In [12]:
```
#라이브러리 불러오기

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from scipy.stats import norm
from scipy import stats
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

In [2]:
```
df=pd.read_csv("z_and_x.csv")
```

In [4]:
```
df.isna().sum()
```

Out[4]:
```
number     0
status     0
z          0
x1         0
x2         0
          ..
x2996      0
x2997      0
x2998      0
x2999      0
x3000      0
Length: 3003, dtype: int64
```

In [7]:
```
df.corr()
```

Out[7]:

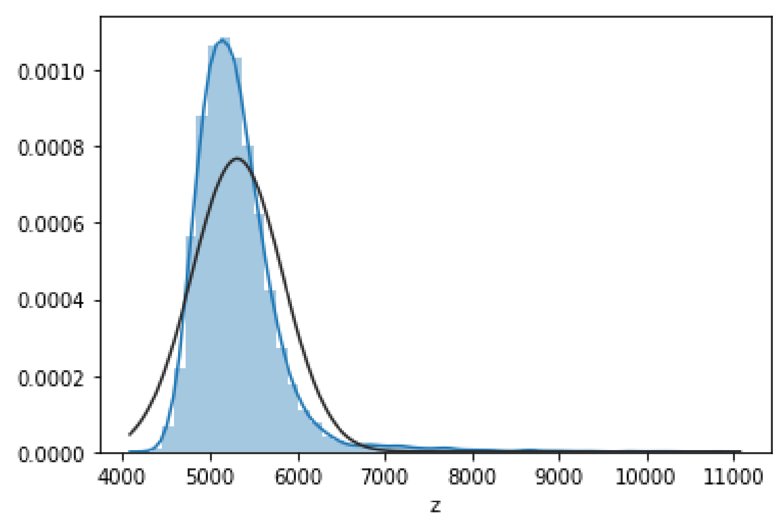
	number	z	x1	x2	x3	x4	x5	x6
number	1.000000	0.015484	-0.860857	-0.686547	0.063235	0.312558	0.323131	0.247730
z	0.015484	1.000000	-0.006280	-0.011100	0.005758	0.019236	0.054189	0.013077
x1	-0.860857	-0.006280	1.000000	0.399574	-0.188369	-0.254169	-0.211260	-0.142679
x2	-0.686547	-0.011100	0.399574	1.000000	-0.397176	-0.321799	-0.222843	-0.153812
x3	0.063235	0.005758	-0.188369	-0.397176	1.000000	-0.243869	-0.159428	-0.120010
...	...	...	...	...	...	...	...	...
x2996	-0.001075	-0.045933	-0.003245	0.006440	-0.005967	0.006767	-0.019736	0.002063
x2997	-0.007943	-0.041803	0.003291	0.016151	-0.011186	-0.009398	-0.029862	0.002016
x2998	-0.000835	-0.044403	0.008616	-0.000702	-0.010370	0.000155	-0.014945	-0.005242
x2999	-0.007859	-0.057929	0.017245	0.002928	-0.027481	0.001674	-0.011416	-0.004733
x3000	0.018882	0.151034	-0.011025	-0.007538	-0.005811	-0.010401	0.025108	0.000469

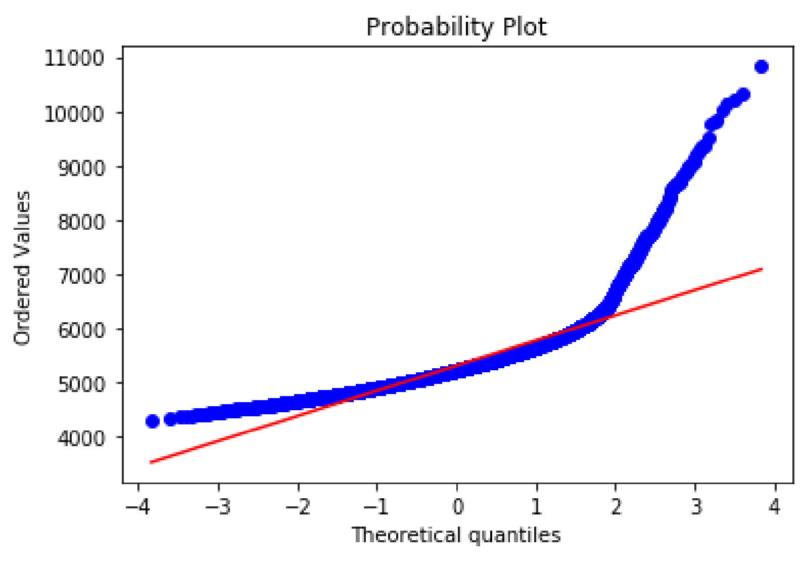
3002 rows × 3002 columns

In [8]:
```
#'z'중심으로 상관관계가 큰 변수 순으로 나열한다.
print("Find most important features relative to z")
corr = df.corr()
corr.sort_values(["z"], ascending = False, inplace=True)
print(corr.z)
```

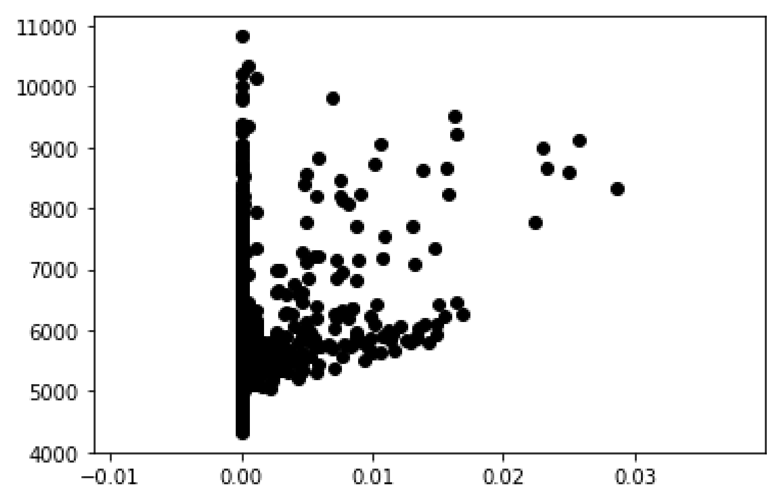
```
Find most important features relative to z
z        1.000000
x993     0.291466
x990     0.282301
x2451    0.275126
x229     0.264278
           ...
x2444   -0.097484
x923    -0.098161
x546    -0.098413
x560    -0.111361
x531    -0.118923
Name: z, Length: 3002, dtype: float64
```

In [13]:
```
#z값들의 분포 확인(정규분포 따르는지)
sns.distplot(df['z'], fit=norm)
fig = plt.figure()
res = stats.probplot(df['z'], plot=plt)
```

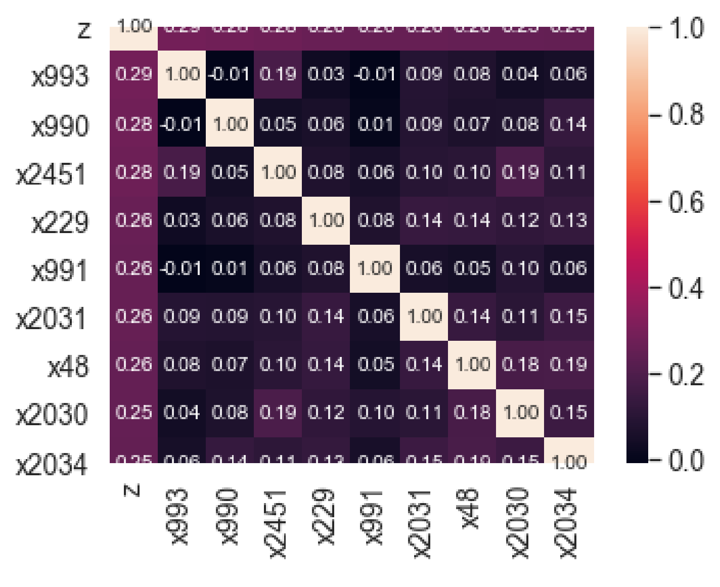




In [14]:
```
plt.scatter(y =df.z, x = df.x993,c = 'black')
plt.show()
```



In [16]:
```
#z값에 영향 주는 10가지 x변수의 상관관계
k = 10
cols = df.corr().nlargest(k, 'z')['z'].index
cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws=
{'size': 10}, yticklabels=cols.values, xticklabels=cols.values)
plt.show()
```



In [33]:
```
#z의 최소값 가질때 행 찾기
z_min=df[df['z']==df['z'].min()]
z_min
```

Out[33]:

	number	status	z	x1	x2	x3	x4	x5	x6	x7	...
6053	23341	optimal	4324.079939	0.000229	0.0	0.0	0.000571	1.320000e-09	0.000114	0.0	...

1 rows × 3003 columns

In [56]:
```
#Min z에 영향 많이 준 x변수의 투자비율 내림차순 정리
z_min.drop(["number","status"],axis=1,inplace=True)
z_min.sort_values(by=6053, ascending=False, axis=1)
```

Out[56]:

	z	x938	x992	x980	x2834	x2719	x2998	x594	x29
6053	4324.079939	0.028577	0.027959	0.022994	0.020798	0.017275	0.01726	0.017161	0.0171

1 rows × 3001 columns

In [72]:
```
#x938의 투자비율이 높으면 z값이 작을지에 대한 확인
df_high_corr=df[["z","x938"]]
df_high_corr.sort_values(by="x938",ascending=False)
```

Out[72]:

	z	x938
8762	4883.537384	0.028577
10112	6050.351064	0.028577
4652	5431.663702	0.028577
1661	5198.550402	0.028577
8392	5306.679310	0.028577
...	...	...
3076	5271.851825	0.000000
3077	5287.904319	0.000000
7305	5147.250482	0.000000
7304	5177.196389	0.000000
5576	4642.547467	0.000000

11152 rows × 2 columns

위에서는 가장 작은 리스크(z)값을 가질때 x938의 투자비율이 높았지만 x938의 투자비율이 높다고 해서 z값이 비례해서 작다는 상관관계는 없음을 확인할 수 있다.

In [86]:
```
#x938의 중복값들 몇개 인지 확인
df_high_corr["x938"].value_counts()
```

Out[86]:
```
0.000000    4433
0.000022     776
0.000023     375
0.000045     242
0.000046     221
            ...
0.000222       2
0.000114       1
0.000135       1
0.000334       1
0.001149       1
Name: x938, Length: 123, dtype: int64
```