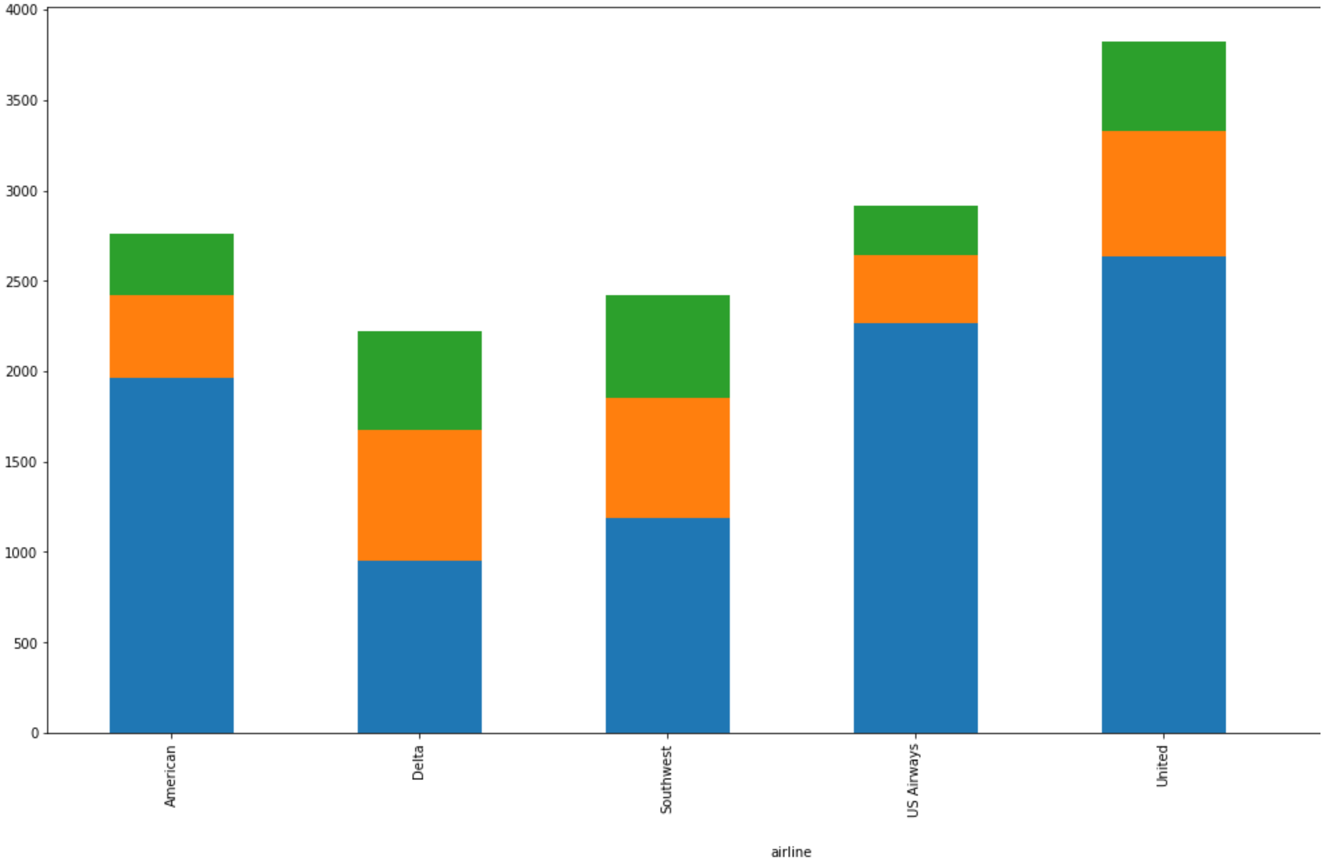
Q1

Code used to Generate the barplot

```
In [3]: import pandas as pd
        import matplotlib.pyplot as plt
        import os
        from wordcloud import WordCloud
        import warnings
        warnings.filterwarnings('ignore')

        df = pd.read_csv("Tweets.csv")

        df2 = df.groupby(['airline', 'airline_sentiment'])['airline'].count().unstack('airline_sentiment').fillna(0)
        df2.plot(kind='bar', stacked=True, figsize=[20,10])
```

```
Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x2c99740ad68>
```



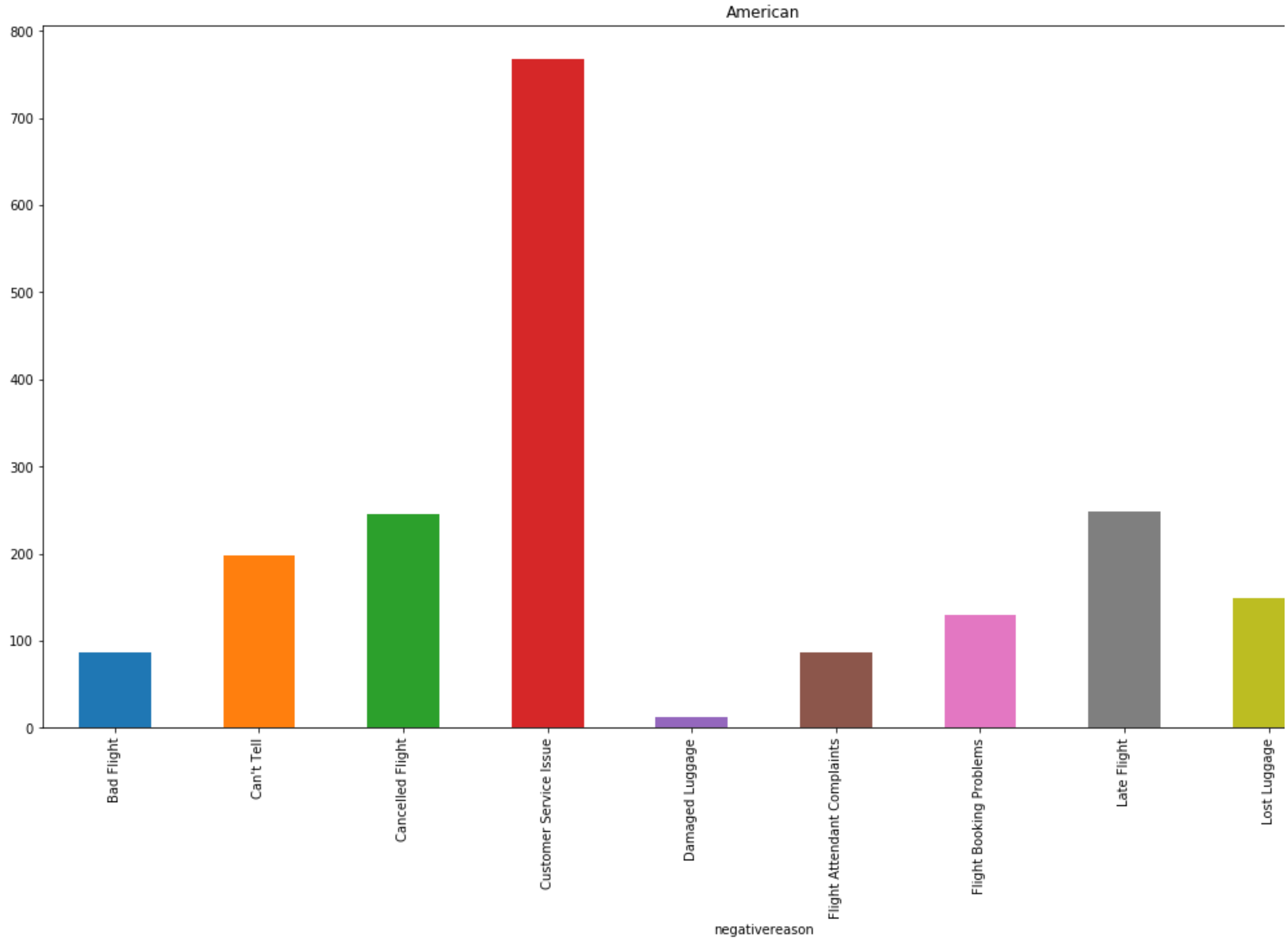
In this barchart we can look the sentiments of reviews expressed by the passengers for 6 different airline carriers, on the Y-AXIS tweets and on the X-AXIS we have the various airline carriers. Each bar graph for airline is broken down into 'Positive', 'Negative tweets.

Q2

Code used to generate the plot for studying various reasons of negative feedback.

```
In [4]: import plotly.plotly as py
        import plotly.tools as tls

        import matplotlib.pyplot as plt

        df2 = df.groupby([ 'negativereason','airline'])['negativereason'].count().unstack('negativereason').fillna(0)
        df2.iloc[0].plot(kind='bar', figsize=[20,10], subplots=True)
```

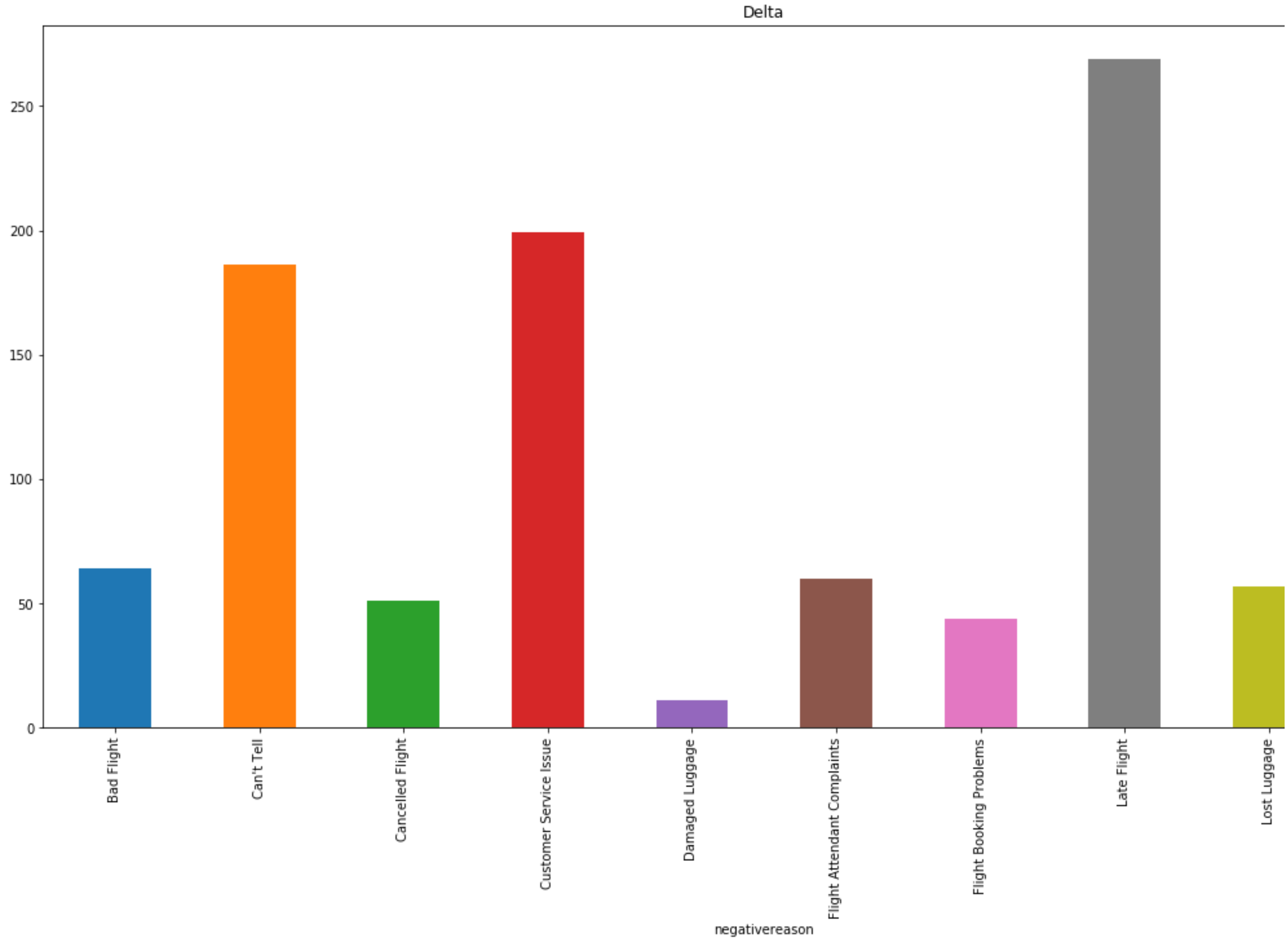
```
Out[4]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002C999AA1320>],
              dtype=object)
```



Here we can see the distribution of feedback for negative reasons, for American Airlines, from the graph it looks the airline neec customer service department as it is biggest cause of concern for passengers.

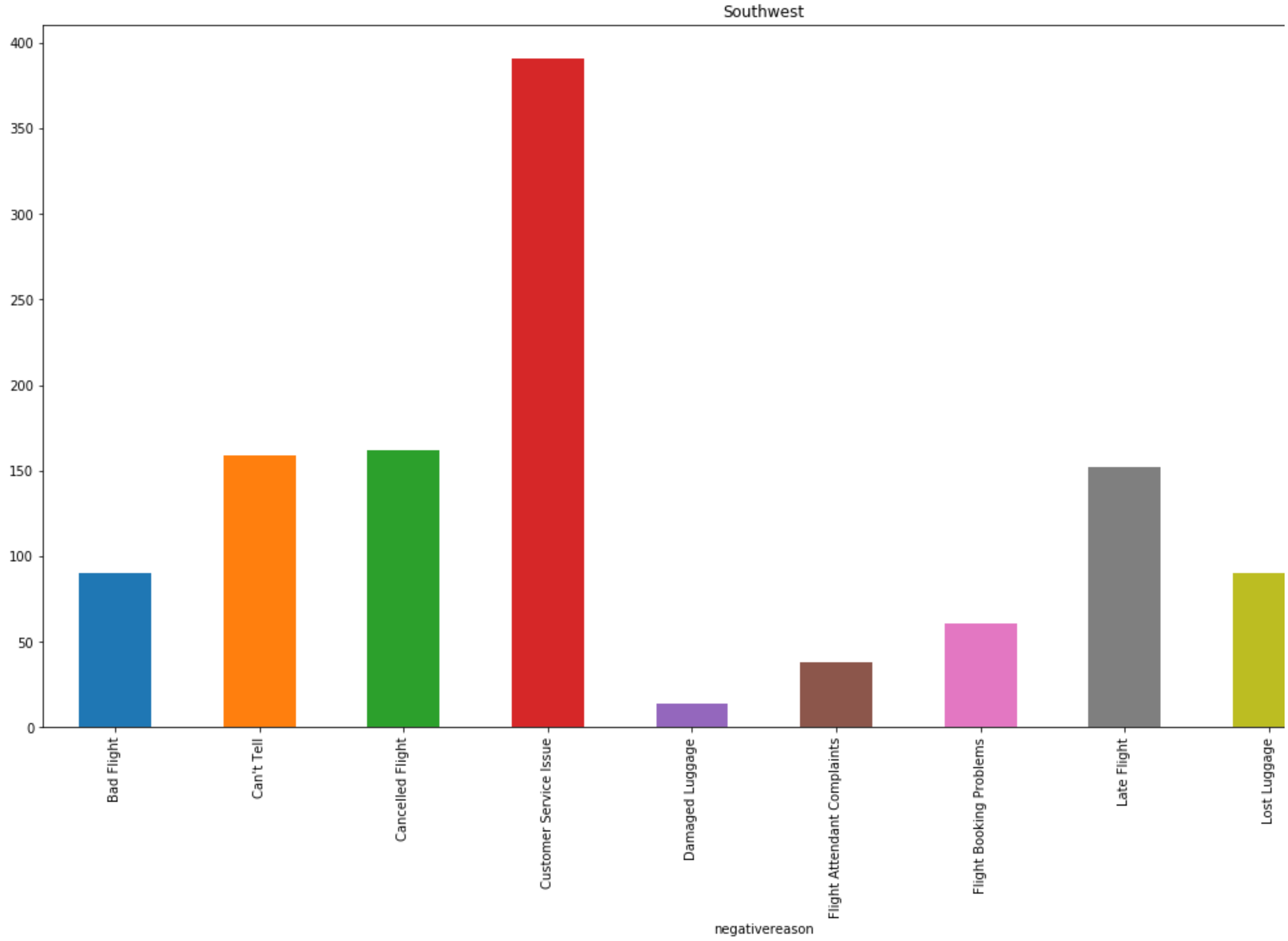
```
In [82]: #Plot for Delta Airlines
         df2.iloc[1].plot(kind='bar', figsize=[20,10], subplots=True)
```

```
Out[82]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x00000271593C0320>],
               dtype=object)
```



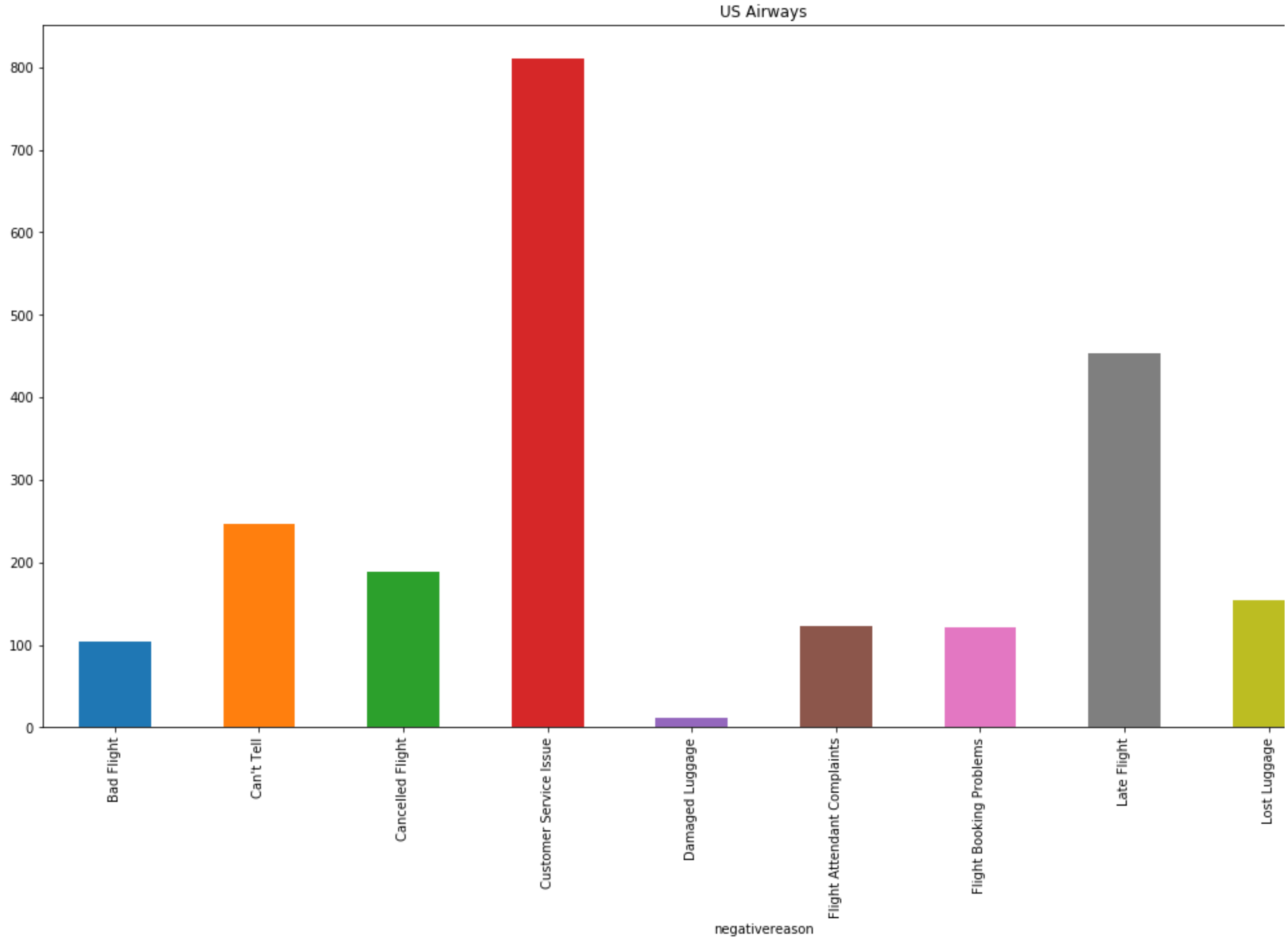
Most common cause of dissatisfaction for Delta: Late flights

```
In [83]: ##Plot for Southwest Airlines
         df2.iloc[2].plot(kind='bar', figsize=[20,10], subplots=True)

Out[83]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x0000027162318160>],
               dtype=object)
```



Most common cause of dissatisfaction for Southwest: Customer service

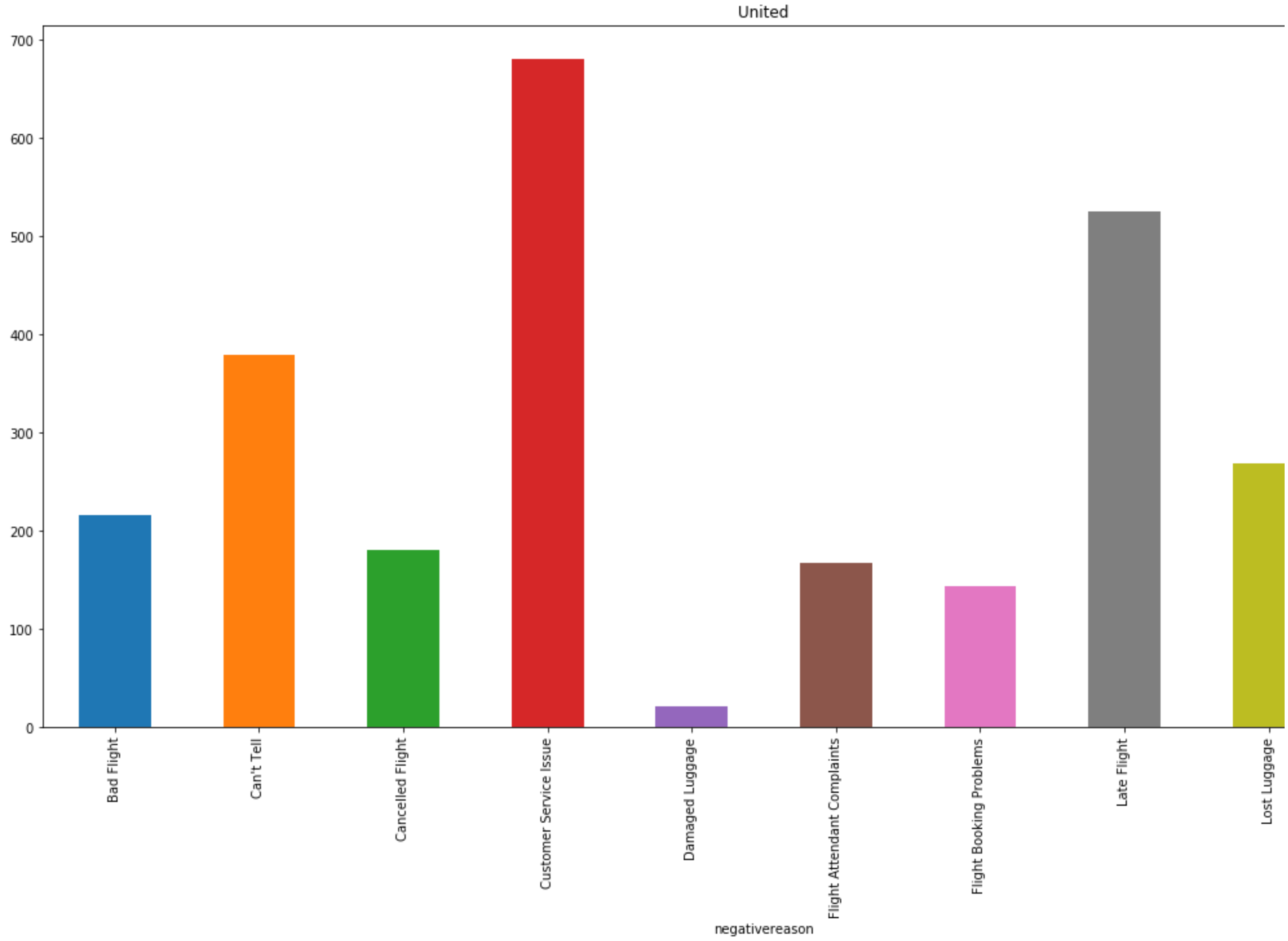
```
In [64]: #Plot for US Airways
         df2.iloc[3].plot(kind='bar', figsize=[20,10], subplots=True)

Out[64]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002837DDCBE48>],
               dtype=object)
```



Most common cause of dissatisfaction for US aiways: Customer service

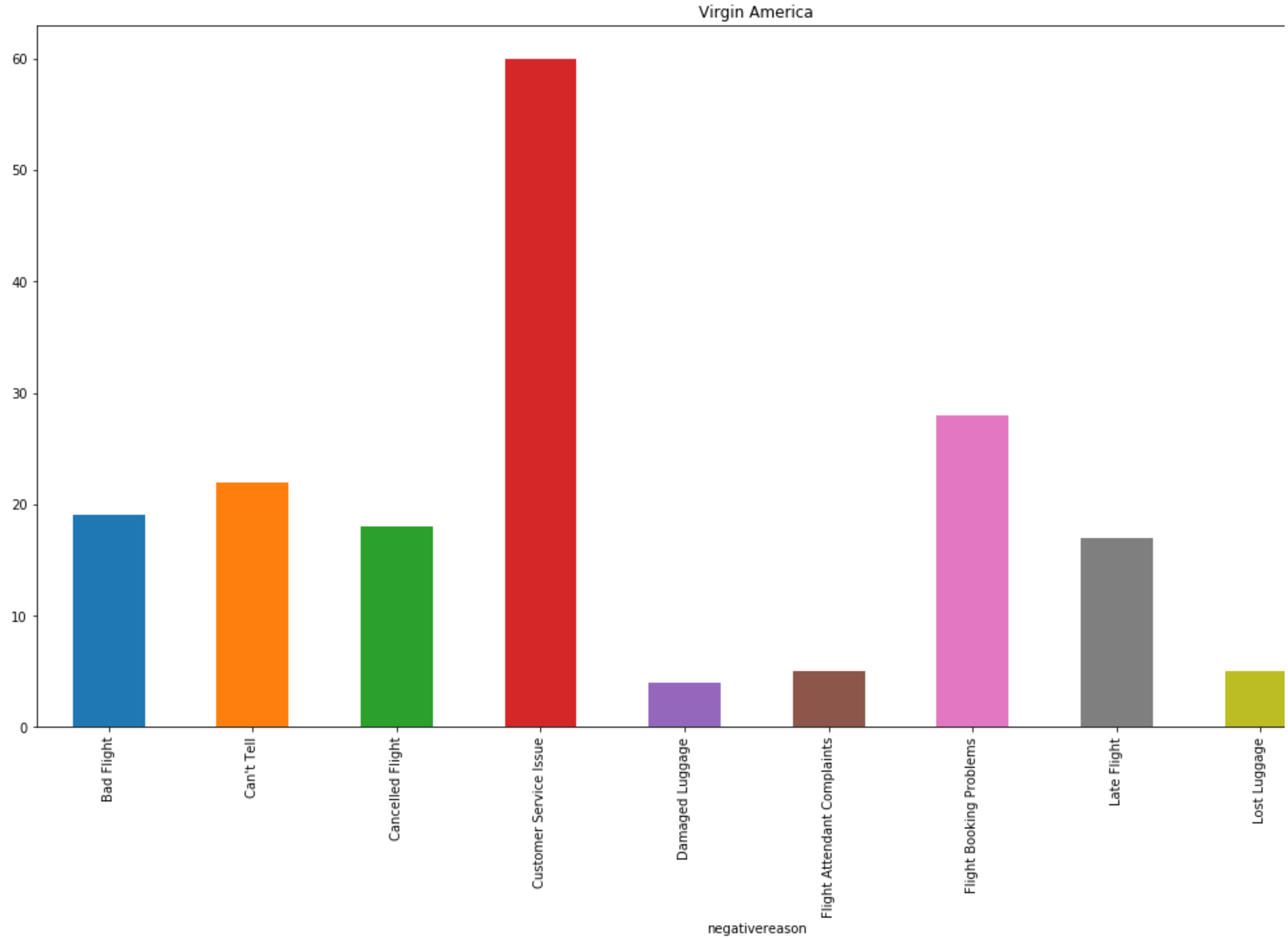
```
In [65]: #Plot for United airlines
         df2.iloc[4].plot(kind='bar', figsize=[20,10], subplots=True)
```

```
Out[65]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002837DF2B550>],
               dtype=object)
```



Most common cause of dissatisfaction for United: Customer service

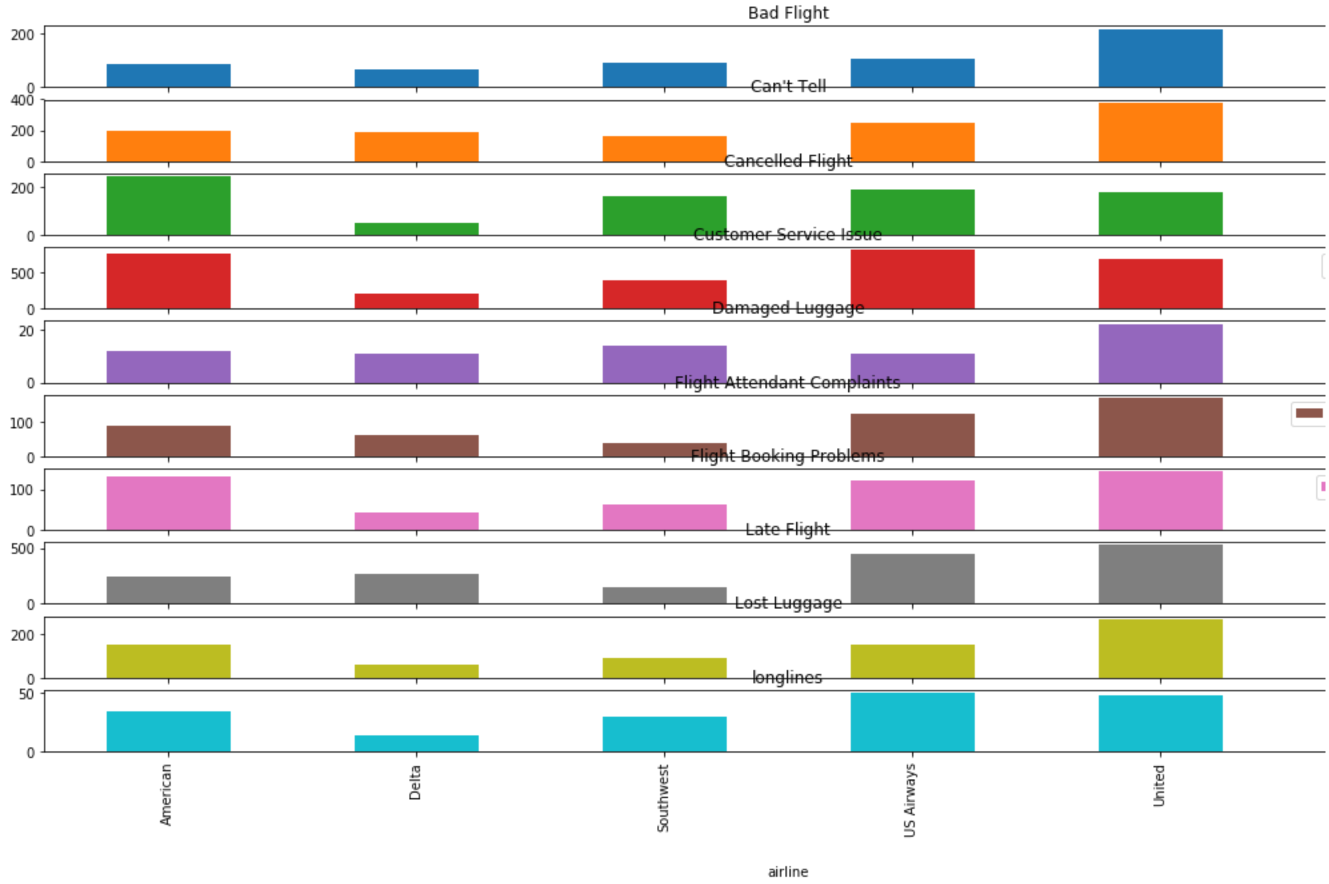
```
In [66]: #Plot for Virgin american airlines
         df2.iloc[5].plot(kind='bar', figsize=[20,10], subplots=True)

Out[66]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002837DFACBA8>],
               dtype=object)
```



Most common cause of dissatisfaction for Virgin: Customer service

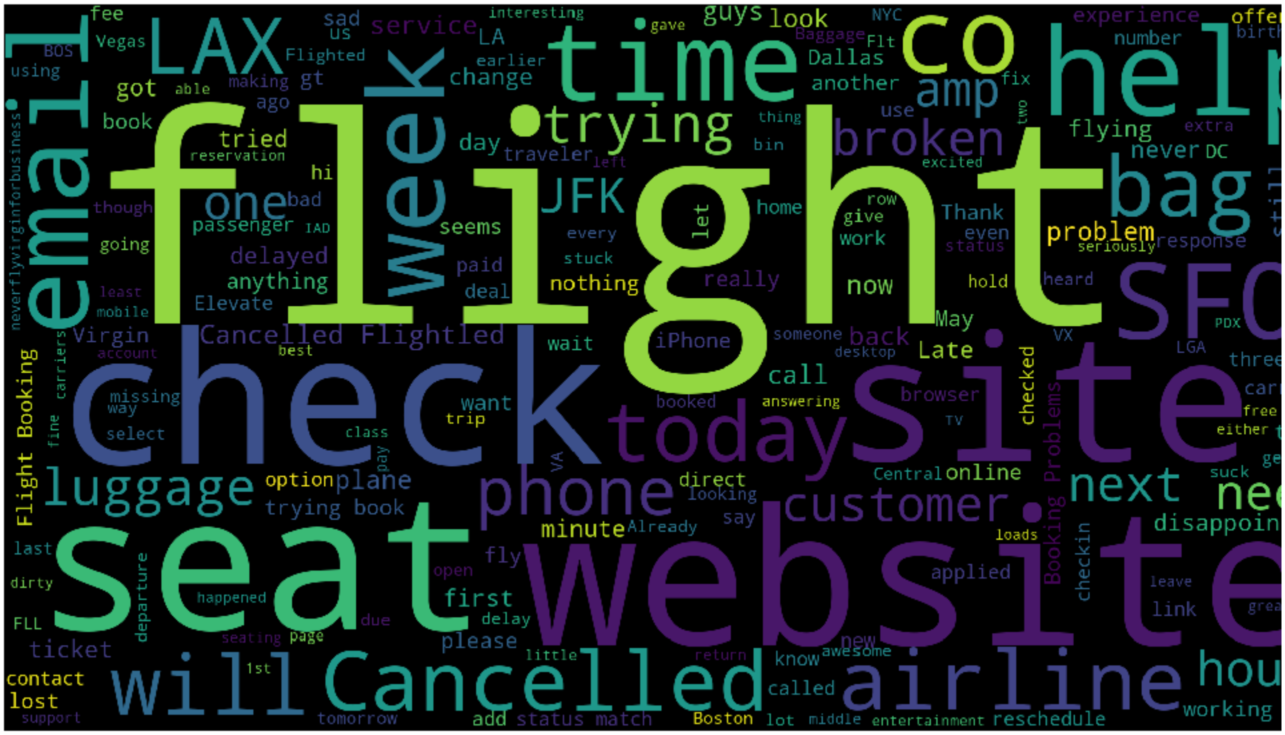
```
In [68]: #All airlines together
         df2.plot(kind='bar', figsize=[20,10], subplots=True)
```

```
Out[68]: array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002837E024B00>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E062128>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E086518>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E0AFA90>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E77F048>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E77F080>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837E7CFB38>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837EED20F0>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837EEFA668>,
                <matplotlib.axes._subplots.AxesSubplot object at 0x000002837EF22BE0>],
               dtype=object)
```



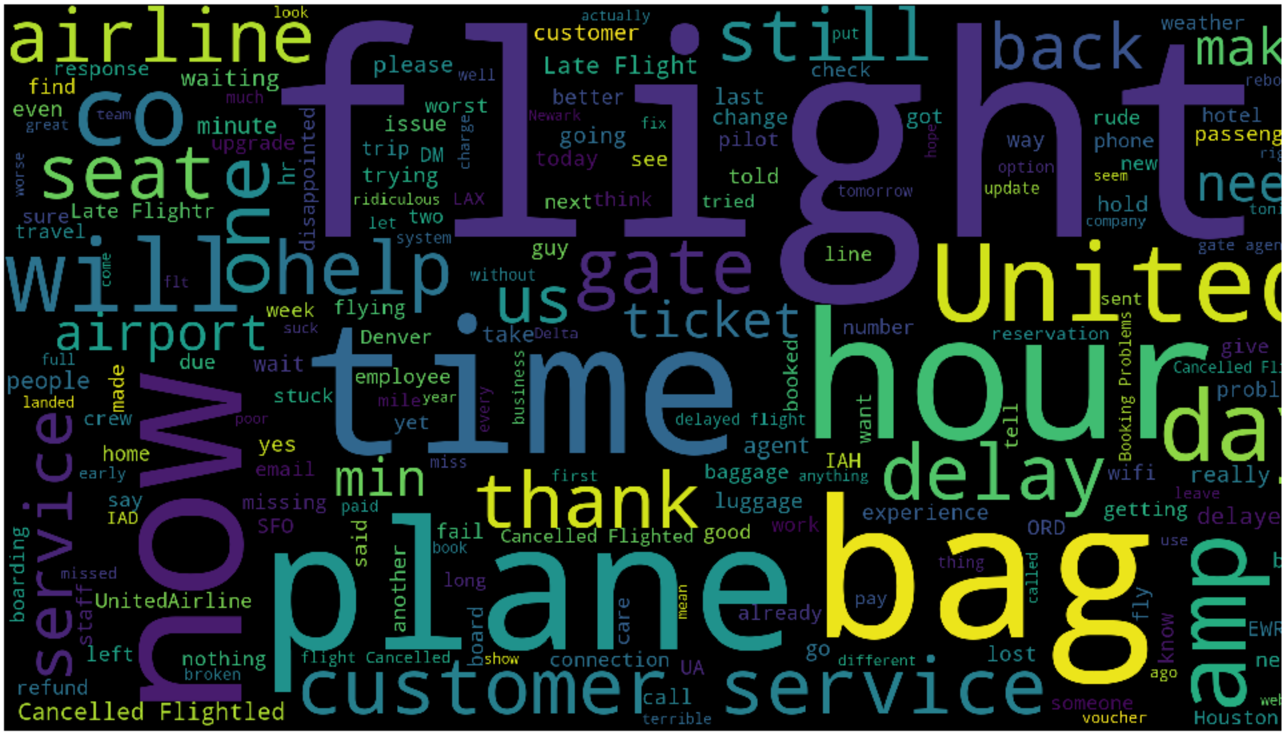
This chart shows the distribution of various reasons people left negative review for each airline, on the X-AXIS we have airline carriers, Y-AXIS gives us the

In [28]:
```
text = open('Virgin_neg.txt','r', encoding="UTF8").read()

# Generate a word cloud object and plot it on the x and y axis
wordcloud = WordCloud(width=1980,height=1080).generate(text)

plt.figure( figsize=(20,10) )
plt.imshow(wordcloud)

#Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
plt.axis("off")

#Show the word cloud
plt.show()
```



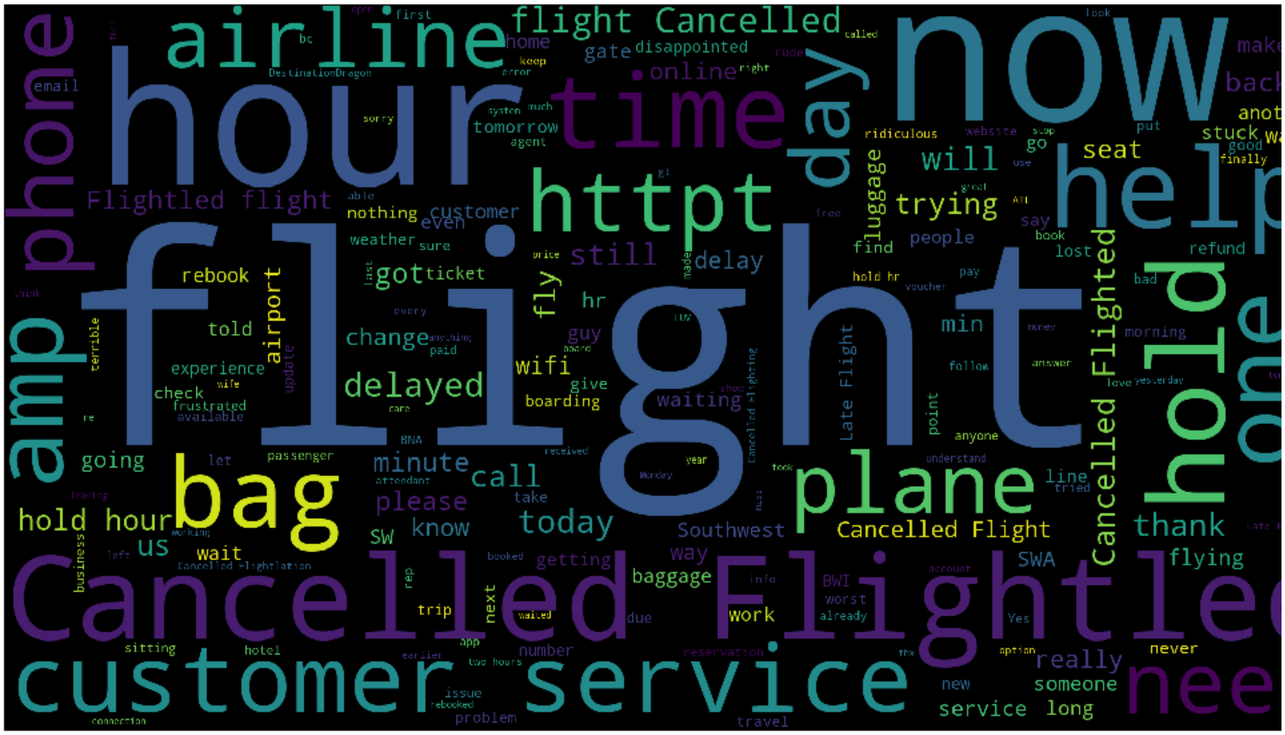
Above is worcloud for American airlines: They seem to have issues with website and seat allotment

```
In [30]: text_united = open('United_neg.txt','r', encoding="UTF8").read()

         # Generate a word cloud object and plot it on the x and y axis
         wordcloud = WordCloud(width=1980,height=1080).generate(text_united)

         plt.figure( figsize=(20,10) )
         plt.imshow(wordcloud)

         #Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
         plt.axis("off")

         #Show the word cloud
         plt.show()
```



Above is wordcloud for United: They seem to have issues with delys, missing bags and customer service

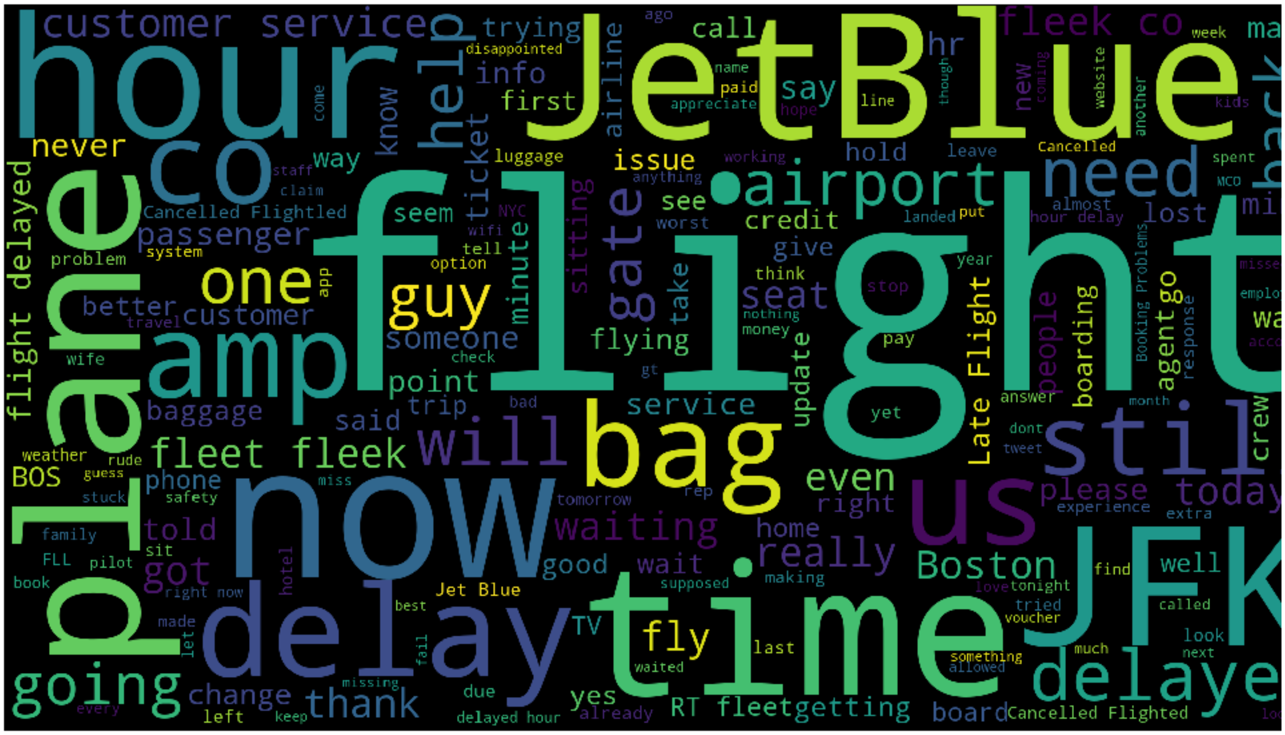
```
In [31]: text_southern = open('Southern_air.txt','r', encoding="UTF8").read()

         # Generate a word cloud object and plot it on the x and y axis
         wordcloud = WordCloud(width=1980,height=1080).generate(text_southern)

         plt.figure( figsize=(20,10) )
         plt.imshow(wordcloud)

         #Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
         plt.axis("off")

         #Show the word cloud
         plt.show()
```



Above is wordcloud for Southern_air: They seem to have issues with cancelled flights

In [32]:

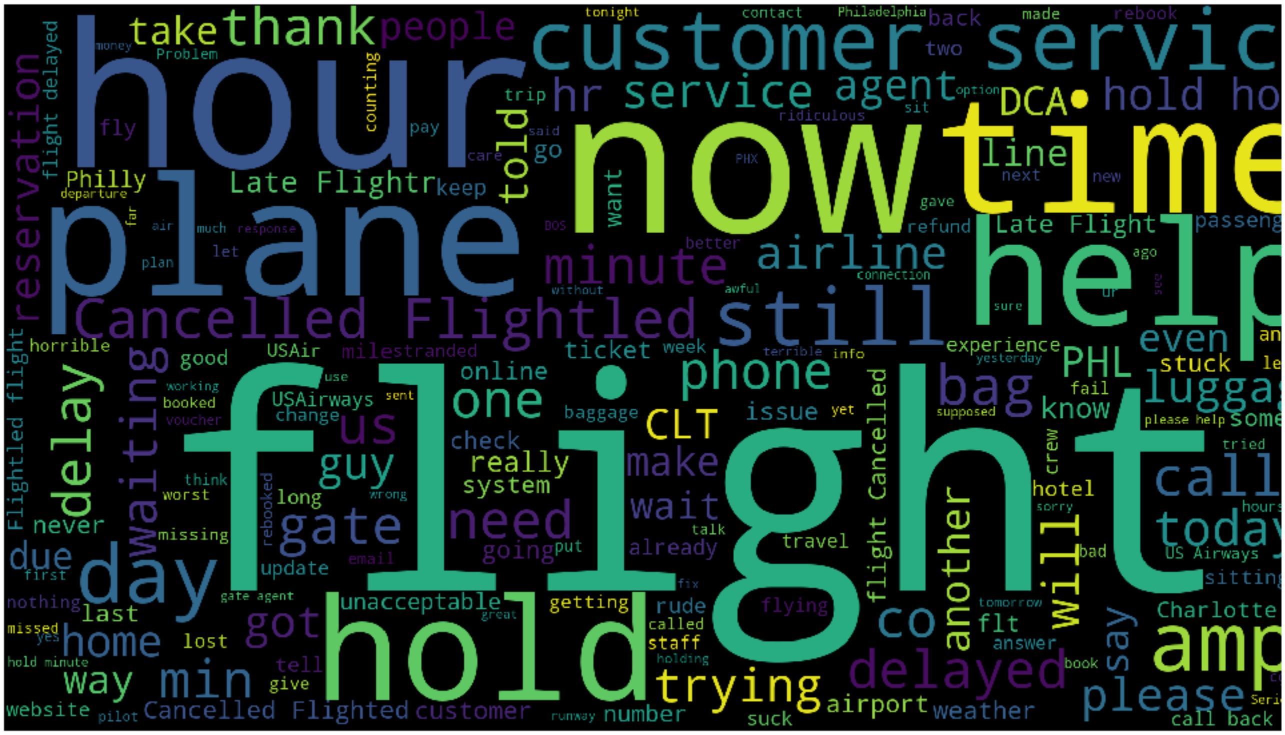
```
text_jet = open('Jet_blue.txt','r', encoding="UTF8").read()

# Generate a word cloud object and plot it on the x and y axis
wordcloud = WordCloud(width=1980,height=1080).generate(text_jet)

plt.figure( figsize=(20,10) )
plt.imshow(wordcloud)

#Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
plt.axis("off")

#Show the word cloud
plt.show()
```



Above is wordcloud for Jetblue: They seem to have issues with delayed flights

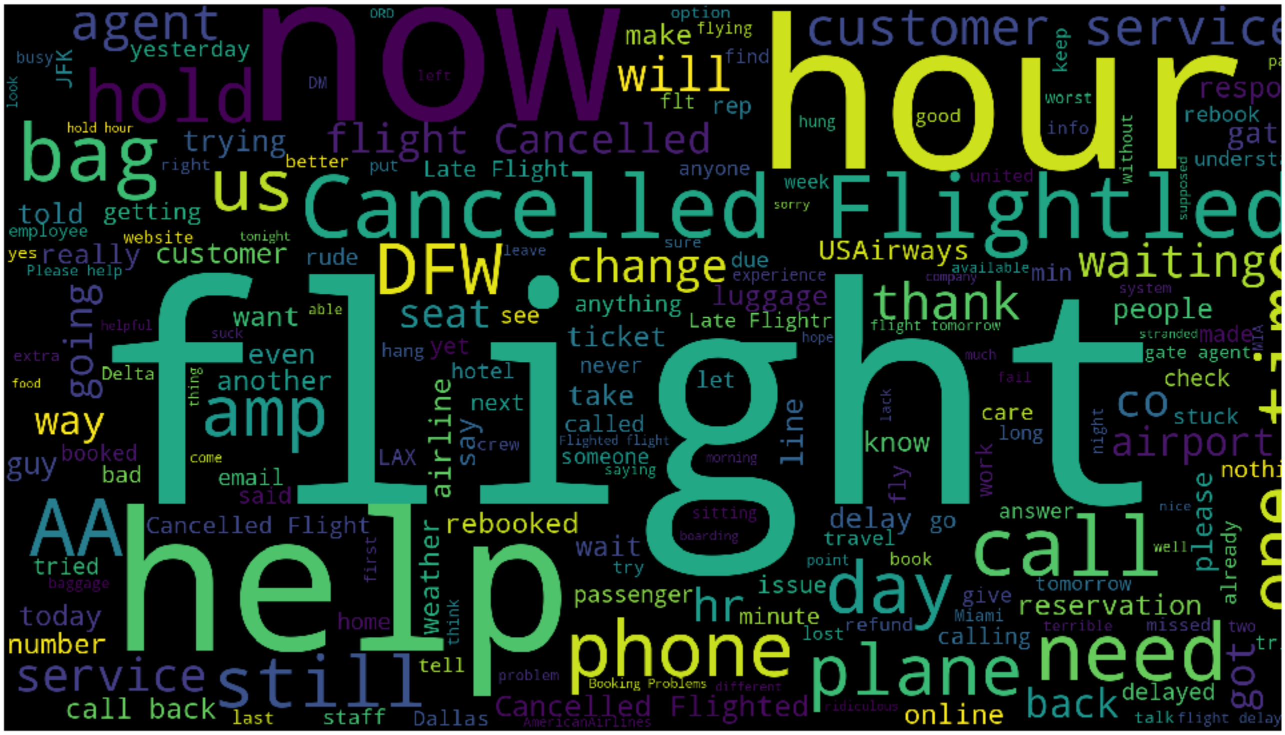
In [33]:

```
text_us = open('Us_air.txt','r', encoding="UTF8").read()

# Generate a word cloud object and plot it on the x and y axis
wordcloud = WordCloud(width=1980,height=1080).generate(text_us)

plt.figure( figsize=(20,10) )
plt.imshow(wordcloud)

#Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
plt.axis("off")

#Show the word cloud
plt.show()
```



Above is wordcloud for US airways: They seem to have issues with customer service

In [34]:
```
text_american = open('American_can.txt','r', encoding="UTF8").read()

# Generate a word cloud object and plot it on the x and y axis
wordcloud = WordCloud(width=1980,height=1080).generate(text_american)

plt.figure( figsize=(20,10) )
plt.imshow(wordcloud)

#Turn off the axis. Otherwise you will see a bunch of extra numbers around the word cloud
plt.axis("off")

#Show the word cloud
plt.show()
```



In [ ]:
```
### Above is wordcloud for American airlines: They seem to have issues with cancelled flights
```

In [7]:
```
#LDA

import spacy
spacy.load('en')
from spacy.lang.en import English
parser = English()
def tokenize(text):
    lda_tokens = []
    tokens = parser(text)
    for token in tokens:
        if token.orth_.isspace():
            continue
        elif token.like_url:
            lda_tokens.append('URL')
        elif token.orth_.startswith('@'):
            lda_tokens.append('SCREEN_NAME')
        else:
            lda_tokens.append(token.lower_)
    return lda_tokens
```

In [8]:

```
import nltk
nltk.download('wordnet')
from nltk.corpus import wordnet as wn
def get_lemma(word):
    lemma = wn.morphy(word)
    if lemma is None:
        return word
    else:
        return lemma

from nltk.stem.wordnet import WordNetLemmatizer
def get_lemma2(word):
    return WordNetLemmatizer().lemmatize(word)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Lameware\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

In [9]:

```
nltk.download('stopwords')
en_stop = set(nltk.corpus.stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Lameware\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [10]:

```
def prepare_text_for_lda(text):
    tokens = tokenize(text)
    tokens = [token for token in tokens if len(token) > 4]
    tokens = [token for token in tokens if token not in en_stop]
    tokens = [get_lemma(token) for token in tokens]
    return tokens
```

In [11]:

```
import random
import pandas as pd
text_data = []
df = pd.read_csv("Tweets.csv")
for line in df.text:
    tokens = prepare_text_for_lda(line)
    if random.random() > .99:
        print(tokens)
    text_data.append(tokens)
```

```
['SCREEN_NAME', 'pretty', 'graphics', 'better', 'minimal', 'iconography']
['.@virginamerica', 'great', 'return', 'call', 'could', 'myvxexperience']
['SCREEN_NAME', 'flying']
['SCREEN_NAME', 'complete', 'doubt', 'vacation', 'responsibility']
['SCREEN_NAME', 'plane', 'delay', 'baggage', 'loading', 'issue']
['SCREEN_NAME', 'thank', 'follow', 'otherwise']
['SCREEN_NAME', 'assign', 'without', 'pay']
['SCREEN_NAME', 'miss', 'flight', 'flight', 'minutes', 'would', 'watch']
['SCREEN_NAME', 'customer', 'service']
['SCREEN_NAME', 'years', 'lose', 'customer']
['SCREEN_NAME', 'frustrate', 'flying', 'experience', 'continuous', 'dunno', 'leave', 'today', 'dunno']
['SCREEN_NAME', 'seriously', 'delay', 'copilot', 'hours', 'flying', 'raleigh', 'disappoint']
['SCREEN_NAME', 'calling', 'tomorrow']
['SCREEN_NAME', 'direct', 'message', 'follow', 'luggage', 'please']
['SCREEN_NAME', 'incredibly', 'unfriendly', 'ground', 'staff', 'false', 'promise', 'twitter', 'SCREEN_NAME']
['SCREEN_NAME', 'skilled', 'staff', 'inadequate', 'system', 'delete', 'return', 'houston', 'norway', 'shoulder', 'shrug']
['SCREEN_NAME', 'would', 'think', 'would', 'clean', 'ramp', 'stick', 'newark', 'would', 'never', 'happen', 'SCREEN_NAME']
['SCREEN_NAME', 'compensation', 'upgrade', 'higher', 'class', 'reschedule', 'flight', 'today', 'appreciate', 'though']
['SCREEN_NAME', 'ready', 'agent', 'going']
['SCREEN_NAME', 'flight', 'seats', 'morning']
['SCREEN_NAME', 'flight', 'delay', 'another', 'transit', 'extremely', 'unlikely']
['SCREEN_NAME', 'tonight', 'flintstone', 'happy', 'upgrade', 'class']
['SCREEN_NAME', 'sitting', 'runway', 'hours', 'ridiculous']
['SCREEN_NAME', 'great', 'lose', 'direct', 'flight', 'manage']
['SCREEN_NAME', 'flight', 'cancel', 'flightled', 'mechanical', 'issue', 'weather']
['SCREEN_NAME', 'charge', 'people', 'freak', 'flight']
['SCREEN_NAME', 'unbelievably', 'disappoint', 'delay', 'delay', 'mess']
['SCREEN_NAME', 'digging', 'mileage', 'earn', 'would', 'mile', 'change', 'award', 'ticket', 'price']
['SCREEN_NAME', 'surprise', 'never', 'experience', 'anything', 'number']
['SCREEN_NAME', 'details', 'flight', 'recording', 'repeat', 'person']
['SCREEN_NAME', 'cancel', 'flightled', '5102newflight', 'upset', 'cancel', 'flight', 'could', 'rent', 'would', 'honest']
['SCREEN_NAME', 'spite', 'flight', 'delay', 'great', 'customer', 'service', 'provide', 'janet', 'baggage', 'employee', 'karen', 'w
['SCREEN_NAME', 'agent', 'clifton', 'father', 'dying', 'need', 'handbag', 'cabin', 'nothing', 'shameful']
['SCREEN_NAME', 'thanks', 'form', 'submit', 'bet', 'someone', 'would', 'honest', 'enough', 'though']
['SCREEN_NAME', 'partner', 'book', 'individual', 'reservation', 'flight', 'silver', 'share', 'premier', 'privilege', 'somehow']
['SCREEN_NAME', 'still', 'sitting', 'waiting', 'plane', 'hours', 'waste']
['SCREEN_NAME', 'airline', 'person', 'working', 'special', 'services']
['SCREEN_NAME', 'thank', 'address']
['SCREEN_NAME', 'SCREEN_NAME', 'really', 'crash', 'sayin']
['SCREEN_NAME', 'happen', 'unite', 'pilot', 'always', 'america', 'treat', 'customer', 'better', 'employee']
['SCREEN_NAME', 'frustrate', 'cancel', 'flight', 'flight', 'overflow', 'toilet', 'plumber']
['SCREEN_NAME', 'flight', 'promo', 'include', 'brian', 'williams', 'tough', 'quickly', 'complaint', 'observation']
['SCREEN_NAME', 'board', 'ua1297', 'refuse', 'carry', 'overhead', 'space', 'plane', 'loads', 'space', 'customerexperience']
['SCREEN_NAME', 'le2v9d', 'number', 'flight', 'first', 'cancel', 'flightled', 'weather', 'iatanbul', 'reflight', 'booking', 'prob
['SCREEN_NAME', 'useless', 'response', 'functioning', 'wheelchair', 'customer']
['SCREEN_NAME', 'flight', 'everyone', 'brave', 'weather', 'passenger', 'waiting', 'flight', 'really']
['SCREEN_NAME', 'frustrate', 'looooong', 'speak', 'person', 'boarding', 'flight', 'tomorrow']
['SCREEN_NAME', 'airport', 'removal', 'method', 'folks', 'cessna', 'become']
['SCREEN_NAME', 'become', 'every', 'airline', 'crook', 'believe', 'taking', 'shouldwearmasks', 'shock']
['SCREEN_NAME', 'ticket', 'book', 'arrival', 'reflect']
['SCREEN_NAME', 'agent', 'wrong', 'departure', 'cause', 'flight', 'stand', 'nothappy']
['SCREEN_NAME', 'reading', 'boarding', 'policy', 'group', 'traveller', 'range', 'possible', 'together']
['SCREEN_NAME', 'finally', 'passbook', 'option', 'thank']
['SCREEN_NAME', 'flight', 'cancel', 'flight', 'hours', 'hang', 'little']
['SCREEN_NAME', 'sayin', 'greenville', 'instead', 'raleigh', 'kid']
['SCREEN_NAME', 'flight', 'cancel', 'flightled', 'today']
['SCREEN_NAME', 'father', 'enough', 'offer', 'hotel', 'coupon']
['SCREEN_NAME', 'imagine', 'dragon', '.....', 'please', 'vega']
['SCREEN_NAME', 'please', 'reply']
['SCREEN_NAME', 'flight']
['SCREEN_NAME', 'loyal', 'customer', 'easy', 'bring', 'resolution', 'issue', 'reward', 'program']
['SCREEN_NAME', 'plane', 'coming', 'flight']
['SCREEN_NAME', 'winner', 'SCREEN_NAME', 'destinationdragons']
['SCREEN_NAME', 'please', 'SCREEN_NAME', 'friday', 'promise', 'southwest', 'going', 'mhtforlife']
['SCREEN_NAME', 'recommend', 'upgrade', 'using', 'email', 'texting', 'caller']
['SCREEN_NAME', 'cancel', 'flight', 'flight', 'another', 'airline', 'leave', 'cancel', 'flight', 'call', 'jetblue', 'flight']
['SCREEN_NAME', 'thankfully', 'court', 'update', 'post']
['SCREEN_NAME', 'respond', 'complaint', 'cultural', 'appropriation', 'spirit', 'airline']
['SCREEN_NAME', 'fleet', 'fleek']
['SCREEN_NAME', 'watch', 'oscar', 'though', 'right', 'flight', 'missingtheoscars']
['SCREEN_NAME', 'enough', 'money', 'parking', 'garage', 'fault', 'pay', '10mins']
['SCREEN_NAME', 'alert', 'arrive', 'flight', 'delay']
['SCREEN_NAME', 'cookie', 'stuckonaplane', 'babyfood']
['SCREEN_NAME', 'suppose', 'change', 'future']
['SCREEN_NAME', 'SCREEN_NAME', 'sorry', 'please', 'inflght']
['SCREEN_NAME', 'thanks', 'figure', 'hopefully', 'hashtag', 'change', 'abcletjetbluestreamfeed']
['SCREEN_NAME', 'plane', 'ground', 'instead', 'divert', 'plane', 'add', 'flight']
['SCREEN_NAME', 'success', 'jetblue']
['SCREEN_NAME', 'thought', 'waiver', 'change', 'travel', 'boston', 'weekend']
['SCREEN_NAME', 'parking', 'orlando', 'airport', 'flight', 'land']
['SCREEN_NAME', 'partnership', 'american', 'express']
['SCREEN_NAME', 'airport', 'phone', 'everyone', 'different', 'things']
['SCREEN_NAME', 'floor', 'instead', 'overhead', 'trip', 'safetyfirst', 'sarcasm']
['SCREEN_NAME', 'understand', 'could', 'standby', 'hours', 'prior', 'flight']
['SCREEN_NAME', 'remember', 'doubtful', 'hopeful']
['SCREEN_NAME', 'thanks', 'getting', 'call', 'number', 'earlier', 'option', 'leave', 'rings']
```

```
['SCREEN_NAME', 'travel', 'question', 'could', 'please', 'follow']
['SCREEN_NAME', 'never', 'unpleasant', 'travel', 'experience', 'might', 'flight', 'tomorrow', 'insane']
['SCREEN_NAME', 'another', 'twist', 'miami', 'return', 'unsecured', 'cargo', 'stressful', 'vacation']
['SCREEN_NAME', 'power', 'outlet', 'site', 'forever']
['SCREEN_NAME', 'SCREEN_NAME', 'always', 'customer', 'service', 'manager', 'threaten', 'customer', 'standing', 'understaffed', 'li
['SCREEN_NAME', 'flight', 'please', 'flight', 'already', 'issue', 'century']
['SCREEN_NAME', 'waiting', 'waiting', 'luggage']
['SCREEN_NAME', 'cavan', 'gofundme', 'going', 'breath', 'tonight', 'airline', 'care', 'money']
['SCREEN_NAME', 'cancel', 'flight', 'reschedule', 'flight', 'still', 'getting', 'update', 'current', 'flight', 'account']
['SCREEN_NAME', 'sadly', 'try', 'airport', 'agent', 'flight', 'strand', 'chance', 'hotel']
['SCREEN_NAME', 'impressive', 'concern', 'flight', 'flight', 'attndt', 'wrecking', 'havoc', 'customer', 'travel', 'smugsmirk', 'no
['SCREEN_NAME', 'collectively', 'since', 'friday', 'customer', 'service', 'issue', 'resolve', 'frustrate']
['SCREEN_NAME', 'need', 'together', 'ticket', 'agent', 'miss', 'flight']
['SCREEN_NAME', 'cancel', 'flight', 'flight', 'rebook', 'minutes', 'still', 'waiting']
['SCREEN_NAME', 'flight', 'cancel', 'flightled', 'reschedule', 'monday', 'unacceptable']
['SCREEN_NAME', 'flight', 'cancel', 'flightlations', 'finally', 'flight', 'given', 'seats', 'seriously']
['SCREEN_NAME', 'chance', 'travel', 'advisory', 'weekend']
['SCREEN_NAME', 'check', 'need', 'add', 'please', 'evaluate']
['SCREEN_NAME', 'thanks', 'flight', 'cancel', 'flightled', 'maintenance', 'reason', 'available', 'flight', 'flight']
['SCREEN_NAME', 'service', 'leaf', 'desire', 'flight', 'early', 'short', 'staff', 'means', 'tarmac', 'thanks']
['SCREEN_NAME', 'thanks']
['SCREEN_NAME', 'anything', 'happen', 'service', 'really', 'sucks.#usairwayssucks']
['SCREEN_NAME', 'recently', 'move', 'license', 'book', 'flight', 'month', 'license', 'change']
['SCREEN_NAME', 'companion', 'voucher', 'online']
['SCREEN_NAME', 'baggage', 'allowance', 'flight', 'glasgow', 'thanks']
['SCREEN_NAME', 'travelling', 'sunday', 'flight', 'cancel', 'flightled', 'flight', 'cancel', 'flightled', 'disconnect', 'reflight
['SCREEN_NAME', 'would', 'consider', 'continue', 'point', 'program', 'receive', 'perk', 'continue', 'customer', 'service', 'sensel
['SCREEN_NAME', 'respond', 'watch', 'cutting', 'prolong', 'beyond', 'frustrate']
['SCREEN_NAME', 'horrible.went', 'online', 'cancel', 'flight', 'flight', 'button-4that', 'call', 'amp;wait', 'mins&amp;put', '.80
['SCREEN_NAME', 'thanks', 'getting', 'airline', 'future']
['SCREEN_NAME', 'flight', 'tomorrow', 'cancel', 'flight']
['SCREEN_NAME', 'could', 'md80/dc10', 'would', 'happy', 'plane']
['SCREEN_NAME', 'another', 'reason', 'south', 'winter', 'month', 'travel']
['SCREEN_NAME', 'sitting', 'plane', 'columbus', 'suppose', 'leave', 'mechanic', 'service', 'shock', 'absorber']
['SCREEN_NAME', 'thank', 'quick', 'customer', 'service', 'today', 'refundprocedurenottoopainful', 'winter', 'weather', 'fault']
['SCREEN_NAME', 'local', 'agent', 'person', 'answer', 'question', 'phone', 'service', 'terrible']
['SCREEN_NAME', 'airline', 'people', 'first', 'receive', 'hotel', 'voucher', 'terrible', 'customer', 'service']
['SCREEN_NAME', 'afford', 'distance', 'trip', 'though']
['SCREEN_NAME', 'intentionally', 'try', 'customer', 'money']
['SCREEN_NAME', 'impossible', 'contact', 'anyone', 'airline', 'board', 'plane', 'deplane']
['SCREEN_NAME', 'unacceptable', 'clear', 'lose', 'customer', 'matter', 'american', 'bankruptcy']
['SCREEN_NAME', 'SCREEN_NAME', 'try', 'anything', 'would', 'stewardess', 'serving', 'drunk', 'drink']
['SCREEN_NAME', 'still', 'talk', 'anyone', 'prepare', 'situation', 'service']
['SCREEN_NAME', 'tjycqh', 'airport', 'since', 'schedule', 'ashamed', 'disgust']
['SCREEN_NAME', 'start', 'water', 'SCREEN_NAME']
['SCREEN_NAME', 'thanks', 'charge', 'anything']
['SCREEN_NAME', 'waiting', 'three', 'hours', 'someone', 'rebook', 'flight', 'weather', 'service', 'actually']
['SCREEN_NAME', 'thank', 'response', 'try', 'number', 'airway', 'number']
['SCREEN_NAME', 'flight', 'cancel', 'flightled', 'answer', 'flight', 'please']
['SCREEN_NAME', 'awesome', 'customer', 'service', 'sneak', 'cookie']
['SCREEN_NAME', 'friend', 'flight', 'aa362', 'dfw&gt;&gt;mke', 'cancel', 'flight', 'weather', 'option']
['SCREEN_NAME', 'flight', 'matter', 'flight', 'cancel', 'flight', 'morning', 'since']
['SCREEN_NAME', 'sitting', 'flight', 'sabre', 'backup', 'place', 'sabre', 'fix']
['SCREEN_NAME', 'please', 'steal', 'incompetent', 'employee', 'abcnews', 'msnbc', 'photography']
['SCREEN_NAME', 'SCREEN_NAME', 'woody', 'making', 'worse', 'understaffing', 'rudeness', 'rookie', 'mistake']
['SCREEN_NAME', 'appreciate', 'update', 'appreciate', 'pilot', 'effort', 'explain', 'accurate', 'authoritative', 'comms', 'vital'
['SCREEN_NAME', 'overweight', 'dozen', 'passenger', 'luggage', 'seriously']
['SCREEN_NAME', 'flight', 'cancel', 'flightled', 'leaving', 'tomorrow', 'morning', 'rebooked', 'tuesday', 'night', 'flight', 'arr
```

In [12]:

```python
from gensim import corpora
dictionary = corpora.Dictionary(text_data)
corpus = [dictionary.doc2bow(text) for text in text_data]

import pickle
pickle.dump(corpus, open('corpus.pkl', 'wb'))
dictionary.save('dictionary.gensim')
```

In [13]:

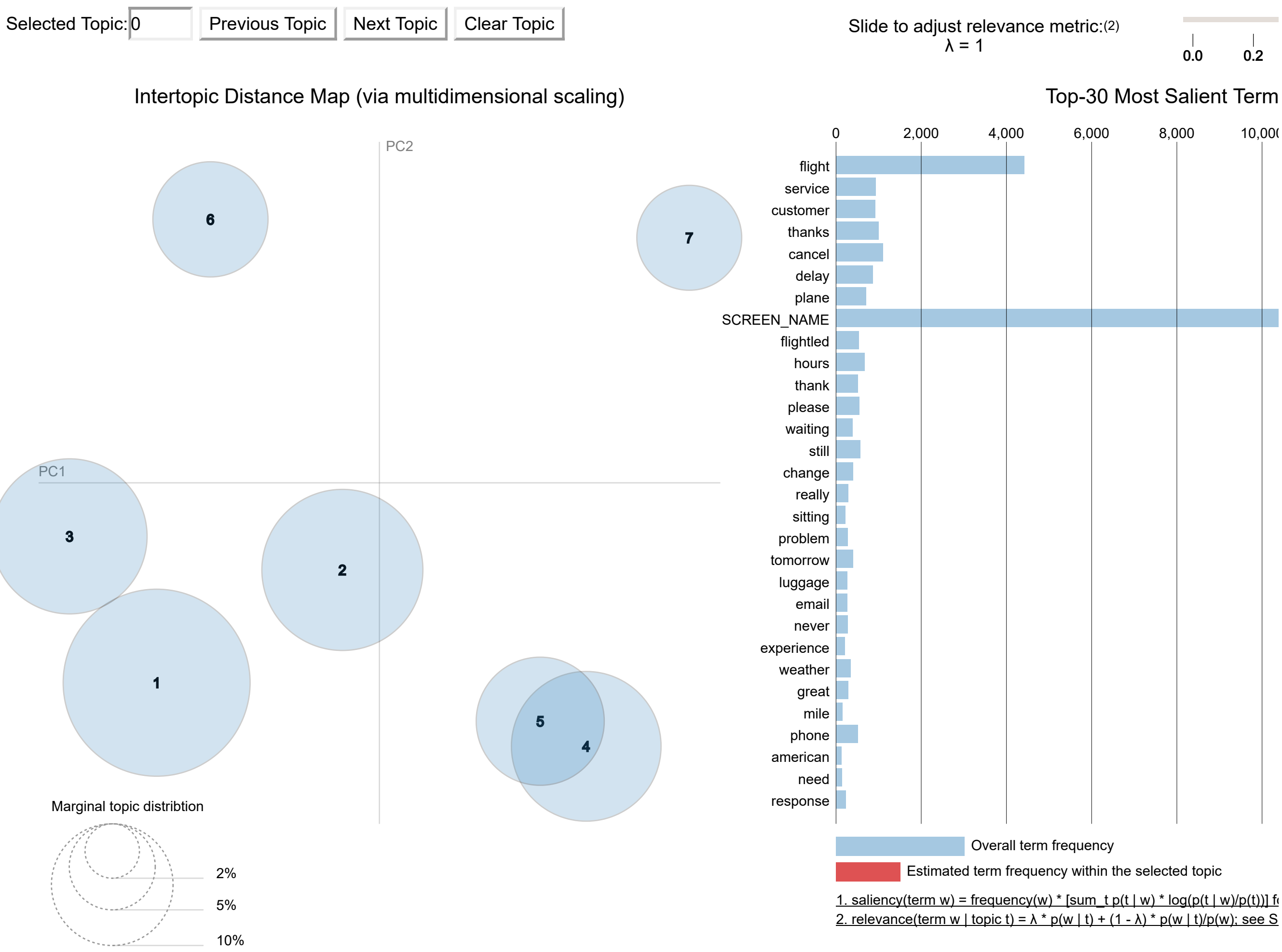
```python
import gensim
NUM_TOPICS = 7
ldamodel = gensim.models.ldamodel.LdaModel(corpus, num_topics = NUM_TOPICS, id2word=dictionary, passes=15)
ldamodel.save('model5.gensim')
topics = ldamodel.print_topics(num_words=4)
for topic in topics:
    print(topic)
```

```
(0, '0.145*"SCREEN_NAME" + 0.119*"flight" + 0.048*"cancel" + 0.024*"flightled"')
(1, '0.118*"SCREEN_NAME" + 0.044*"flight" + 0.025*"sitting" + 0.019*"plane"')
(2, '0.147*"SCREEN_NAME" + 0.064*"service" + 0.063*"customer" + 0.016*"email"')
(3, '0.150*"SCREEN_NAME" + 0.059*"flight" + 0.055*"delay" + 0.035*"plane"')
(4, '0.211*"SCREEN_NAME" + 0.059*"thanks" + 0.028*"thank" + 0.023*"flight"')
(5, '0.151*"SCREEN_NAME" + 0.032*"please" + 0.024*"really" + 0.017*"luggage"')
(6, '0.147*"SCREEN_NAME" + 0.018*"american" + 0.013*"ticket" + 0.012*"system"')
```

In [14]:

```
import pyLDAvis.gensim

pyLDAvis.enable_notebook()
panel = pyLDAvis.gensim.prepare(ldamodel, corpus, dictionary)
pyLDAvis.display(panel)
```

Out[14]:



In this topic modelling I have chosen 7 topics to reflect all the possible negative reasons.

In [ ]:

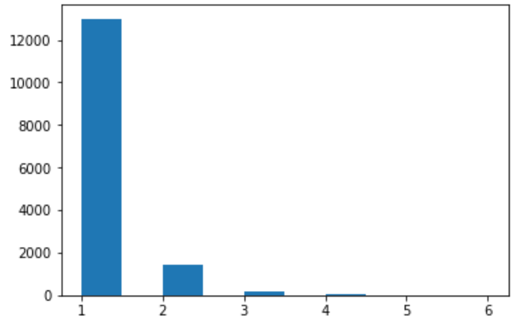
```
Additional analysis, let's look at the number of '@' in the tweets to indentify a pattern.
```

In [143]:

```
list =[]
for i in range(14640):
    list.append(df['text'][i].count('@'))

newdat=pd.DataFrame(list)

newdat.columns=["at_count"]

newdat.head(5)

df_3=pd.concat([df,newdat],axis=1)

plt.hist(df_3["at_count"])
```

Out[143]:

```
(array([1.2995e+04, 0.0000e+00, 1.4200e+03, 0.0000e+00, 1.6900e+02,
        0.0000e+00, 4.1000e+01, 0.0000e+00, 1.3000e+01, 2.0000e+00]),
 array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5, 6. ]),
 <a list of 10 Patch objects>)
```



Here we can see that the number of mentions i.e usage of '@' symbols follows a poisson distribution

Let's look at the relationship b/w length of the tweet and sentiment, first we create a new column called tweet_length append it to the data frame and then we shall look at the cou

In [15]:

```
#Tweet Length and sentiment
list2 =[]
for i in range(14640):
    list2.append(len(df['text'][i]))

newdat2=pd.DataFrame(list2)

newdat2.columns=["tweet_length"]

df_4=pd.concat([df,newdat2],axis=1)

# df_4.head(10)
```

In [43]:

```
df_new = df_4.groupby(['tweet_length', 'airline_sentiment'])['airline_sentiment'].count()
# df2.plot(kind='bar', stacked=True, figsize=[20,10])

df_new.head(10)
```

Out[43]:

```
tweet_length  airline_sentiment
12            negative             1
              neutral              5
              positive             2
13            negative             1
              neutral              2
              positive             1
14            negative             2
              neutral              7
              positive             5
15            negative             1
Name: airline_sentiment, dtype: int64
```

In [17]:

```
# postive = pd.read_csv('postive_plots.txt', sep='\t',header=None)
# # df_new.plot(kind='bar', stacked=True, figsize=[20,10])
# postive.plot(kind='line', figsize=[20,10])


import pandas as pd
import numpy as np
import matplotlib.pylab as plt
%matplotlib inline
import seaborn as sns
plot = pd.read_csv('plotfile.txt', sep='\t')

plot.columns = ['Number','Sentiment','Value']

plot.groupby(['Sentiment']).groups.keys()
plot1 = plot.loc[plot['Sentiment'] == 'negative']
plot2 = plot.loc[plot['Sentiment'] == 'positive']
plot3 = plot.loc[plot['Sentiment'] == 'neutral']
```

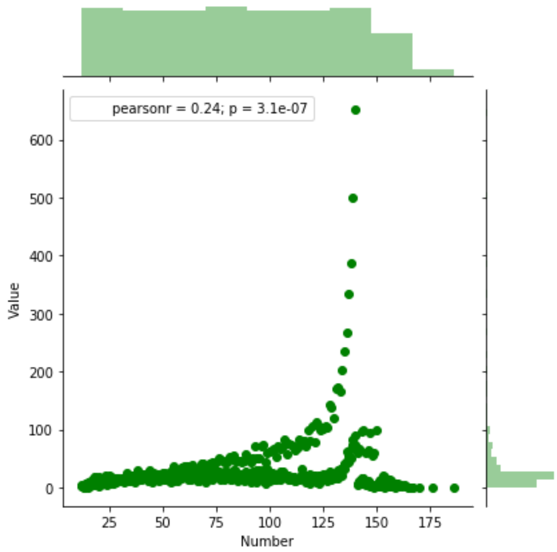
In [28]:

```
sns.jointplot(x='Number', y='Value', data=plot, color='g')
```

```
C:\Users\Lameware\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.

C:\Users\Lameware\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning:

The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
```

Out[28]:

```
<seaborn.axisgrid.JointGrid at 0x2c9a7860080>
```

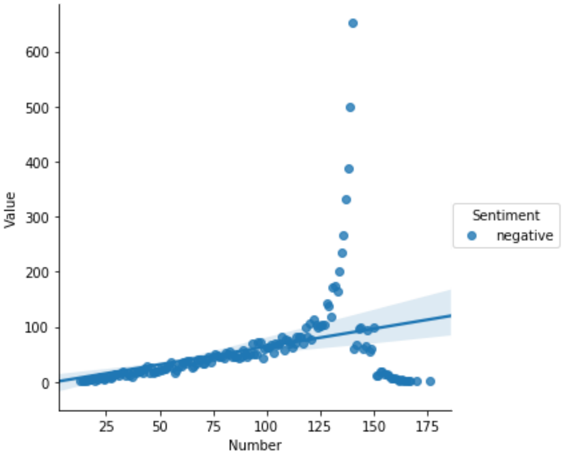


In this plot we can see that the first curve, (negative sentiments) is much more frequently associated with longer tweets as compared to positve/neutral

```
In [89]: sns.lmplot(x='Number', y='Value', data=plot1,
                    fit_reg=True,
                    hue='Sentiment')
```

```
C:\Users\Lameware\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning:

Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future
x, `arr[np.array(seq)]`, which will result either in an error or a different result.
```

```
Out[89]: <seaborn.axisgrid.FacetGrid at 0x27162c28518>
```

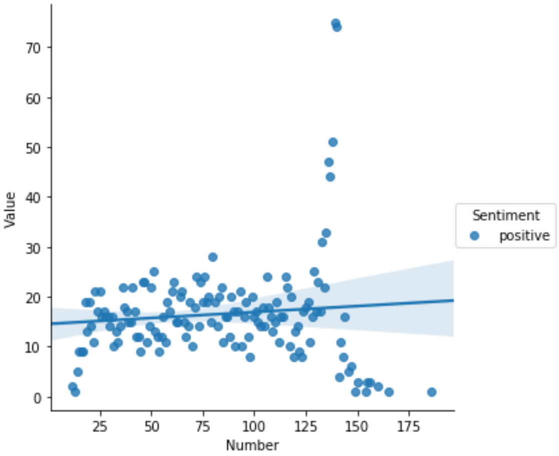


This plot shows us the frequency vs tweet length for negative sentiments, we can see that for expressing themselves and makir writing negative feedback tend to write longer tweets and use multiple mentions to make it more visible.

```
In [93]: sns.lmplot(x='Number', y='Value', data=plot2,
                    fit_reg=True,
                    hue='Sentiment')
```

```
C:\Users\Lameware\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning:

Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future
x, `arr[np.array(seq)]`, which will result either in an error or a different result.
```

```
Out[93]: <seaborn.axisgrid.FacetGrid at 0x27162deca90>
```

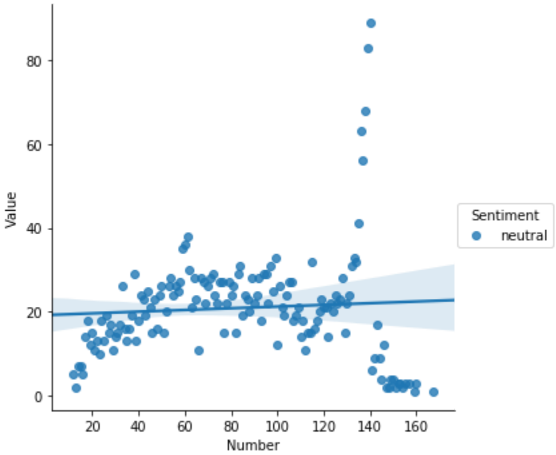


Frequency vs tweet_length plot for positve sentiment varies greatly from negative sentiment (look at the y-axis for both plots), p tweets while expressing gratitude, or satisfacation with service.

```
In [92]: sns.lmplot(x='Number', y='Value', data=plot3,
                    fit_reg=True,
                    hue='Sentiment')
```

```
C:\Users\Lameware\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning:

Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future
x, `arr[np.array(seq)]`, which will result either in an error or a different result.
```

```
Out[92]: <seaborn.axisgrid.FacetGrid at 0x27162e51c88>
```



Similar trend as postive sentiment, shorter length is prevelent.