# Dependencies and Setup  
import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd  
import re  
import datetime as dt  
import pandas\_profiling as pp  
import numpy as np  
%matplotlib inline  
plt.show()

# Importing the csv  
df= pd.read\_csv('fidelity\_mutual\_funds\_return\_w\_risk.csv')  
df.head()

name morningstar\_category \  
0 Baron Partners Fund Institutional Shares (BPTIX) Large Growth   
1 Baron Partners Fund Retail Shares (BPTRX) Large Growth   
2 Morgan Stanley Institutional Fund, Inc. Incept... Small Growth   
3 Morgan Stanley Institutional Fund, Inc. Incept... Small Growth   
4 Morgan Stanley Institutional Fund, Inc. Incept... Small Growth   
  
 ytdDaily yr1 yr3 yr5 yr10 life\_of\_fund net\_expense\_ratio \  
0 44.60 110.27 65.56 47.54 29.02 27.02 1.30   
1 44.28 109.72 65.13 47.15 28.68 20.45 1.56   
2 20.05 81.15 55.11 38.37 22.37 14.14 1.00   
3 19.75 80.67 54.70 37.98 22.01 13.83 1.35   
4 NaN 79.39 53.46 36.90 21.11 13.00 2.10   
  
 gross\_expense\_ratio morningstar\_rating\_overall risk std\_dev \  
0 1.30 1137.0 6 40.41   
1 1.56 1137.0 6 40.39   
2 1.19 574.0 7 40.44   
3 1.45 574.0 7 40.48   
4 2.27 574.0 7 40.42   
  
 sharpe\_ratio\_3\_yr beta r2 minimum\_investment last\_dividend   
0 1.60 1.51 0.63 1000000.0 0.2224   
1 1.59 1.51 0.63 2500.0 0.1243   
2 1.34 1.40 0.71 5000000.0 0.0000   
3 1.32 1.40 0.71 2500.0 0.0000   
4 1.30 1.40 0.71 2500.0 NaN

# dataset structure  
df.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 9626 entries, 0 to 9625  
Data columns (total 18 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 name 9626 non-null object   
 1 morningstar\_category 9626 non-null object   
 2 ytdDaily 9359 non-null float64  
 3 yr1 9520 non-null float64  
 4 yr3 9204 non-null float64  
 5 yr5 8922 non-null float64  
 6 yr10 7283 non-null float64  
 7 life\_of\_fund 9613 non-null float64  
 8 net\_expense\_ratio 9625 non-null float64  
 9 gross\_expense\_ratio 9624 non-null float64  
 10 morningstar\_rating\_overall 9102 non-null float64  
 11 risk 9626 non-null int64   
 12 std\_dev 9154 non-null float64  
 13 sharpe\_ratio\_3\_yr 9154 non-null float64  
 14 beta 8189 non-null float64  
 15 r2 8189 non-null float64  
 16 minimum\_investment 9624 non-null float64  
 17 last\_dividend 9178 non-null float64  
dtypes: float64(15), int64(1), object(2)  
memory usage: 1.3+ MB

# 5 number summary  
df.describe()

ytdDaily yr1 yr3 yr5 yr10 \  
count 9359.000000 9520.000000 9204.000000 8922.000000 7283.000000   
mean 12.841688 25.532315 12.170789 9.872851 8.455235   
std 12.137653 20.381654 8.276280 7.080595 5.485703   
min -22.090000 -37.660000 -29.380000 -23.430000 -19.020000   
25% 1.890000 6.010000 5.460000 3.740000 3.760000   
50% 11.970000 26.135000 11.265000 9.120000 8.180000   
75% 22.070000 40.180000 16.950000 14.160000 12.530000   
max 83.850000 146.460000 65.560000 47.540000 29.020000   
  
 life\_of\_fund net\_expense\_ratio gross\_expense\_ratio \  
count 9613.000000 9625.000000 9624.000000   
mean 7.796146 1.154081 1.412057   
std 5.245337 0.570965 1.333127   
min -11.740000 0.000000 0.000000   
25% 4.650000 0.760000 0.850000   
50% 7.080000 1.050000 1.190000   
75% 10.170000 1.490000 1.700000   
max 82.450000 5.250000 46.990000   
  
 morningstar\_rating\_overall risk std\_dev \  
count 9102.000000 9626.000000 9154.000000   
mean 464.152604 5.271556 14.547154   
std 362.456368 1.665109 8.134948   
min 10.000000 0.000000 0.180000   
25% 186.000000 4.000000 7.192500   
50% 361.000000 6.000000 16.080000   
75% 631.000000 6.000000 19.870000   
max 1250.000000 9.000000 59.740000   
  
 sharpe\_ratio\_3\_yr beta r2 minimum\_investment \  
count 9154.000000 8189.000000 8189.000000 9.624000e+03   
mean 0.784549 0.788115 0.805881 2.455934e+05   
std 0.391578 1.218248 0.282312 8.381462e+05   
min -2.450000 -21.000000 0.000000 0.000000e+00   
25% 0.550000 0.820000 0.810000 2.500000e+03   
50% 0.780000 0.960000 0.930000 2.500000e+03   
75% 1.030000 1.050000 0.970000 5.000000e+03   
max 2.610000 8.750000 1.000000 1.000000e+07   
  
 last\_dividend   
count 9178.000000   
mean 0.116582   
std 0.499280   
min 0.000000   
25% 0.013883   
50% 0.034480   
75% 0.118995   
max 24.012771

# Data Profiling

# Perform data profiling  
profile\_report = pp.ProfileReport(df)  
profile\_report.to\_file("fidelity\_funds\_profiling.html")  
profile\_report

{"model\_id":"8504419f169d465eacc8ba53bb35bea6","version\_major":2,"version\_minor":0}

{"model\_id":"db244866901c4fa1b4f03b1f327e94af","version\_major":2,"version\_minor":0}

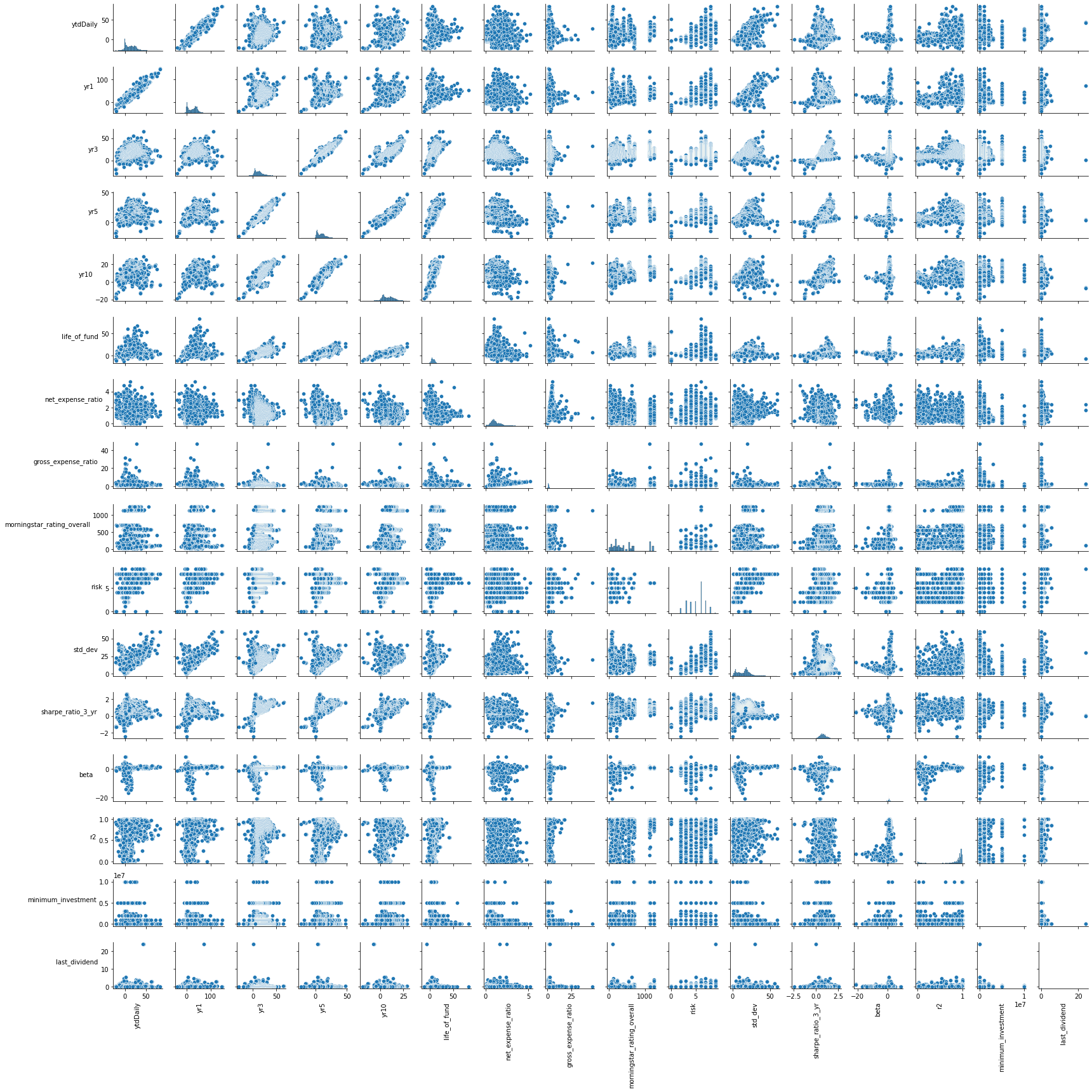
{"model\_id":"888eadaf23bf496ebd251ba5d5978681","version\_major":2,"version\_minor":0}

{"model\_id":"20cec21a911645758b097ed864882553","version\_major":2,"version\_minor":0}

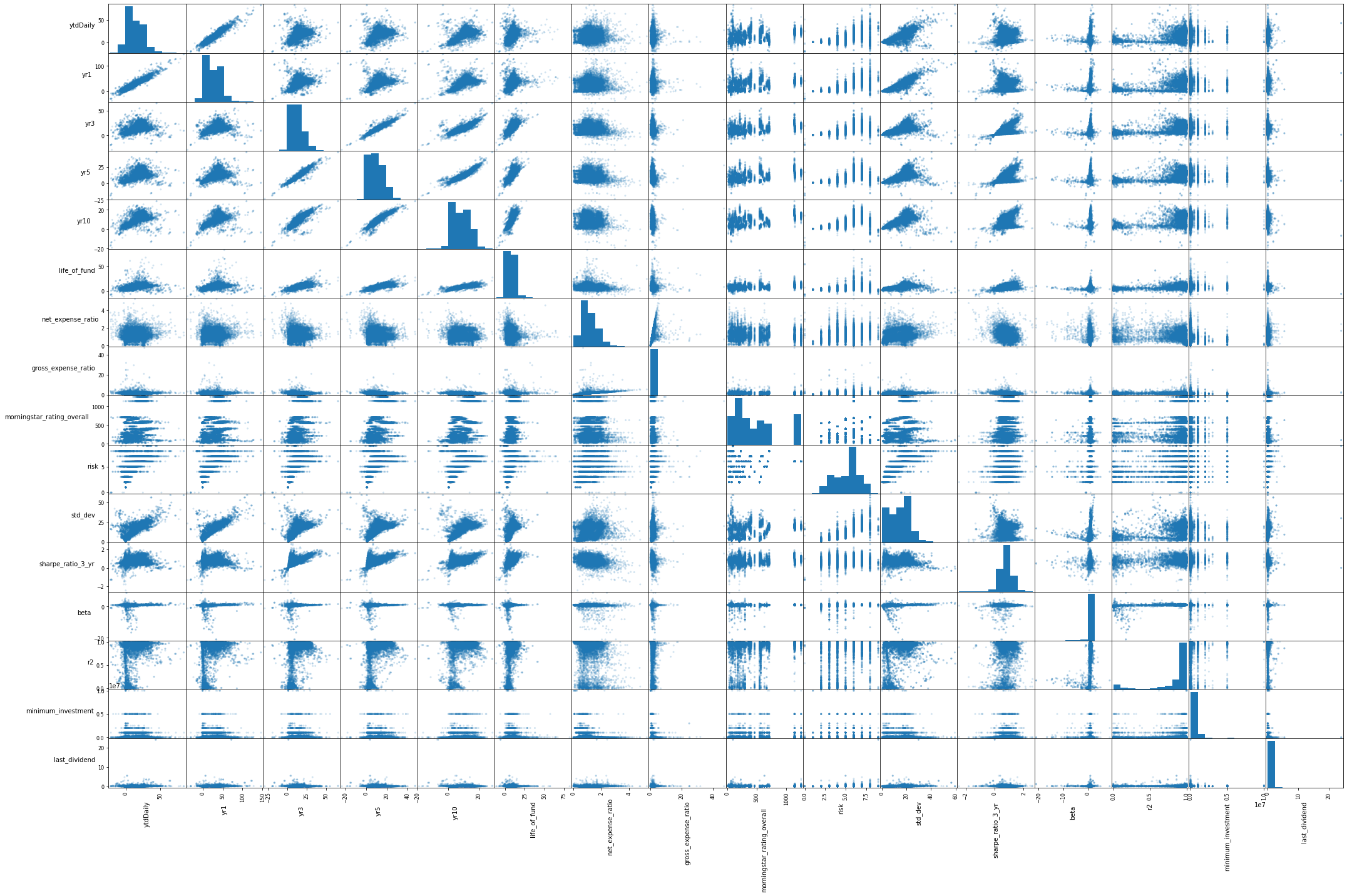
<IPython.core.display.HTML object>

# Pair plot

# Create the default pairplot using seaborn  
g =sns.pairplot(df, height =1.5)  
for ax in g.axes.flat:  
 # rotate x axis labels  
 ax.set\_xlabel(ax.get\_xlabel(), rotation = 90)  
 # rotate y axis labels  
 ax.set\_ylabel(ax.get\_ylabel(), rotation = 0)  
 # set y labels alignment  
 ax.yaxis.get\_label().set\_horizontalalignment('right')  
plt.tight\_layout()  
plt.savefig('pairplot\_all.png', facecolor='w')  
plt.show()



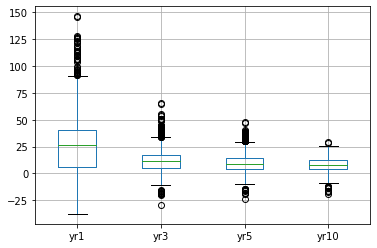
# Create the default scatter matrix using pandas  
axes = pd.plotting.scatter\_matrix(df, alpha=0.2,figsize=(30, 20))  
  
for ax in axes.flatten():  
 ax.xaxis.label.set\_rotation(90)  
 ax.yaxis.label.set\_rotation(0)  
 ax.yaxis.label.set\_ha('right')  
  
plt.tight\_layout()  
plt.gcf().subplots\_adjust(wspace=0, hspace=0)  
plt.savefig('pandas\_scatter\_matrix.png', facecolor='w')  
plt.show()



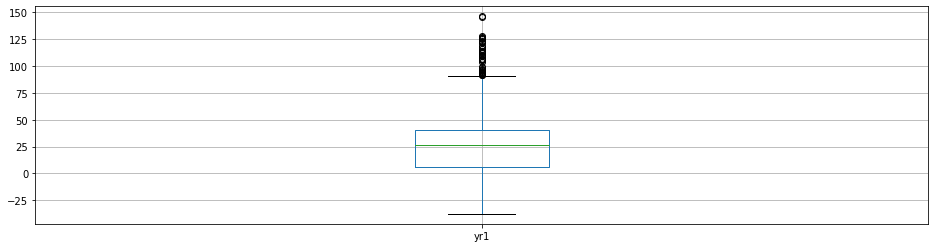
# Box Plot of returns

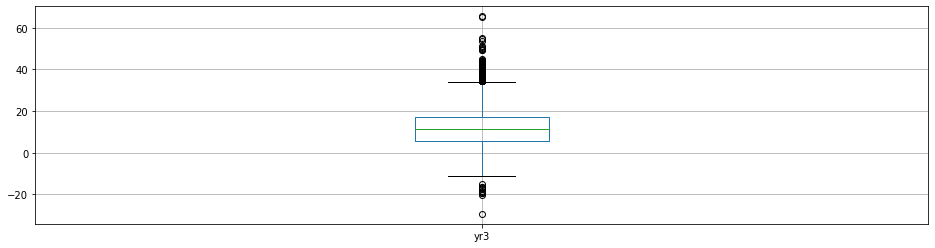
df.boxplot(['yr1', 'yr3', 'yr5', 'yr10'])

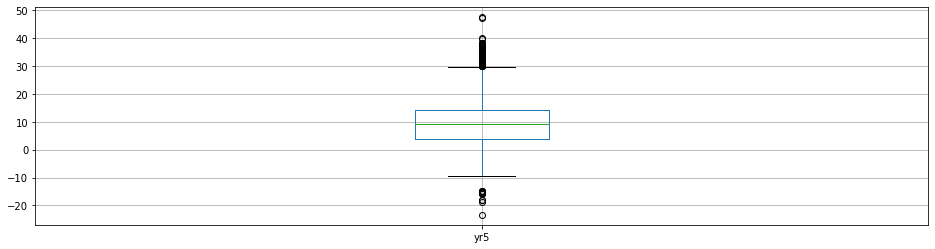
<AxesSubplot:>

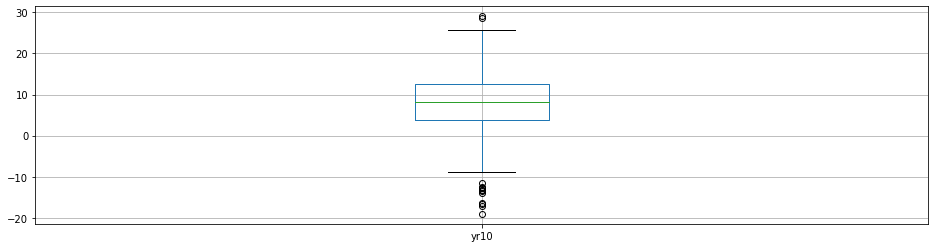


for column in df.columns[3:7]:  
 plt.figure(figsize = (16,4))  
 df.boxplot([column])



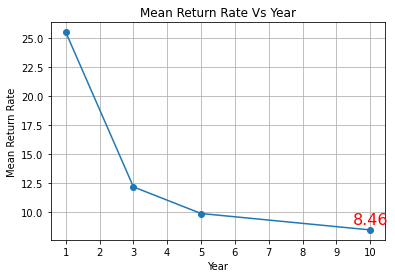






# Mean trend line

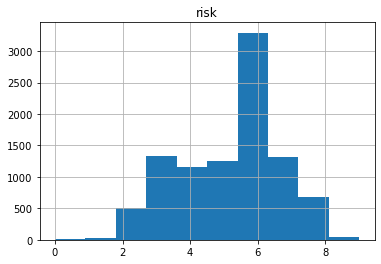
year = [1,3,5,10]  
mean\_return = [df['yr1'].mean(), df['yr3'].mean() , df['yr5'].mean(), df['yr10'].mean()]  
   
plt.plot(year, mean\_return, marker = 'o')  
plt.grid(zorder=0)  
plt.title('Mean Return Rate Vs Year')  
plt.xlabel('Year')  
plt.ylabel('Mean Return Rate')  
plt.xticks(np.arange(min(year), max(year)+1, 1.0))  
plt.annotate(round(min(mean\_return),2),(max(year)-.5, min(mean\_return)+.5),fontsize=16,color="red")  
plt.show()



# Distribution of Risk

df.hist(column = 'risk')

array([[<AxesSubplot:title={'center':'risk'}>]], dtype=object)



# Distribution of Category

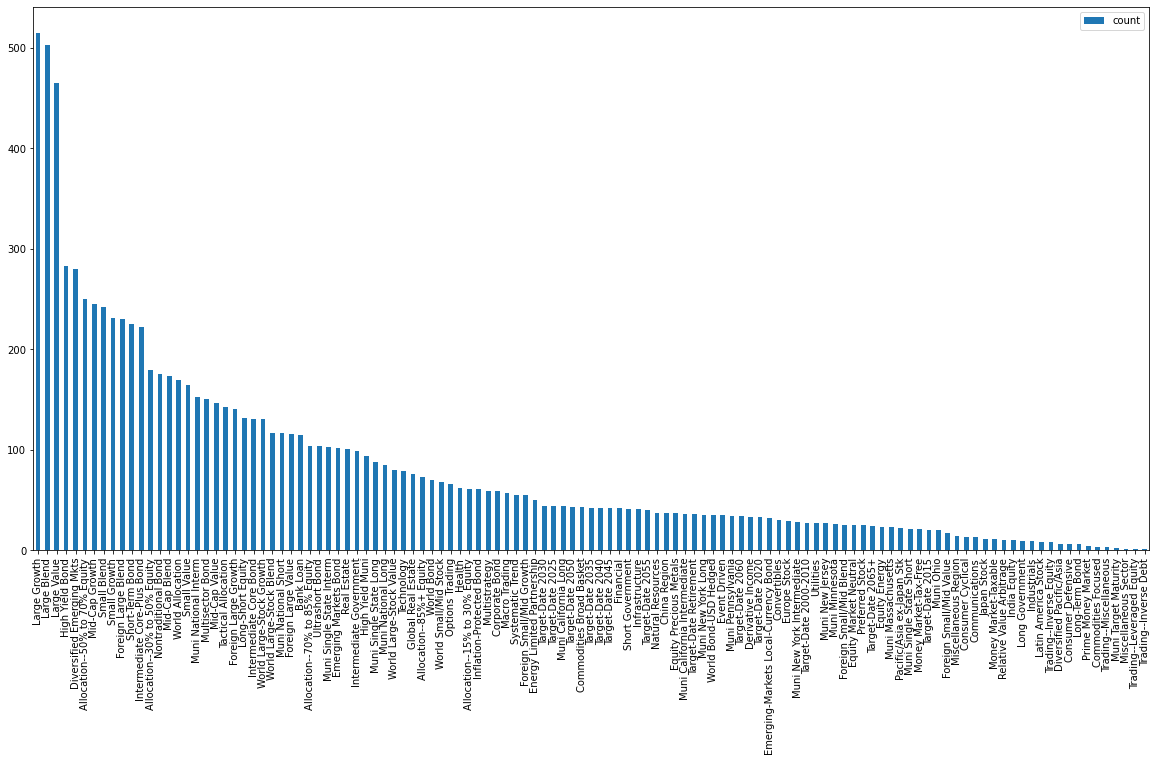
from collections import Counter

# Unique categories  
df['morningstar\_category'].nunique()

119

cat\_cnt = Counter(df['morningstar\_category'])  
cat\_cnt\_df=pd.DataFrame.from\_dict(cat\_cnt, orient='index', columns = ['count'])  
cat\_cnt\_df.sort\_values('count', ascending= False, inplace=True)  
cat\_cnt\_df.plot(kind='bar', figsize = (20,10))

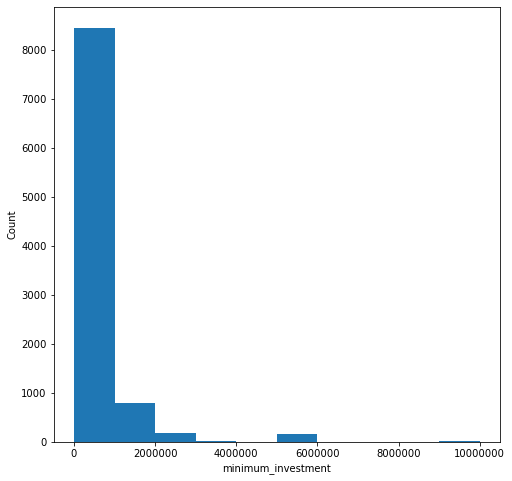
<AxesSubplot:>



# Distribution of minimum investments

plt.figure(figsize=(8,8))  
plt.hist(df['minimum\_investment'])   
plt.ticklabel\_format(axis="x", style="sci", scilimits=(0,10))  
plt.ylabel('Count')  
plt.xlabel('minimum\_investment')

Text(0.5, 0, 'minimum\_investment')



# Distribution of returns

for column in df.columns[3:7]:  
 sns.histplot(df[column], kde=True, bins=20)  
 plt.show()

