In [105]:

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import pandas as pd
import scipy
```

2-1. The error of left and right singular vectors w.r.t. iteration, when eps = 0.01

In [73]:

```
error_col_df = pd.read_csv("error_col.csv")
error_row_df = pd.read_csv("error_row.csv")
```

In [74]:

```
error_col_df.rename({'Unnamed: 0': 'iteration', '0': 'error'}, axis='columns
```

In [75]:

```
error_row_df.rename({'Unnamed: 0': 'iteration', '0': 'error'}, axis='columns
```

In [76]:

```
error_col_df.head(5)
```

Out[76]:

	iteration	error
0	0	0.570087
1	1	0.096296
2	2	0.091610
3	3	0.082515
4	4	0.089372

In [77]:

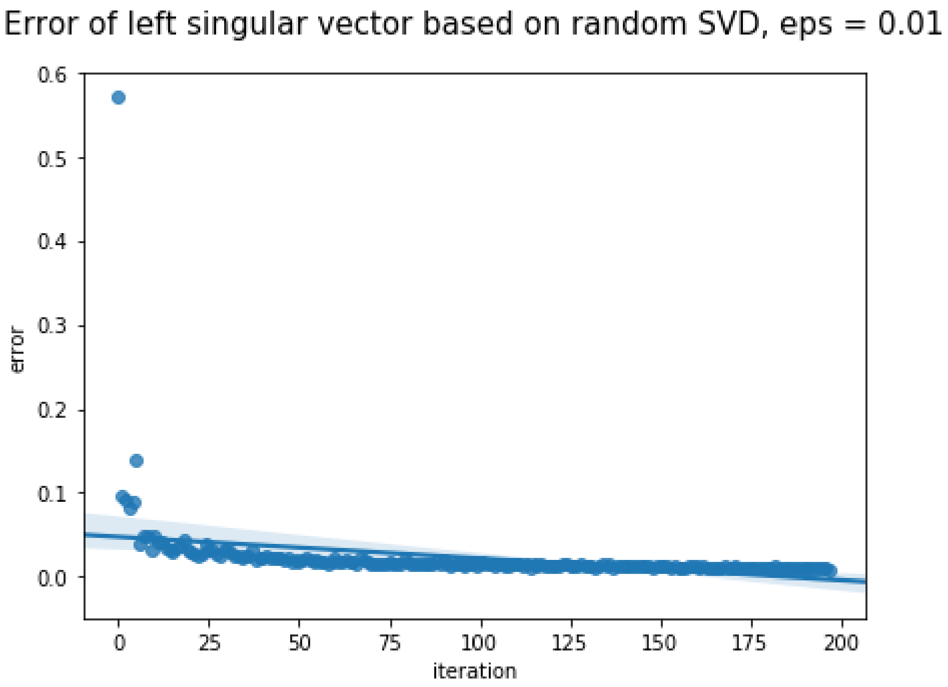
```
error_row_df.head(5)
```

Out[77]:

	iteration	error
0	0	0.291421
1	1	0.225698
2	2	0.102942
3	3	0.078447
4	4	0.043021

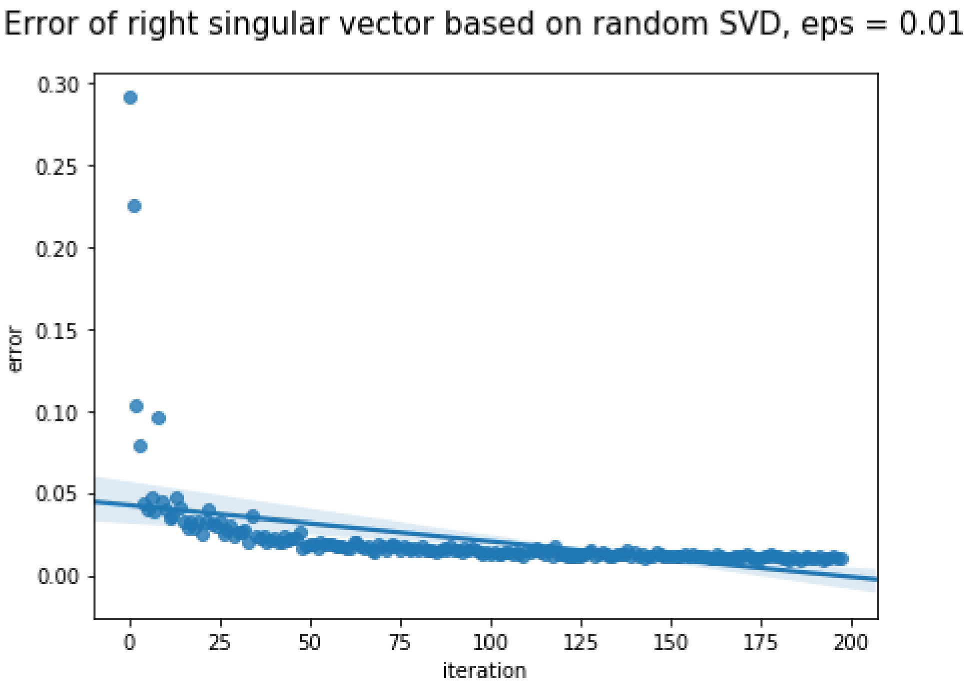
In [120]:
```
dims = (7, 5)
fig, ax = plt.subplots(figsize=dims)
plt.title('Error of left singular vector based on random SVD, eps = 0.01', y
sns.regplot(ax = ax, x=error_col_df["iteration"], y=error_col_df["error"], c
```

Out[120]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a18602710>
```



In [121]:
```
fig, ax = plt.subplots(figsize=dims)
plt.title('Error of right singular vector based on random SVD, eps = 0.01',
sns.regplot(ax = ax, x=error_row_df["iteration"], y=error_row_df["error"], c
```

Out[121]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a18baf160>
```



2-2. The average of c w.r.t. r = [2, 5, 15, 20] when eps = 0.05

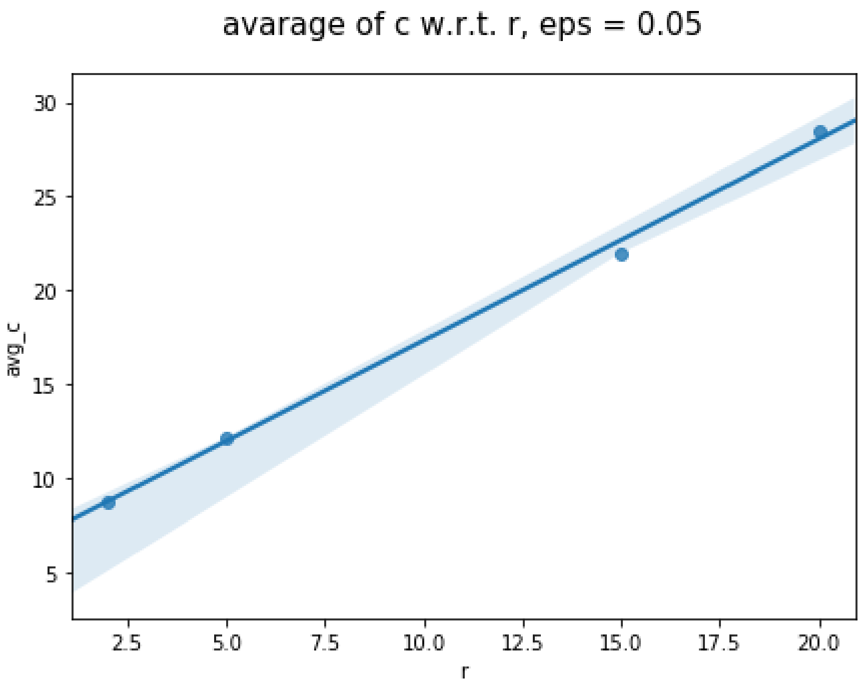
In [122]:
```
cols = ['avg_c']
avg_c = pd.read_csv("avg_c_all.csv",names = cols)
avg_c["r"] = pd.DataFrame([2,5,15,20])
```

In [123]:
```
avg_c
```

Out[123]:

	avg_c	r
0	8.8	2
1	12.2	5
2	22.0	15
3	28.5	20

In [124]:
```
fig, ax = plt.subplots(figsize=dims)
plt.title('avarage of c w.r.t. r, eps = 0.05', y=1.05, size=15)
p = sns.regplot(ax = ax, x=avg_c["r"], y=avg_c["avg_c"], data=avg_c)
```



In [125]:
```
slope, intercept, r_value, p_value, std_err = (
    scipy.stats.linregress(x=p.get_lines()[0].get_xdata(),y=p.get_lines()[0
```

In [126]:
```
print(slope, intercept)
```

```
1.069718309859155 6.642957746478878
```