This example is based on the EDA example in Doing Data Science Ch. 2. There are 31 datasets named nyt1.csv, nyt2.csv,…,nyt31.csv, which you can find here: https://github.com/oreillymedia/doing_data_science (https://github.com/oreillymedia/doing_data_science). I have already downloaded the dataset for you to use under the folder: /nytdata

Each file represents one (simulated) day's worth of ads shown and clicks recorded on the New York Times home page in May 2012. Each row represents a single user. There are five columns: age, gender (0=female, 1=male), number impressions, number clicks, and logged-in.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Lets start off analyzing one file

In [2]:

```
df =  pd.read_csv("dds_ch2_nyt/nyt1.csv") # read-in one file
df.head()
```

Out[2]:

	Age	Gender	Impressions	Clicks	Signed_In
0	36	0	3	0	1
1	73	1	3	0	1
2	30	0	3	0	1
3	49	1	3	0	1
4	47	1	11	0	1

Once you have the data loaded, it's time for some EDA:

Create a new variable, age_group, that categorizes users as "<18", "18-24", "25-34", "35-44", "45-54", "55-64", and "65+".

In [3]:

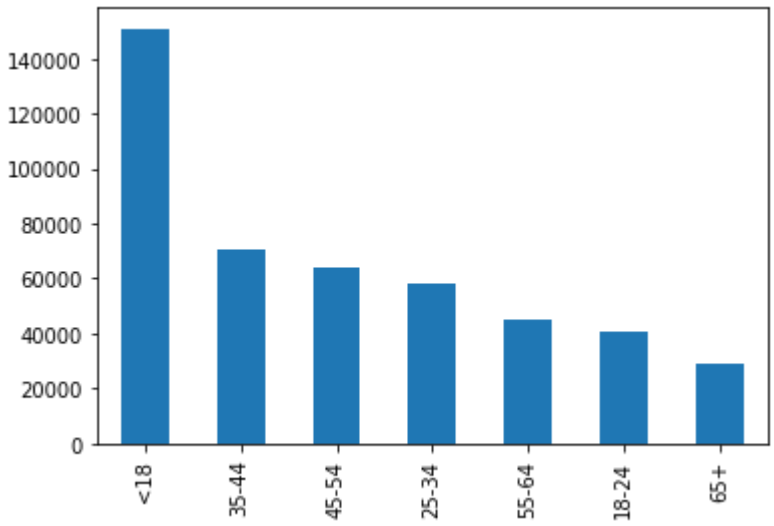
```
bins= [0,18,25,35,45,55,65,108]
labels = ['<18','18-24','25-34','35-44','45-54','55-64','65+']
df['age_group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)
```

In [4]:

```
df['age_group'].value_counts().plot(kind='bar')
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c65bc10748>
```



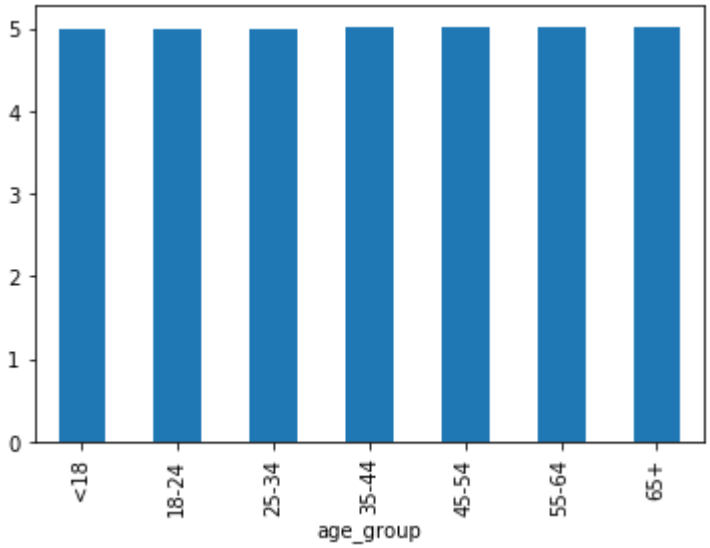
Plot the distributions of number impressions and click-through-rate for these six age categories.

In [5]:

```
df.groupby("age_group")["Impressions"].mean().plot(kind='bar')
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c65eddd208>
```

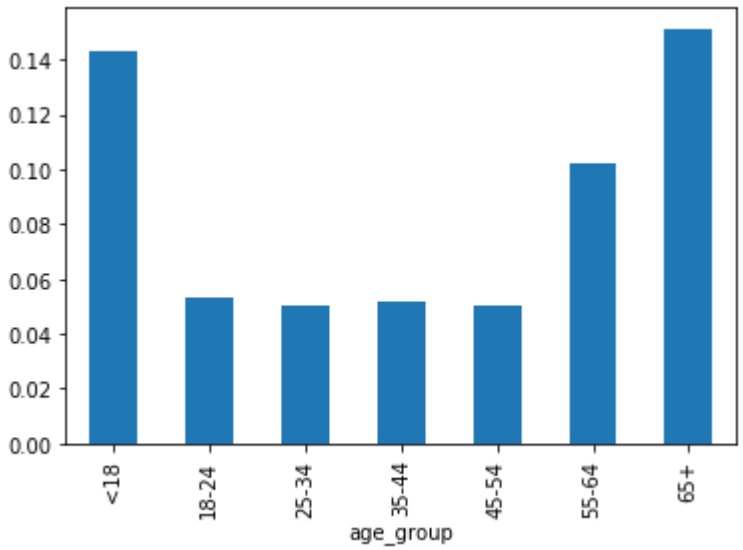


In [6]:

```
df.groupby("age_group").mean()["Clicks"].plot(kind='bar')
```

Out[6]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c65ee4de48>
```



Explore the data and make visual and quantitative comparisons across user segments/demographics (<18-year-old males versus < 18-year-old females or logged-in versus not, for example).

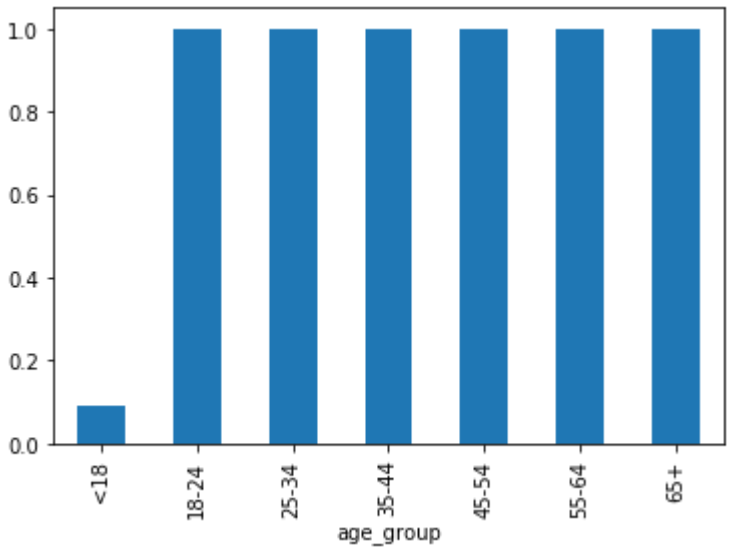
Distribution of Signed_In among age_group.
All ages signed for 100% and only <18 signed for 9%

In [7]:

```
df.groupby('age_group')['Signed_In'].mean().plot.bar()
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c65eedc148>
```



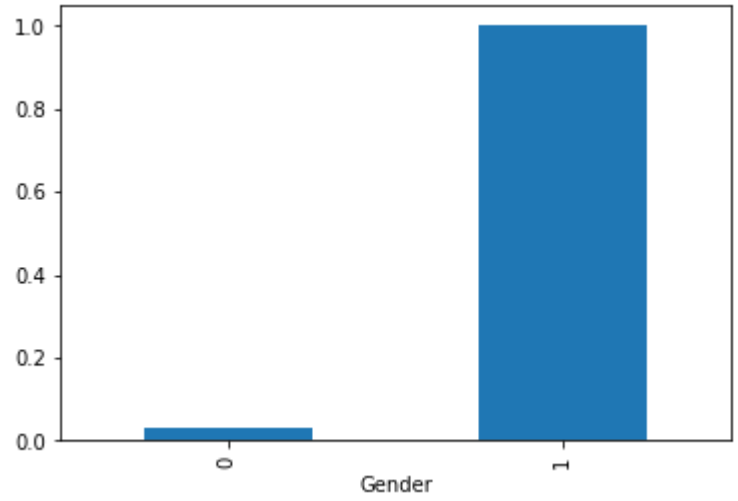
Distribution of Signed_In among Gender for age_group <18
Male signed in for 100%, Female signed in for 3%.
All people who is not signed in belongs to group of Female of age <18. And 97% of Female with age <18 are not signed in.

In [8]:

```
df.pivot_table(index='age_group', columns='Gender', values='Signed_In',
               aggfunc=np.mean).loc['<18'].plot.bar()
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c65ef40ec8>
```



We analyzed just one file, but the dataset includes 31 files (

In [9]:

```
import glob # used to read mutliple-files
files = glob.glob('dds_ch2_nyt/nyt*.csv')
dfs = []
for file in files:
    df = pd.read_csv(file)
    df['filename'] = file[-6:-4]
    dfs.append(df)
df = pd.concat(dfs, ignore_index=True)
```

In [10]:

```
df.head()
```

Out[10]:

	Age	Gender	Impressions	Clicks	Signed_In	filename
0	36	0	3	0	1	t1
1	73	1	3	0	1	t1
2	30	0	3	0	1	t1
3	49	1	3	0	1	t1
4	47	1	11	0	1	t1

Analyze trends over time since we now have a historic view of the data over 31 days.

Drop all lines with Age = 0

In [11]:

```
df_new = df[df['Age'] != 0].reset_index(drop=True)
```

Add new column age_group

In [12]:

```
bins= [0,18,25,35,45,55,65,108]
labels = ['<18','18-24','25-34','35-44','45-54','55-64','65+']
df['age_group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)
```

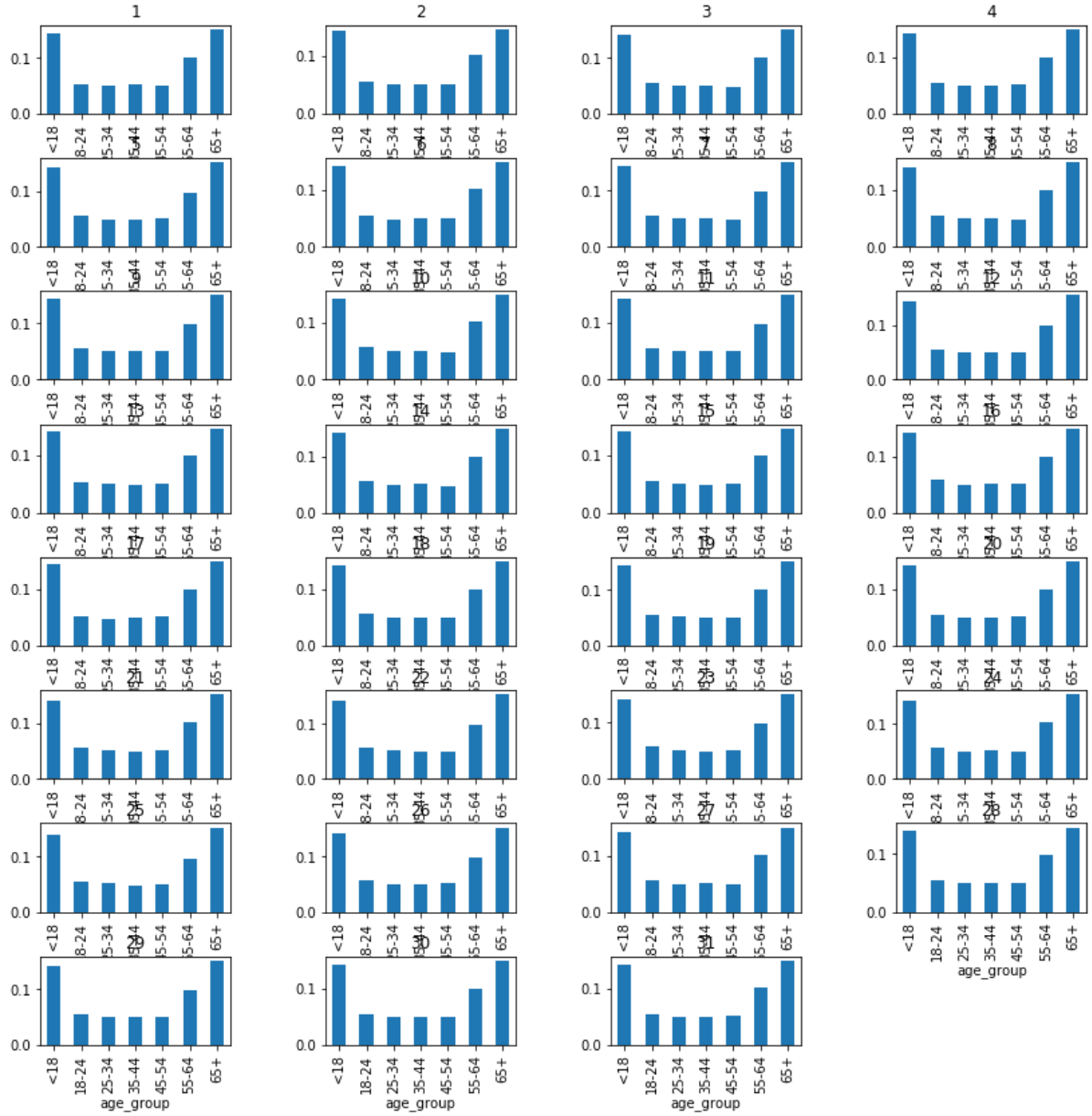
Plot 31 histograms (assuming files in temporary order: 1 is oldest, 31 is newest)
average amount of clicks for each age group, file by file

In [13]:

```
cube = df.pivot_table(index = 'filename', columns = 'age_group',
                          values = 'Clicks', aggfunc = np.mean)
```

In [14]:

```
plt.figure(figsize = (14, 14))
plt.subplots_adjust(hspace = .5, wspace=.5)
for i in range(31):
    plt.subplot(8, 4, i+1)
    plt.title(str(i+1))
    if i < 9:
        cube.loc['t'+str(i+1)].plot.bar()
    else:
        cube.loc[str(i+1)].plot.bar()
```


1
2
3
4
0.0
0.1
<18
18-24
25-34
35-44
45-54
55-64
65+
age_group

In [15]:

```
cube
```

Out[15]:

age_group	<18	18-24	25-34	35-44	45-54	55-64	65+
filename							
10	0.141738	0.056574	0.051094	0.050666	0.047919	0.102387	0.148353
11	0.142153	0.055371	0.049670	0.049380	0.049904	0.098441	0.147955
12	0.143592	0.055351	0.050915	0.049545	0.049670	0.098402	0.153880
13	0.142580	0.054366	0.049996	0.049080	0.051341	0.099238	0.147507
14	0.140956	0.055297	0.048450	0.050696	0.047829	0.099695	0.149640
15	0.142761	0.055169	0.050083	0.048249	0.051136	0.100918	0.147434
16	0.141940	0.057544	0.050185	0.051617	0.051793	0.100063	0.149641
17	0.143263	0.052947	0.048191	0.049861	0.051999	0.099641	0.148668
18	0.142106	0.057218	0.050168	0.050047	0.050299	0.100256	0.148773
19	0.144004	0.054771	0.052331	0.050314	0.049764	0.101086	0.150796
20	0.142254	0.054444	0.049807	0.049257	0.050942	0.099324	0.149088
21	0.141397	0.055737	0.051669	0.048357	0.051901	0.101449	0.152648
22	0.142192	0.056213	0.051803	0.049064	0.048478	0.099595	0.154438
23	0.142334	0.057279	0.051306	0.048940	0.051930	0.098041	0.150719
24	0.142082	0.056174	0.049213	0.051510	0.049995	0.104382	0.154506
25	0.141157	0.054448	0.052398	0.047089	0.050833	0.097127	0.153550
26	0.141400	0.056060	0.049546	0.048835	0.051550	0.098814	0.152539
27	0.142562	0.056573	0.049288	0.050392	0.050328	0.101229	0.150987
28	0.141450	0.054152	0.049784	0.050988	0.050277	0.098467	0.146285
29	0.142944	0.054796	0.049160	0.050767	0.049461	0.098085	0.152145
30	0.141912	0.055050	0.050412	0.049569	0.049424	0.098937	0.150239
31	0.143429	0.055090	0.049857	0.050723	0.050919	0.101744	0.150379
t1	0.142745	0.053251	0.050486	0.051679	0.050274	0.101837	0.151294
t2	0.143053	0.056645	0.050230	0.051019	0.051227	0.101605	0.146404
t3	0.142241	0.056133	0.049663	0.050715	0.048965	0.099940	0.151569
t4	0.142278	0.054681	0.050135	0.050827	0.051295	0.100752	0.149912
t5	0.142581	0.056111	0.049553	0.049613	0.050261	0.097049	0.152668
t6	0.142884	0.054235	0.048985	0.050802	0.049909	0.101388	0.149404
t7	0.143911	0.054774	0.049774	0.050055	0.049352	0.099051	0.150312
t8	0.141422	0.054748	0.049187	0.050245	0.048803	0.100974	0.149303
t9	0.143433	0.055279	0.051100	0.051002	0.049970	0.098752	0.149121

I tried different criterias and all graths look approximately the same over time. See example above

Only noticeble trend over time I found in age group <18, Female, Signed in percentage

In [16]:

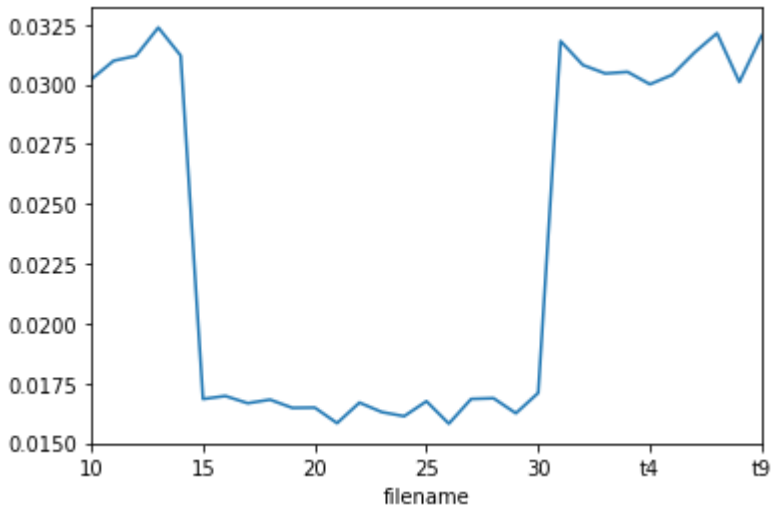
```
cube = df.pivot_table(index = 'filename', columns = ['age_group', 'Gender'],
                      values = 'Signed_In', aggfunc = np.mean)
```

In [17]:

```
cube.xs('<18', level='age_group', axis=1)[0].plot()
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c605a2b7c8>
```



Looks like for the first half of data files, average signed in females of age <18 was about 3%.
For the second half of data it droped to ~1,6%.
The last sample it came back to 3%.