF"].max()
```

```
3
1 8
```

```
In [16]: no_of_ducks.index = no_of_ducks['Date']
```

```
In [17]: duck_yr = dict(no_of_ducks['Date'].dt.year.value_counts())
         plt.rcParams['figure.figsize'] = [30, 10]
         plt.xticks(range(2000,2020))
         plt.yticks(range(0,8))
         plt.bar(*zip(*sorted(duck_yr.items())))
         plt.title("Duck per year")
         plt.xlabel("Year")
         plt.ylabel("Count")
         plt.show()
```



#2.

. Most duck against Australia, max duck he made at Mohali,India. It's his home ground as well

- Home -5 , per - 1.64473684211 Away -13 , per - 4.27631578947 %
  - First ball Duck 3 times, all cases he got out within first 8 ball he faced, 2013 he made maxmimum no of ducks in a year.

```
In [18]: all_odi_score = ODI_data[(ODI_data['Bat1'] != 'DNB') & (ODI_data['Bat1'] != 'TDNB')]
```

```
In [19]: all_odi_score["BF"] = all_odi_score["BF"].apply(pd.to_numeric)
         all_odi_score["Runs"] = all_odi_score["Runs"].apply(pd.to_numeric)
         all_odi_score["4s"] = all_odi_score["4s"].apply(pd.to_numeric)
         all_odi_score["6s"] = all_odi_score["6s"].apply(pd.to_numeric)
```

```
In [20]: all_odi_score['Runs'].max()
```

```
Out[20]: 150
```

```
In [21]: all_odi_score[all_odi_score['Runs'] ==150]
```

```
Out[21]:     Bat1 Bat2  Runs   BF      SR  4s  6s   Ground        Date       Game#  \
         389  150    -   150  127  118.11  21   3  Cuttack 2017-01-19  ODI # 3821

             Match_Type Opposition
         389        ODI    England
```

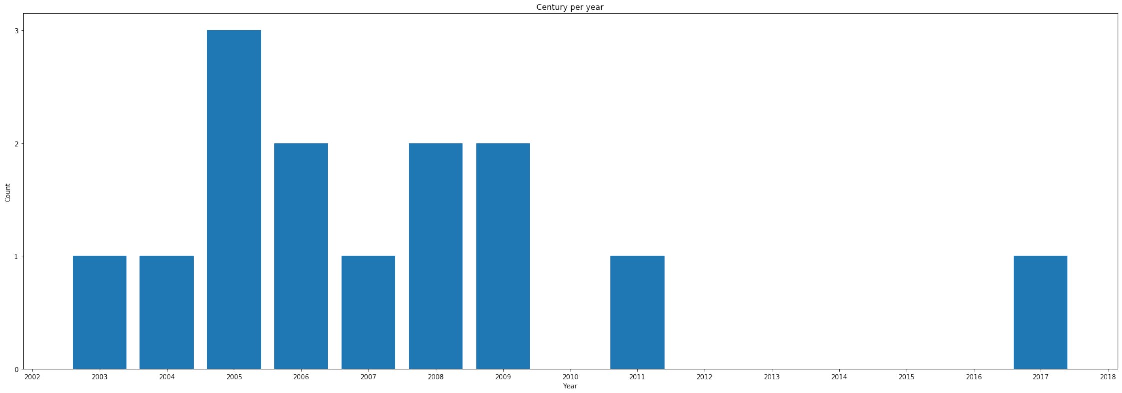
```
In [22]: odi_cen = all_odi_score[all_odi_score['Runs'] >=100]
In [23]: odi_cen['Opposition'].value_counts().reset_index(name='counts').head(5)
```

```
Out[23]:            index  counts
         0       England       4
         1    West Indies      3
         2      Australia      2
         3   South Africa      1
         4      Sri Lanka      1
```

```
In [24]: odi_cen['Ground'].value_counts().reset_index(name='counts').head(5)
```

```
Out[24]:                  index  counts
         0        Colombo (RPS)       2
         1   Hyderabad (Deccan)       2
         2               Cuttack      1
         3                Indore      1
         4              Kingston      1
```

```
In [25]: cen_yr = dict(odi_cen['Date'].dt.year.value_counts())
         plt.xticks(range(2000,2020))
         plt.yticks(range(0,5))
         plt.bar(*zip(*sorted(cen_yr.items())))
         plt.title("Century per year")
         plt.xlabel("Year")
         plt.ylabel("Count")
         plt.show()
```



```
In [26]: odi_cen
```

```
Out[26]:      Bat1 Bat2  Runs   BF      SR  4s  6s              Ground        Date  \
         70   102*    -   102   85  120.00   9   4               Dhaka 2003-04-11
         85    139    -   139  122  113.93  16   2              Sydney 2004-01-22
         125   110    -   110  114   96.49  11   1       Colombo (RPS) 2005-08-07
         130   120    -   120  124   96.77  12   1              Harare 2005-09-04
```

```
141   103    -   103  122   84.42  10   3  Hyderabad (Deccan) 2005-11-16
154  107*    -   107   93  115.05  14   0             Karachi 2006-02-19
159   103    -   103   76  135.52  10   3              Margao 2006-04-03
210   121    -   121  115  105.21  12   3  Hyderabad (Deccan) 2007-10-05
247  138*    -   138   78  176.92  16   6             Rajkot 2008-11-14
248   118    -   118  122   96.72  15   2              Indore 2008-11-17
256   117    -   117   95  123.15  17   1        Colombo (RPS) 2009-02-03
275   131    -   131  102  128.43  10   7            Kingston 2009-06-26
327   113    -   113  123   91.86  10   2             Chennai 2011-03-20
389   150    -   150  127  118.11  21   3             Cuttack 2017-01-19

          Game# Match_Type    Opposition
70   ODI # 2001        ODI    Bangladesh
85   ODI # 2086        ODI     Australia
125  ODI # 2267        ODI   West Indies
130  ODI # 2280        ODI      Zimbabwe
141  ODI # 2297        ODI  South Africa
154  ODI # 2333        ODI      Pakistan
159  ODI # 2359        ODI       England
210  ODI # 2625        ODI     Australia
247  ODI # 2774        ODI       England
248  ODI # 2777        ODI       England
256  ODI # 2813        ODI     Sri Lanka
275  ODI # 2852        ODI   West Indies
327  ODI # 3141        ODI   West Indies
389  ODI # 3821        ODI       England
```

```
In [27]: len(odi_cen)
```

```
Out[27]: 14
```

#3.

- Most 100+ score against England.
- Max Score 150 that too against Engaland and last Century in ODI cricket. Colombo (RPS) and Hydrabad he scored 2 century each.

```
In [28]: odi_half_cen = all_odi_score[(all_odi_score['Runs'] >=50) & (all_odi_score['Runs'] <=9
```

```
In [29]: len(odi_half_cen)
```

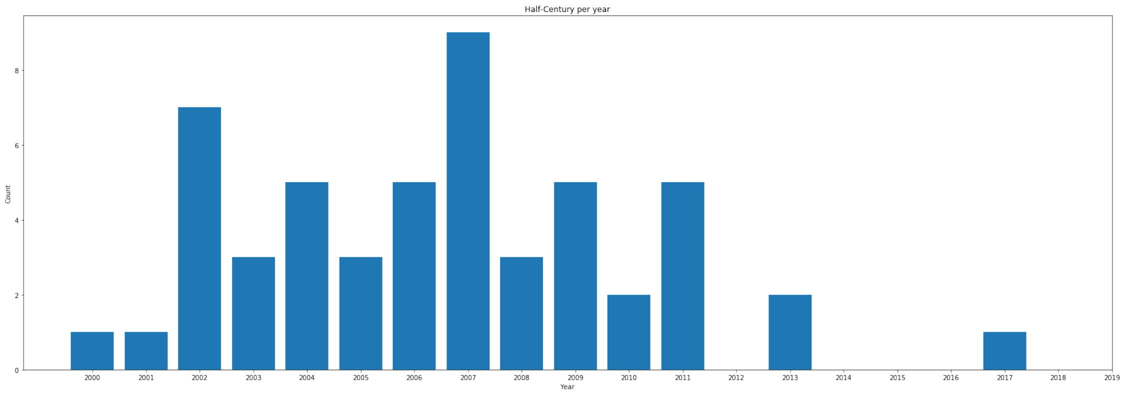
```
Out[29]: 52
```

```
In [30]: odi_half_cen.head()
```

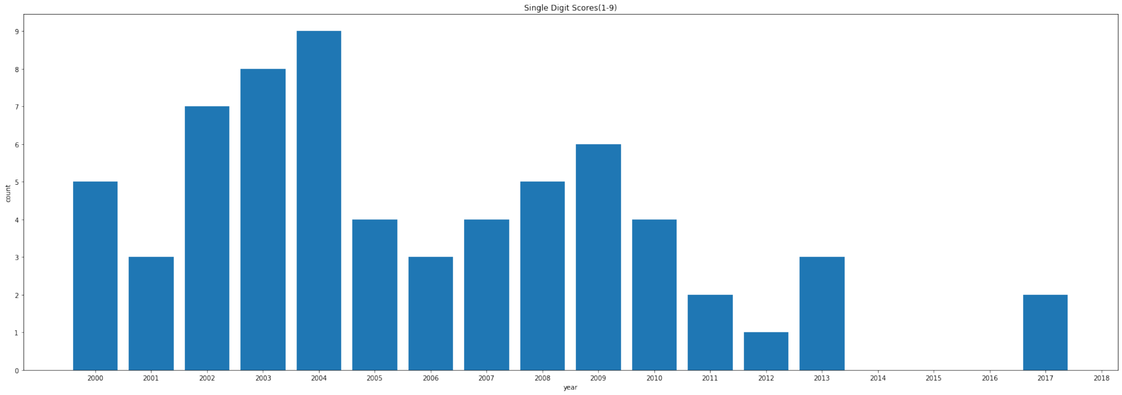
```
Out[30]:    Bat1 Bat2  Runs   BF      SR  4s  6s           Ground       Date  \
         1    84    -    84   80  105.00  12   0    Nairobi (Gym) 2000-10-07
         18  98*    -    98  110   89.09   6   1    Colombo (SSC) 2001-08-01
```

```
28  80*    -    80   60  133.33   8   1  Hyderabad (Deccan) 2002-03-16
29   75    -    75   52  144.23   6   3            Guwahati 2002-03-19
33  64*    -    64   65   98.46   7   0              Lord's 2002-06-29

         Game# Match_Type Opposition
1   ODI # 1633        ODI  Australia
18  ODI # 1742        ODI  Sri Lanka
28  ODI # 1817        ODI   Zimbabwe
29  ODI # 1818        ODI   Zimbabwe
33  ODI # 1848        ODI    England
```

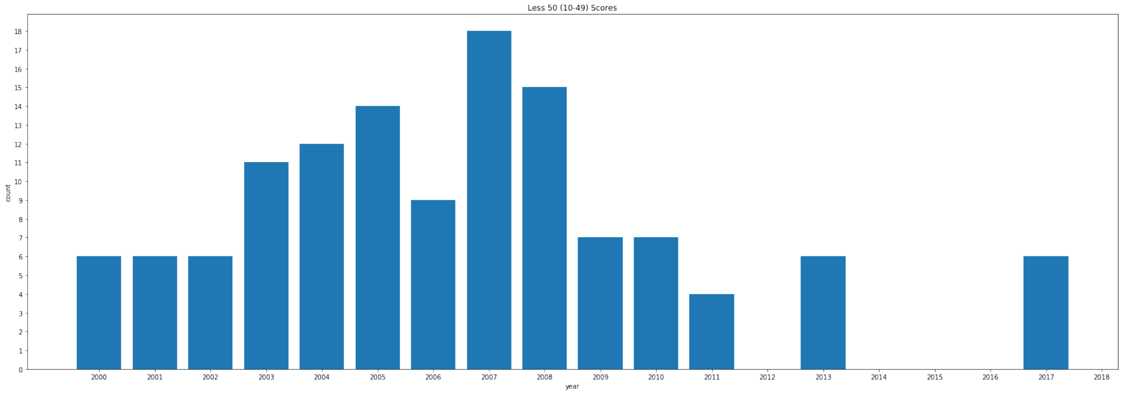
```
In [31]: half_cen_yr = dict(odi_half_cen['Date'].dt.year.value_counts())
         plt.rcParams['figure.figsize'] = [30, 10]
         plt.bar(*zip(*sorted(half_cen_yr.items())))
         plt.xticks(range(2000,2020))
         plt.title("Half-Century per year")
         plt.xlabel("Year")
         plt.ylabel("Count")
         plt.show()
```



```
In [32]: one_digit = all_odi_score[(all_odi_score['Runs'] >0) & (all_odi_score['Runs'] <10)]

In [33]: len(one_digit)

Out[33]: 66

In [34]: one_digit_data = dict(one_digit['Date'].dt.year.value_counts())
         plt.xticks(range(2000,2020))
         plt.yticks(range(0,12))
         plt.bar(*zip(*sorted(one_digit_data.items())))
         plt.title("Single Digit Scores(1-9)")
         plt.xlabel("year")
         plt.ylabel("count")
         plt.show()
```

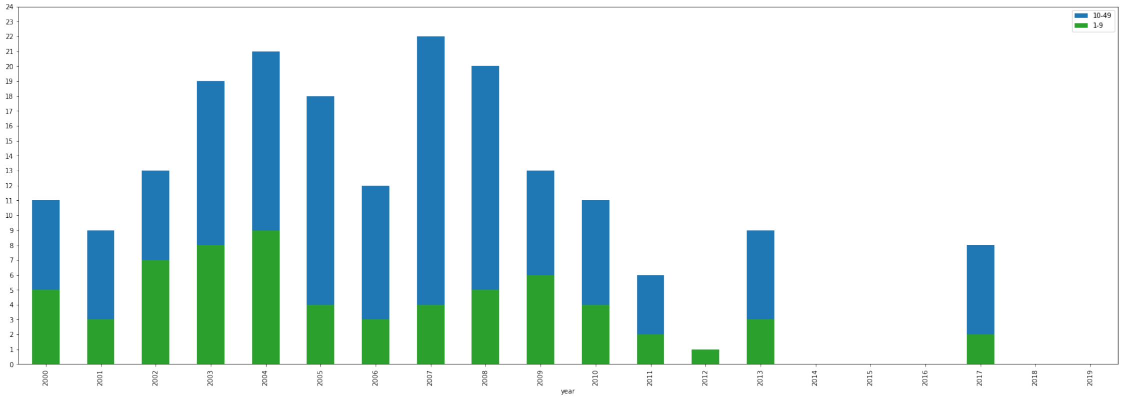


```
In [35]: less50 = all_odi_score[(all_odi_score['Runs'] >=10) & (all_odi_score['Runs'] <50)]

In [36]: len(less50)

Out[36]: 127

In [37]: less50_data = dict(less50['Date'].dt.year.value_counts())
         plt.xticks(range(2000,2020))
         plt.yticks(range(0,30))
         plt.bar(*zip(*sorted(less50_data.items())))
         plt.title("Less 50 (10-49) Scores")
         plt.xlabel("year")
         plt.ylabel("count")
         plt.show()
```

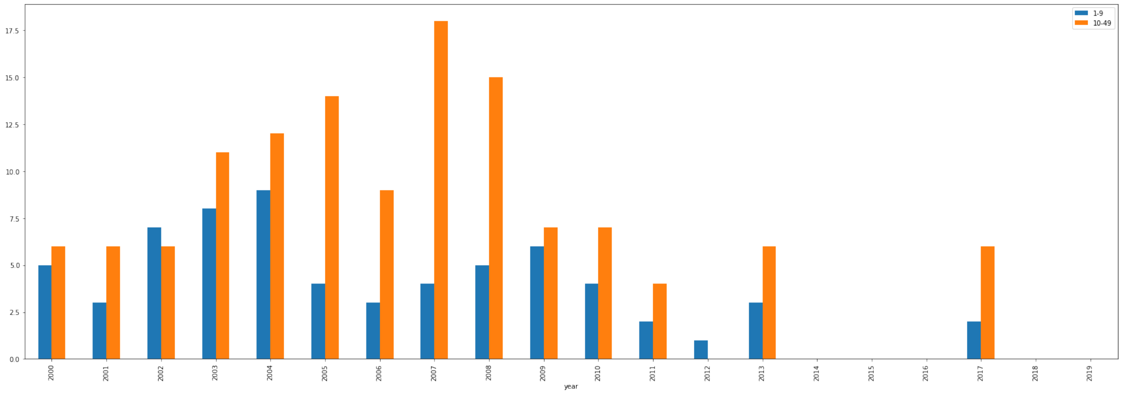


```
In [38]: one_digit_data.update({i:0 for i in set(range(2000,2020)) - set(one_digit_data.keys()
         a=pd.DataFrame(sorted(one_digit_data.items()),columns=["year","1-9"])
         b=pd.DataFrame(sorted(less50_data.items()),columns=["year","10-49"])
         less_50=pd.merge(a, b, on='year', how='outer')
         less_50.replace(np.nan, 0, inplace=True)
         less_50['10-49']=less_50['10-49'].astype(int)
         less_50['less 50'] = less_50.apply(lambda x: x['1-9'] + x['10-49'], axis=1)
```

```
In [39]: ax = less_50.plot(x="year", y="less 50", kind="bar", label='10-49')
         less_50.plot(x="year", y="1-9", kind="bar", ax=ax, color="C2")
         plt.yticks(range(0,25))
         plt.show()
```

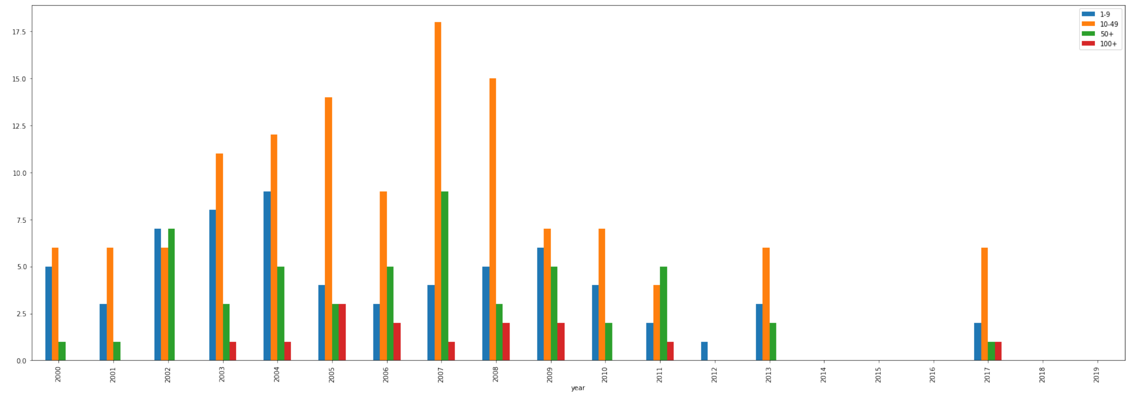


```
In [40]: o=less_50.drop(['less 50'], axis=1).set_index('year').plot.bar()
```

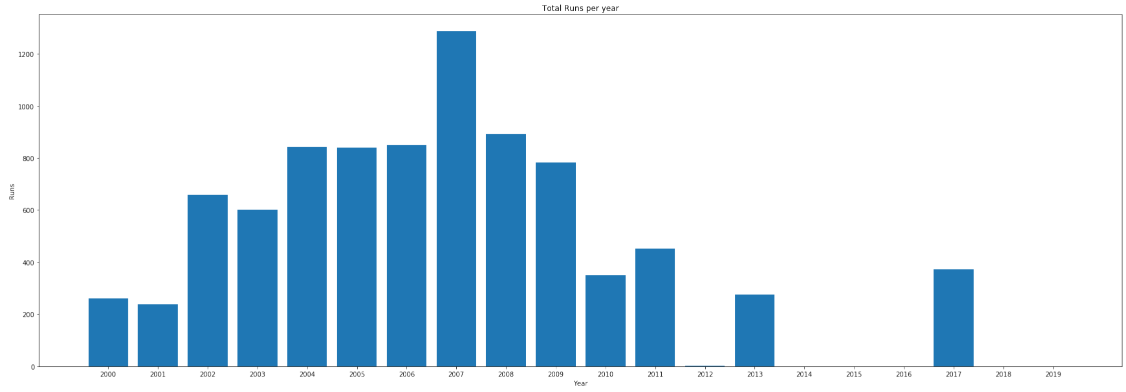


```
In [41]: a50=pd.DataFrame(sorted(half_cen_yr.items()), columns=['year','50+'])
         a100=pd.DataFrame(sorted(cen_yr.items()), columns=['year','100+'])
         all_score=pd.merge(less_50, pd.merge(a50, a100, on='year', how='outer'), on='year', ho
         all_score.replace(np.nan, 0, inplace=True)
         all_score['50+']= all_score['50+'].astype(int)
         all_score['100+']= all_score['100+'].astype(int)
```

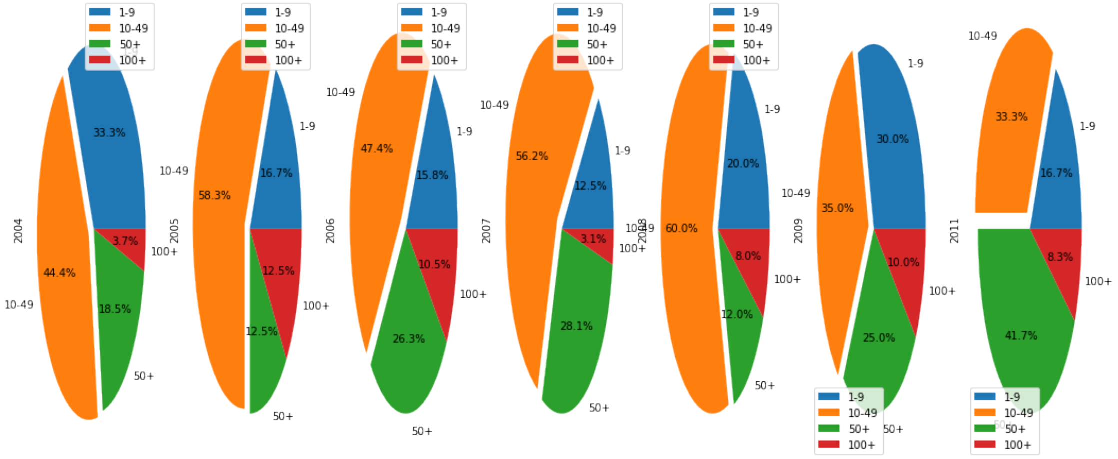
```
In [42]: o=all_score.drop(['less 50'], axis=1).set_index('year').plot.bar()
```



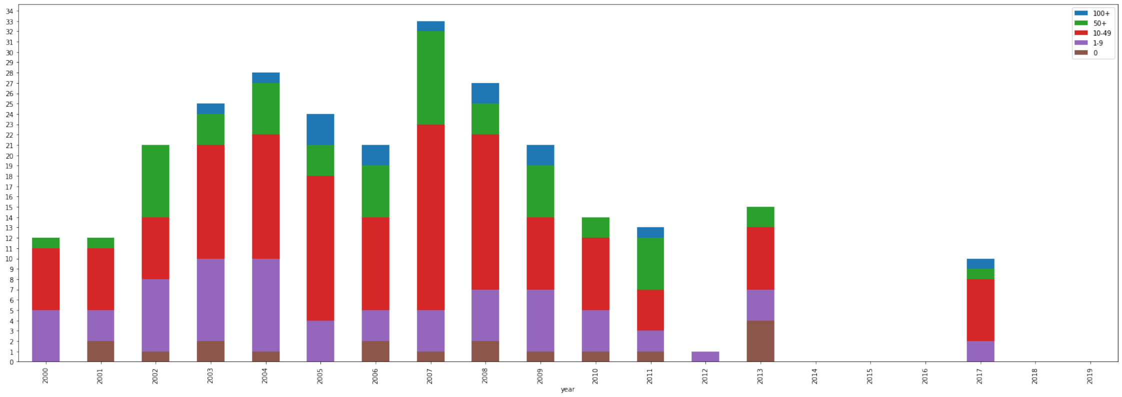
```
In [43]: all_score_df = all_score.set_index('year')

In [44]: all_odi_score["yr"] = all_odi_score.apply(lambda row: row['Date'].year, axis=1)
         yearly_odi_data = all_odi_score.groupby('yr')
         all_group = sorted(yearly_odi_data.groups.keys())
         info = {}
         for i in range(2000,2020):
             total = 0
             if i in all_group:
                 total = yearly_odi_data.get_group(i)['Runs'].sum()
             info[i]=total

In [45]: plt.xticks(range(2000,2020))
         plt.rcParams['figure.figsize'] = [30, 10]
         plt.bar(*zip(*sorted(info.items())))
         plt.rcParams['figure.figsize'] = [20,20]
         plt.title("Total Runs per year")
         plt.xlabel("Year")
         plt.ylabel("Runs")
         plt.show()
```



```
In [46]: explode = (0, 0.1, 0, 0)
         o=all_score_df.loc[[2004,2005,2006,2007,2008,2009,2011]].drop(['less 50'], axis=1).T.
```



```
In [47]: all_count=all_score.drop('less 50', axis=1)
         all_count = pd.merge(all_count, pd.DataFrame(sorted(duck_yr.items()), columns=["year"
         all_count.replace(np.nan, 0, inplace=True)
         all_count['0'] = all_count['0'].astype(int)
         all_count = all_count[['year', '0', '1-9', '10-49','50+','100+']]
         cols=list(all_count.columns)
         for i,j in enumerate(cols[2:],1):
             all_count[j] = all_count.apply(lambda row: row[cols[i]]+row[j], axis=1)
         plt.rcParams['figure.figsize'] = [30, 10]
         ax = all_count.plot(x="year", y="100+", kind="bar")
         all_count.plot(x="year", y="50+", kind="bar", ax=ax, color="C2")
         all_count.plot(x="year", y="10-49", kind="bar", ax=ax, color="C3")
         all_count.plot(x="year", y="1-9", kind="bar", ax=ax, color="C4")
         all_count.plot(x="year", y="0", kind="bar", ax=ax, color="C5")
         plt.yticks(range(0,35))
         plt.show()
```



#4.

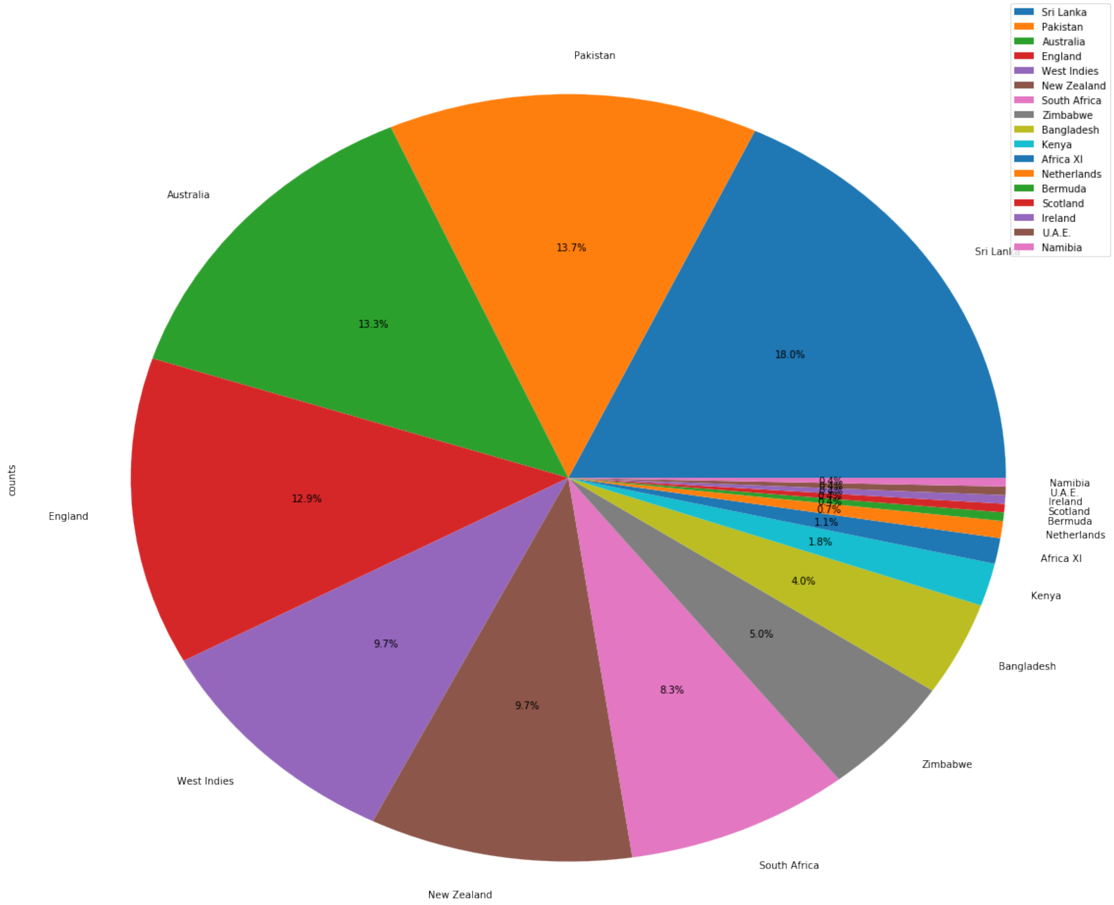
- Clearly 2007 was very significant year for him, where he scored most of the runs. He struggled during intial days.

Like In life, to continue to grow we need a good Mentor , he backed by the captain during that time.

- Comeback was not great, lost the momentum after 2012(cancer Recovery year). But truly a fighter.
  Health is precious.

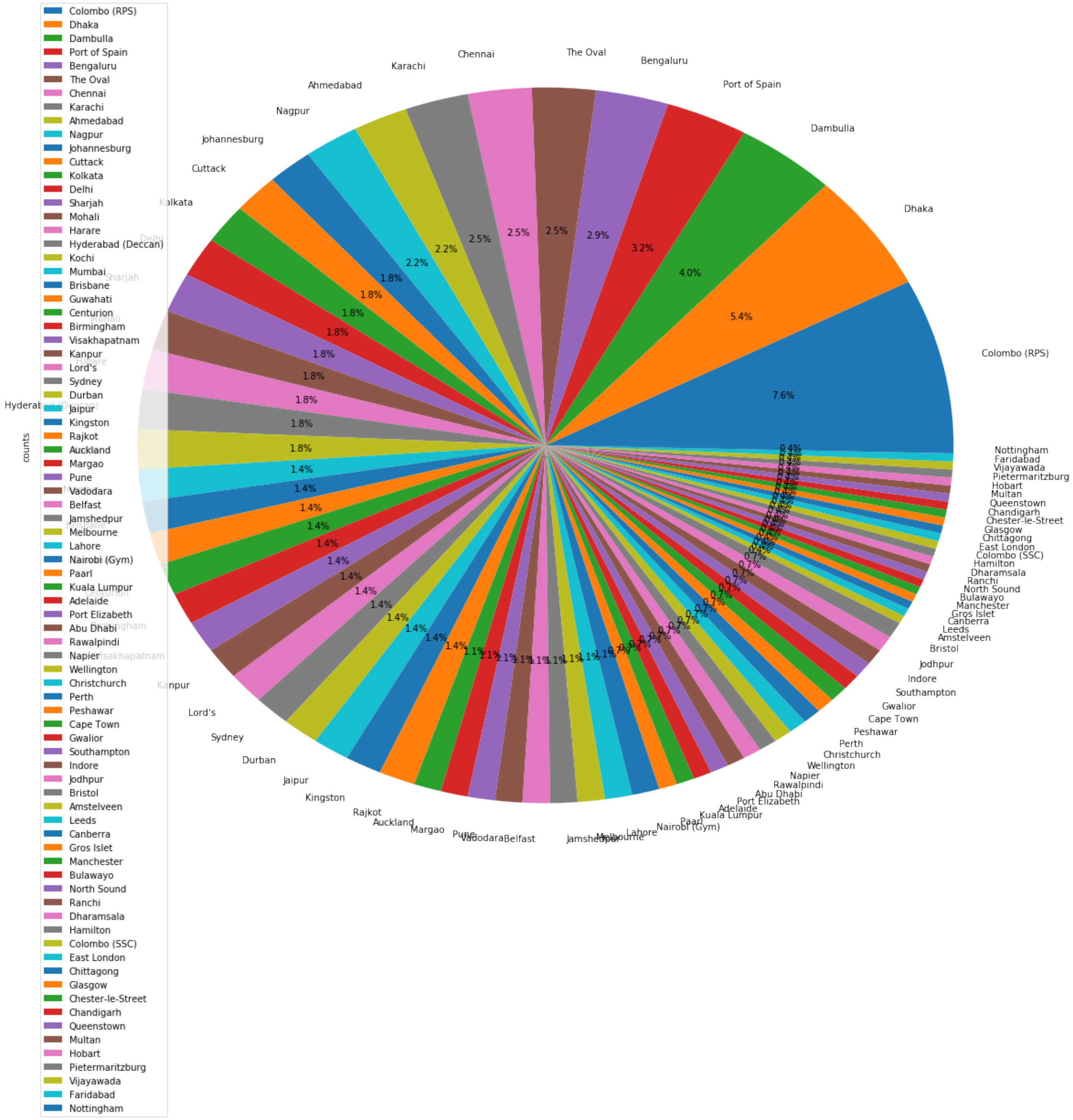
Played Against Team Most -

```
In [48]: o=all_odi_score['Opposition'].value_counts().reset_index(name='counts').set_index('inc
```



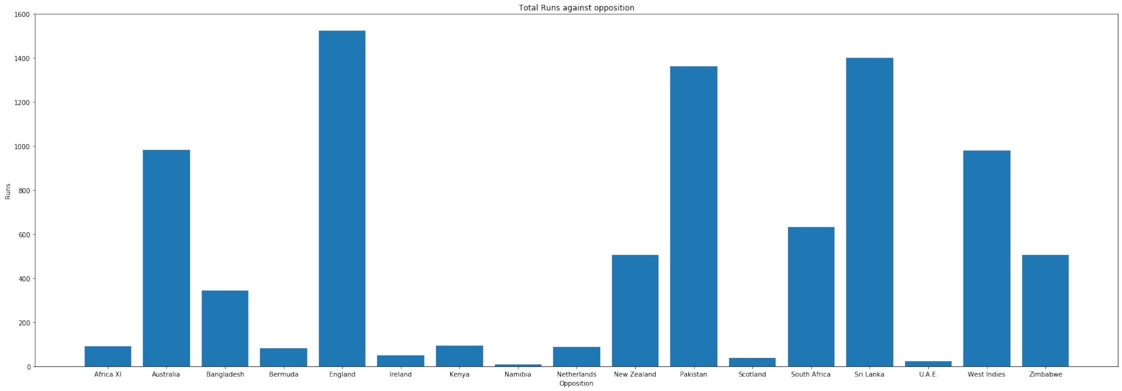
Ground He Played Most -

```
In [49]: o=all_odi_score['Ground'].value_counts().reset_index(name='counts').set_index('index')
```

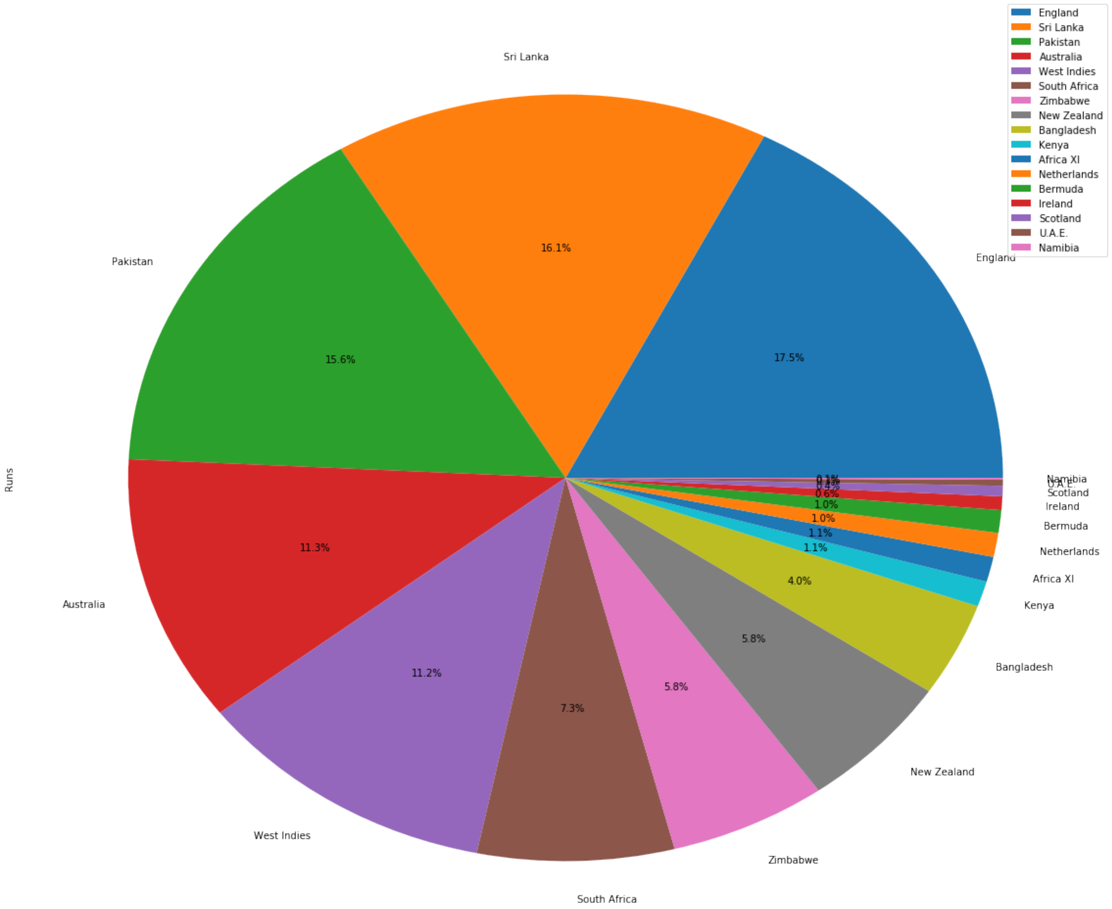


Runs per Team -

```
In [50]: team_group = all_odi_score.groupby("Opposition")
         group_teams = sorted(team_group.groups.keys())
         group_info = {}
         for i in group_teams:
             total = 0
             total = team_group.get_group(i)['Runs'].sum()
             group_info[i]=total
         plt.bar(*zip(*sorted(group_info.items())))
         plt.rcParams['figure.figsize'] = [30, 10]
         plt.title("Total Runs against opposition")
         plt.xlabel("Opposition")
         plt.ylabel("Runs")
         plt.show()
```



```
In [51]: o=pd.DataFrame(sorted(group_info.items(), key=lambda kv: kv[1], reverse=True), column
```



There are diffferent aspect we can analysis his performance - Win vs His batting contributions, Home vs Away performance etc.

```
In [ ]:
```