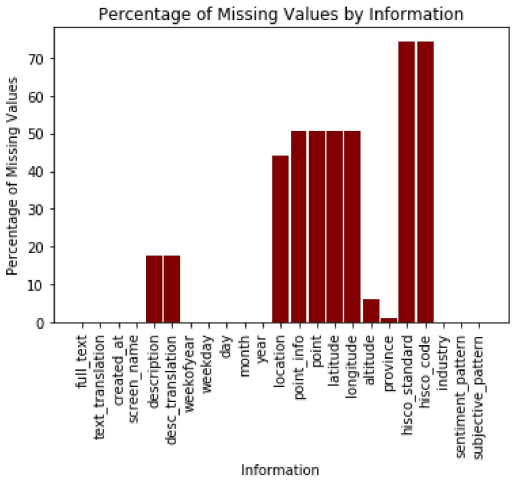
In [1]:
```
import json
import pandas as pd

data_list = [] #I am creating an empty list to store the data from the json files.

for filename in [r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk0.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk1.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk2.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk3.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk4.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk5.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk6.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk7.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk8.json',
                 r'C:\Users\dimop\Desktop\Dutch Social Media\archive (2)\dutch_tweets_chunk9.json']:
    with open(filename) as f:
        data = json.load(f)
        data_list.extend(data)

data = pd.DataFrame(data_list) #Creating a data frame to store all the data
print(data.head())
print(data.shape)
```

```
                                        full_text  \
0  @pflegearzt @Friedelkorn @LAguja44 Pardon, wol...
1  RT @grantshapps: Aviation demand is reduced du...
2  RT @DDStandaard: De droom van D66 wordt werkel...
3  RT @DDStandaard: De droom van D66 wordt werkel...
4  De droom van D66 wordt werkelijkheid: COVID-19...

                                    text_translation     created_at  \
0  @pflegearzt @Friedelkorn @ LAguja44 Pardon wol...  1583756789000
1  RT @grantshapps: Aviation demand is reduced du...  1583756794000
2  RT @DDStandaard: The D66 dream come true: COVI...  1583756797000
3  RT @DDStandaard: The D66 dream come true: COVI...  1583756797000
4  The D66 dream becomes reality: COVID-19 super ...  1583756807000

   screen_name                                        description  \
0  TheoRettich  I ❤️science, therefore a Commie.   ☭ FALGSC: P...
1  davidiwanow  I tweet a lot but love to engage & converse. P...
2     EricL65                                                 None
3     EricL65                                                 None
```

In [72]:
```
import matplotlib.pyplot as plt
missing_values = data.isnull().sum() #I am finding the missing values from my dataframe.
missing_percent = (missing_values / len(data)) * 100 #Counting the missing values and turning them in percentages.
missing_df = pd.DataFrame({'column_name': missing_values.index, 'missing_percent': missing_percent.values}) #I am making a data
print(missing_df)
plt.bar(missing_df['column_name'], missing_df['missing_percent'], color='maroon', width=0.9)
plt.xticks(rotation=90)
plt.xlabel('Information')
plt.ylabel('Percentage of Missing Values')
plt.title('Percentage of Missing Values by Information')
plt.show()
```

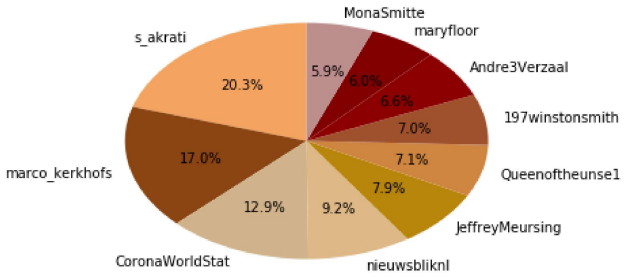
```
          column_name  missing_percent
0           full_text         0.004054
1     text_translation         0.005160
2          created_at         0.002948
3         screen_name         0.003317
4         description        17.577817
5     desc_translation        17.587030
6         weekofyear         0.005160
7             weekday         0.007371
8                 day         0.005160
9               month         0.007371
10               year         0.007371
11           location        44.217629
12         point_info        50.451828
13              point        50.451828
14           latitude        50.451828
15          longitude        50.451828
16           altitude         6.234567
17           province         1.144312
18      hisco_standard        74.393201
19         hisco_code        74.393201
20           industry         0.000000
21   sentiment_pattern         0.000000
22  subjective_pattern         0.000000
```



Explanation: According to the bar graph, the most common missing values from our data are HISCO codes(~74% of the values are missing) and the specific location of the person that makes the tweet(~50% of the values that are related to point, latitude and longtitude are missing).

In [64]:
```
influencers=data['screen_name'].value_counts().head(10) #I am counting the names that appear most in the data set of the tweets
print(influencers)
influencers.plot(kind='pie',autopct='%1.1f%%',startangle=90,colors = ['#F4A460', '#8B4513', '#D2B48C', '#DEB887', '#B8860B', '#
top_influencers = influencers.index.tolist()
```

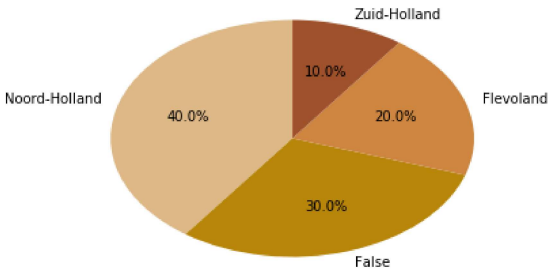
```
s_akrati           1152
marco_kerkhofs      962
CoronaWorldStat     728
nieuwsbliknl        523
JeffreyMeursing     450
Queenoftheunse1     402
197winstonsmith     394
Andre3Verzaal       376
maryfloor           342
MonaSmitte          336
Name: screen_name, dtype: int64
['s_akrati', 'marco_kerkhofs', 'CoronaWorldStat', 'nieuwsbliknl', 'JeffreyMeursing', 'Queenoftheunse1', '197winstonsmith', 'An
dre3Verzaal', 'maryfloor', 'MonaSmitte']
```



Explanation: We can see the top 10 people who tweeted the most in the period of time where we have our data. These are the most active people on Twitter in this particular period and region. s_akrati has created the most tweets ~20% of the tweets among these top-10 were theirs.

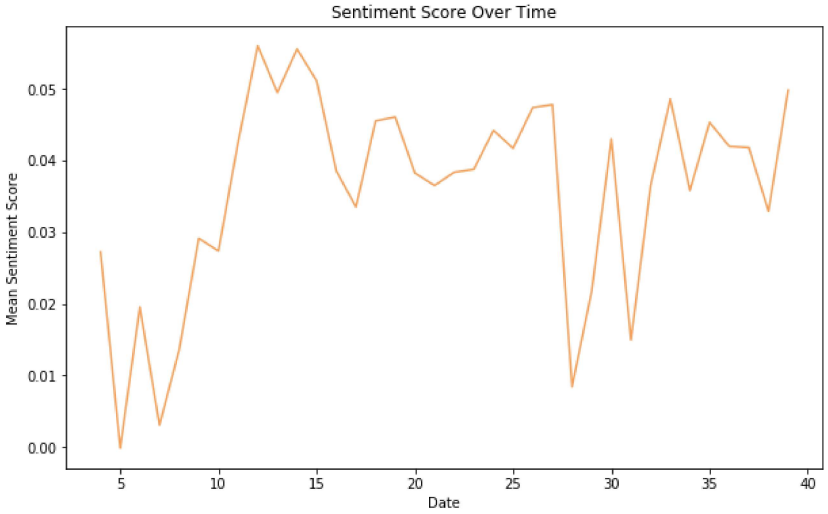
In [80]:
```
a_unique.loc[data_unique['screen_name'].isin(top_influencers), 'province'].value_counts()
kind='pie', autopct='%1.1f%%', startangle=90, colors=['#DEB887', '#B8860B', '#CD853F', '#A0522D', '#8B0000', '#800000', '#BC8F8F
```

Out[80]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x176b9937978>
```



Explanation: It seems that between the top 10 influencers the 40% of them lives in Noord-Holland area and the least of them (around 10%) in the Zuid-Holland. However, there is a big percentage of people who do not show the area they live(30%).

In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

mean_sentiment_by_date = data.groupby('weekofyear')['sentiment_pattern'].mean()
mean_sentiment_by_date.plot(figsize=(10, 6),color='#F4A460')
plt.xlabel('Date')
plt.ylabel('Mean Sentiment Score')
plt.title('Sentiment Score Over Time')
plt.show()
```

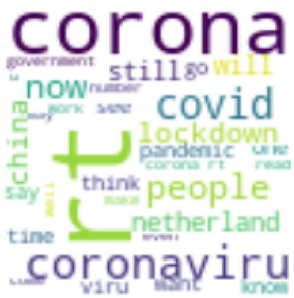


Explanation: This is a linegraph of the sentiment score over time in the region that we are studying. around 5th and 7th week there was a huge drop while around the 15th week there was an increase which means that the sentiment was very positive at this period of time. There was another drop around 28 and 30th week which indicates that the sentiment was mostly negative on the tweets at this period of time.

In [11]:
```
!pip install wordcloud
```

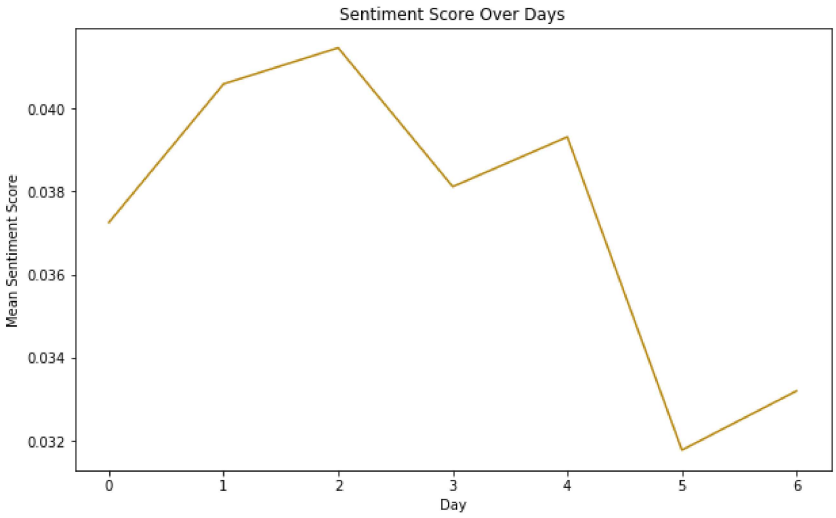
```
Collecting wordcloud
  Downloading https://files.pythonhosted.org/packages/41/bc/fdc10d6a6504db7dbd75077c5df44aebd29d6a439e3bf5ff9f4eb8180b44/wordcloud-1.8.2.2-cp36-cp36m-win_amd64.whl (https://files.pythonhosted.org/packages/41/bc/fdc10d6a6504db7dbd75077c5df44aebd29d6a439e3bf5ff9f4eb8180b44/wordcloud-1.8.2.2-cp36-cp36m-win_amd64.whl) (161kB)
Requirement already satisfied: matplotlib in c:\users\dimop\anaconda3\lib\site-packages (from wordcloud) (2.2.2)
Requirement already satisfied: pillow in c:\users\dimop\anaconda3\lib\site-packages (from wordcloud) (5.1.0)
Requirement already satisfied: numpy>=1.6.1 in c:\users\dimop\anaconda3\lib\site-packages (from wordcloud) (1.19.5)
Requirement already satisfied: cycler>=0.10 in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.2.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.7.3)
Requirement already satisfied: pytz in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2018.4)
Requirement already satisfied: six>=1.10 in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dimop\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.0.1)
Requirement already satisfied: setuptools in c:\users\dimop\anaconda3\lib\site-packages (from kiwisolver>=1.0.1->matplotlib->wordcloud) (39.1.0)
Installing collected packages: wordcloud
Successfully installed wordcloud-1.8.2.2

distributed 1.21.8 requires msgpack, which is not installed.
You are using pip version 10.0.1, however version 21.3.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

In [27]:
```
import re
from wordcloud import WordCloud, STOPWORDS

text = data['text_translation'].str.cat(sep=' ') #We are choosing the column with the translated tweets.

text = text.lower()  # I am converting to lowercase
text = re.sub(r'http\S+', '', text)  # I am removing URLs
text = re.sub(r'@\S+', '', text)  #I am removing mentions
text = re.sub(r'[^\w\s]', '', text)  # I am removing punctuation
text = re.sub(r'\d+', '', text)  #I am removing numbers

wordcloud = WordCloud(width=100, height=100, background_color='white', stopwords=STOPWORDS).generate(text) #Creating the word c

plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```



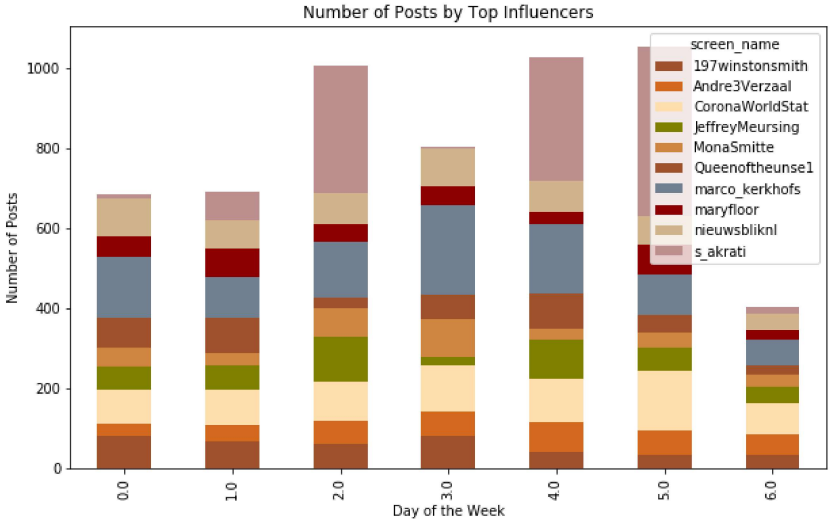
Explanation: Here we have created a word cloud which presents the most commonly used words in the tweets of the people in our data. We can see that the words that are used the most are words related to the pandemic and Covid-19.

In [82]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

mean_sentiment_by_day = data.groupby('weekday')['sentiment_pattern'].mean()
mean_sentiment_by_day.plot(figsize=(10, 6),color='#B8860B')
plt.xlabel('Day')
plt.ylabel('Mean Sentiment Score')
plt.title('Sentiment Score Over Days')
plt.show()
```



Explanation: In the line graph above, we can see the sentiment score according to the week days. We are not sure the the day 0 is Monday. So, we can see that the highest sentiment score, meaning the day that has the most positive posts is the day 2 and the least postive posts are on the 5th day.

In [100]:
```
c[data['screen_name'].isin(top_influencers)]
tweets.groupby(['weekday', 'screen_name'])['text_translation'].count()

ts_by_day.unstack()

ind='bar', stacked=True, figsize=(10,6),color=['sienna','chocolate','navajowhite', 'olive', '#CD853F', '#A0522D','slategrey', '#
he Week')
f Posts')
 Posts by Top Influencers')
```



Explanation: Here we have a stacked bar showing how many tweets per day do the top 10 post. We can see that this top 10 has a pretty steady rhythm of posting each day. Most of them post every day with the 6th day of the week being the one with the least number of posts in total for all the influencers.

In [ ]: