Mandatory assignment 1: Data analysis project - ATP Tennis scores

This project consists of three main sections which are trying to describe, analyse and derive useful conclusions in the realm of ATP results from 2000-2010. Section one will first of all combine all 10 individual csv-files into on merged file and hereafter apply the nessesary cleaning and structuring, which means column deletion and elimination of rows which are containing empty data. In section two we apply descriptive statistics and find useful

- **Step 1: Package import:** The first part of coding in our project is the import of useful packages and

In [53]:

```
import os
import glob
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf

from pandas import DataFrame as df
from scipy.stats import trim_mean, kurtosis
from scipy.stats.mstats import mode, gmean, hmean
```

- **Step 2: Setting of local environment:** This part is locating our main environment folder at the desktop called atp-matches-dataset.

In [54]:

```
#sti til mappe der skal arbejdes i
os.chdir("/Users/Christofferku/Desktop/atp-matches-dataset/")
```

- **Step 3: Deletion of old runs:** We will later save the merged csv in out local folder and we do therefore delete earlier version of this copy. Moreover the only files which have to be merged in step 4 is the annual data and cannot be contaminated by other csv files.

In [55]:

```
#hvis filen vi danner i forvejen findes slettes den så der kan køres en ny
if os.path.exists("Tennis_mod.csv"):
    os.remove("Tennis_mod.csv")
else:
    print('File does not exists')
```

- **Step 4: Merging all files :** all files with the extension 'csv' is chosen and merged into one joint table file in tennis_total by the function panda.concat

In [56]:

```
#alle filer med format csv medtages og samles i tennis_total
extension = 'csv'
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]
Tennis_total = pd.concat([pd.read_csv(f) for f in all_filenames ])
```

> **Description and analysis:** xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 5: Choosing the relevant columns and information:** We have decided to examinate the ....

In [57]:

```
#Vælger hvilke kolonner i tennis_total vi vil have med og danner det endelige da
taset Tennis_mod
keep_col = ['tourney_id','tourney_name','surface','draw_size','winner_ht', 'winn
er_age', 'winner_rank', 'winner_rank_points', 'winner_ioc']
Tennis_mod=Tennis_total[keep_col]
```

> **Description and analysis:** xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 6: Filtering blank cells :** We remove all blank cells in the

In [58]:

```
#fjerner rækker med blanke celler
Tennis_mod = Tennis_mod[Tennis_mod['tourney_id'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['tourney_name'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['surface'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['draw_size'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_ht'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_age'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_rank'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_rank_points'].notnull()]
```

Description and analysis: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 7: Save the final table in local folder :** the file is

In [59]:

```
#Laver tennis_mod til csv som gemmes i samme mappe med stien
Tennis_mod.to_csv( "Tennis_mod.csv", index=False, encoding='utf-8-sig')
```

Description and analysis: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 8: Print the final table :** We print the tennis_mod table as an overview fo.....

In [60]:

```
#et stk. printet samlet tabel
print(Tennis_mod)
```

```
     tourney_id                tourney_name surface  draw_size  win
ner_ht  \
0      2000-717                     Orlando    Clay         32
185.0
1      2000-717                     Orlando    Clay         32
183.0
2      2000-717                     Orlando    Clay         32
185.0
3      2000-717                     Orlando    Clay         32
183.0
4      2000-717                     Orlando    Clay         32
185.0
5      2000-717                     Orlando    Clay         32
185.0
6      2000-717                     Orlando    Clay         32
178.0
7      2000-717                     Orlando    Clay         32
178.0
8      2000-717                     Orlando    Clay         32
183.0
9      2000-717                     Orlando    Clay         32
188.0
10     2000-717                     Orlando    Clay         32
173.0
11     2000-717                     Orlando    Clay         32
180.0
12     2000-717                     Orlando    Clay         32
185.0
13     2000-717                     Orlando    Clay         32
188.0
14     2000-717                     Orlando    Clay         32
185.0
15     2000-717                     Orlando    Clay         32
185.0
16     2000-717                     Orlando    Clay         32
183.0
17     2000-717                     Orlando    Clay         32
185.0
18     2000-717                     Orlando    Clay         32
185.0
19     2000-717                     Orlando    Clay         32
178.0
20     2000-717                     Orlando    Clay         32
183.0
21     2000-717                     Orlando    Clay         32
180.0
22     2000-717                     Orlando    Clay         32
188.0
23     2000-717                     Orlando    Clay         32
185.0
24     2000-717                     Orlando    Clay         32
183.0
25     2000-717                     Orlando    Clay         32
178.0
26     2000-717                     Orlando    Clay         32
183.0
27     2000-717                     Orlando    Clay         32
185.0
28     2000-717                     Orlando    Clay         32
183.0
29     2000-717                     Orlando    Clay         32
```

```
183.0
...         ...                          ...     ...         ...
...
3027  2010-D075  Davis Cup WG PO: COL vs USA    Clay          4
188.0
3028  2010-D075  Davis Cup WG PO: COL vs USA    Clay          4
188.0
3029  2010-D075  Davis Cup WG PO: COL vs USA    Clay          4
188.0
3030  2010-D076  Davis Cup WG PO: ISR vs AUT    Hard          4
175.0
3031  2010-D076  Davis Cup WG PO: ISR vs AUT    Hard          4
183.0
3032  2010-D076  Davis Cup WG PO: ISR vs AUT    Hard          4
183.0
3034  2010-D077  Davis Cup WG PO: GER vs RSA    Clay          4
178.0
3035  2010-D077  Davis Cup WG PO: GER vs RSA    Clay          4
190.0
3036  2010-D077  Davis Cup WG PO: GER vs RSA    Clay          4
190.0
3037  2010-D077  Davis Cup WG PO: GER vs RSA    Clay          4
190.0
3038  2010-D078  Davis Cup WG PO: SWE vs ITA    Hard          4
188.0
3039  2010-D078  Davis Cup WG PO: SWE vs ITA    Hard          4
193.0
3040  2010-D078  Davis Cup WG PO: SWE vs ITA    Hard          4
193.0
3041  2010-D078  Davis Cup WG PO: SWE vs ITA    Hard          4
178.0
3042  2010-D079  Davis Cup WG PO: IND vs BRA    Hard          4
188.0
3043  2010-D079  Davis Cup WG PO: IND vs BRA    Hard          4
175.0
3044  2010-D079  Davis Cup WG PO: IND vs BRA    Hard          4
180.0
3045  2010-D079  Davis Cup WG PO: IND vs BRA    Hard          4
190.0
3046  2010-D080  Davis Cup WG PO: AUS vs BEL    Hard          4
180.0
3047  2010-D080  Davis Cup WG PO: AUS vs BEL    Hard          4
168.0
3048  2010-D080  Davis Cup WG PO: AUS vs BEL    Hard          4
168.0
3049  2010-D080  Davis Cup WG PO: AUS vs BEL    Hard          4
178.0
3050  2010-D081  Davis Cup WG PO: KAZ vs SUI    Hard          4
185.0
3051  2010-D081  Davis Cup WG PO: KAZ vs SUI    Hard          4
183.0
3052  2010-D081  Davis Cup WG PO: KAZ vs SUI    Hard          4
185.0
3053  2010-D081  Davis Cup WG PO: KAZ vs SUI    Hard          4
183.0
3054  2010-D082  Davis Cup WG PO: ROU vs ECU    Clay          4
198.0
3055  2010-D082  Davis Cup WG PO: ROU vs ECU    Clay          4
178.0
3056  2010-D082  Davis Cup WG PO: ROU vs ECU    Clay          4
185.0
```

```
3057  2010-D082  Davis Cup WG PO: ROU vs ECU    Clay          4
178.0

      winner_age  winner_rank  winner_rank_points winner_ioc
0      27.181383        113.0                351.0        FRA
1      19.756331        352.0                 76.0        CHI
2      20.881588        103.0                380.0        THA
3      30.047912        107.0                371.0        NED
4      30.075291         74.0                543.0        AUS
5      22.020534         92.0                429.0        CZE
6      30.368241        120.0                322.0        ARG
7      23.739904         79.0                516.0        USA
8      20.558522         89.0                464.0        CHI
9      25.538672        125.0                315.0        SVK
10     22.001369        165.0                221.0        USA
11     34.872005         72.0                550.0        ITA
12     22.869268        205.0                160.0        ROU
13     22.447639        100.0                385.0        GER
14     19.783710        148.0                253.0        BEL
15     23.460643        216.0                155.0        PAR
16     19.756331        352.0                 76.0        CHI
17     20.881588        103.0                380.0        THA
18     22.020534         92.0                429.0        CZE
19     30.368241        120.0                322.0        ARG
20     20.558522         89.0                464.0        CHI
21     34.872005         72.0                550.0        ITA
22     22.447639        100.0                385.0        GER
23     23.460643        216.0                155.0        PAR
24     19.756331        352.0                 76.0        CHI
25     30.368241        120.0                322.0        ARG
26     20.558522         89.0                464.0        CHI
27     23.460643        216.0                155.0        PAR
28     19.756331        352.0                 76.0        CHI
29     20.558522         89.0                464.0        CHI
...          ...          ...                  ...        ...
3027   28.761123         19.0               1931.0        USA
3028   22.795346         61.0                801.0        COL
3029   28.761123         19.0               1931.0        USA
3030   25.442847         85.0                618.0        ISR
3031   29.311431         13.0               2605.0        AUT
3032   29.311431         13.0               2605.0        AUT
3034   26.910335         31.0               1270.0        GER
3035   26.940452         45.0                978.0        GER
3036   24.602327        101.0                515.0        GER
3037   26.940452         45.0                978.0        GER
3038   29.166324         50.0                915.0        ITA
3039   26.080767          5.0               4910.0        SWE
3040   26.080767          5.0               4910.0        SWE
3041   23.307324         71.0                720.0        ITA
3042   22.704997         27.0               1455.0        BRA
3043   29.727584         75.0                683.0        BRA
3044   25.579740        113.0                478.0        IND
3045   30.527036        479.0                 64.0        IND
3046   29.549624         36.0               1135.0        AUS
3047   29.650924         79.0                650.0        BEL
3048   29.650924         79.0                650.0        BEL
3049   26.502396        117.0                474.0        BEL
3050   23.145791         39.0               1080.0        KAZ
3051   22.715948         81.0                628.0        KAZ
3052   23.145791         39.0               1080.0        KAZ
3053   22.715948         81.0                628.0        KAZ
```

```
3054   29.147159         54.0              905.0        ROU
3055   25.631759        130.0              413.0        ROU
3056   28.303901        183.0              272.0        ROU
3057   25.631759        130.0              413.0        ROU

[33488 rows x 9 columns]
```

Description and analysis: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 9: Descriptive analysis :** we use the describe from the imported package dsstat...

In [61]:

```
#et stk. deskriptiv analyse
DataDescribe=Tennis_mod.describe()
print(DataDescribe)
```

```
          draw_size     winner_ht    winner_age   winner_rank  \
count  33488.000000  33488.000000  33488.000000  33488.000000
mean      55.030936    185.055393     25.403898     69.081402
std       37.830872      6.378877      3.424202     98.794001
min        4.000000    168.000000     15.824778      1.000000
25%       32.000000    180.000000     22.858316     18.000000
50%       32.000000    185.000000     25.229295     44.000000
75%       64.000000    190.000000     27.759754     84.000000
max      128.000000    208.000000     38.291581   1554.000000

       winner_rank_points
count        33488.000000
mean          1311.180065
std           1497.596499
min              1.000000
25%            509.000000
50%            855.000000
75%           1498.000000
max          15390.000000
```

Description and analysis: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 10: OLS :** We remo

In [62]:

```
#et stk. OLS
results = smf.ols('winner_rank ~ winner_age + winner_ht', data=Tennis_mod).fit()
print(results.summary())
```

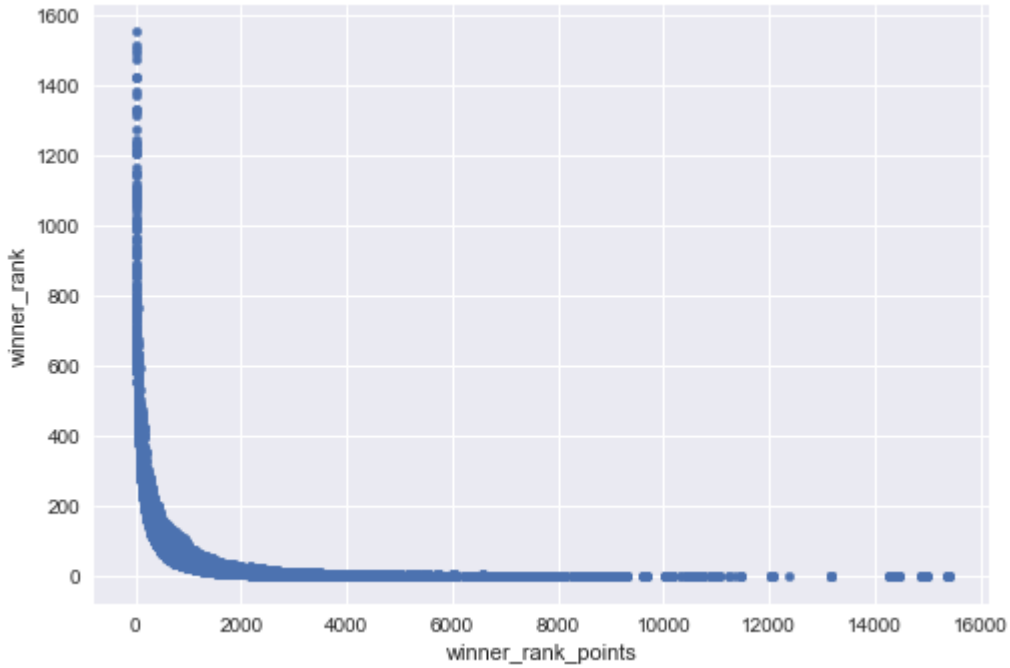
```
                            OLS Regression Results
==============================================================================
Dep. Variable:            winner_rank   R-squared:
0.001
Model:                            OLS   Adj. R-squared:
0.001
Method:                 Least Squares   F-statistic:
16.35
Date:                Mon, 25 Mar 2019   Prob (F-statistic):
8.00e-08
Time:                        22:19:23   Log-Likelihood:            -
2.0131e+05
No. Observations:               33488   AIC:
4.026e+05
Df Residuals:                   33485   BIC:
4.027e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
Intercept    149.4017     16.470      9.071      0.000     117.120
181.683
winner_age    -0.6861      0.158     -4.344      0.000      -0.996
-0.377
winner_ht     -0.3398      0.085     -4.008      0.000      -0.506
-0.174
==============================================================================
Omnibus:                    35927.891   Durbin-Watson:
1.289
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          3
146417.992
Skew:                           5.440   Prob(JB):
0.00
Kurtosis:                      49.223   Cond. No.
5.70e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors
is correctly specified.
[2] The condition number is large, 5.7e+03. This might indicate that
there are
strong multicollinearity or other numerical problems.
```

Description and analysis: xxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

- **Step 11: Plot :** We remo

In [63]:

```
#et stk. plot
plt.style.use('seaborn')
Tennis_mod.plot(x='winner_rank_points', y='winner_rank', kind='scatter')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
import os
import glob
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf

from pandas import DataFrame as df
from scipy.stats import trim_mean, kurtosis
from scipy.stats.mstats import mode, gmean, hmean

#sti til mappe der skal arbejdes i
os.chdir("/Users/Christofferku/Desktop/atp-matches-dataset/")

#hvis filen vi danner i forvejen findes slettes den så der kan køres en ny
if os.path.exists("Tennis_mod.csv"):
    os.remove("Tennis_mod.csv")
else:
    print('File does not exists')

#alle filer med format csv medtages og samles i tennis_total
extension = 'csv'
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]
Tennis_total = pd.concat([pd.read_csv(f) for f in all_filenames ])

#Vælger hvilke kolonner i tennis_total vi vil have med og danner det endelige da
taset Tennis_mod
keep_col = ['tourney_id','tourney_name','surface','draw_size','winner_ht', 'winn
er_age', 'winner_rank', 'winner_rank_points', 'winner_ioc']
Tennis_mod=Tennis_total[keep_col]

#fjerner rækker med blanke celler
Tennis_mod = Tennis_mod[Tennis_mod['tourney_id'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['tourney_name'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['surface'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['draw_size'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_ht'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_age'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_rank'].notnull()]
Tennis_mod = Tennis_mod[Tennis_mod['winner_rank_points'].notnull()]

#Laver tennis_mod til csv som gemmes i samme mappe med stien
Tennis_mod.to_csv( "Tennis_mod.csv", index=False, encoding='utf-8-sig')

#et stk. printet samlet tabel
print(Tennis_mod)

#et stk. deskriptiv analyse
DataDescribe=Tennis_mod.describe()
print(DataDescribe)

#et stk. OLS
results = smf.ols('winner_rank ~ winner_age + winner_ht', data=Tennis_mod).fit()
print(results.summary())

#et stk. plot
plt.style.use('seaborn')
Tennis_mod.plot(x='winner_rank_points', y='winner_rank', kind='scatter')
```

```
plt.show()


#bumbum.
```

In [ ]:

```

```

In [ ]:

```

```