Analysis of Yelp Business Intelligence Data

We will analyze a subset of Yelp's business, reviews and user data. This dataset comes to us from Kaggle although we have taken steps to pull this data into a publis s3 bucket:
`s3://cis9760-yelpdataset/yelp-light/*business.json`

Installation and Initial Setup

Begin by installing the necessary libraries that you may need to conduct your analysis. At the very least, you must install `pandas` and `matplotlib`

In [1]:
```
%%info
```

Current session configs: `{'conf': {'spark.pyspark.python': 'python3', 'spark.pyspark.virtualenv.enabled': 'true', 'spark.pyspark.virtualenv.type': 'native', 'spark.pyspark.virtualenv.bin.path': '/usr/bin/virtualenv'}, 'kind': 'pyspark'}`

No active sessions.

In [2]:
```
sc.install_pypi_package( "IPython" )
```

▸ Spark Job Progress

```
Starting Spark application
```

ID	YARN Application ID	Kind	State	
0	application_1651374988795_0001	pyspark	idle	Link (http://ip-172-3 2.compute.internal:20888/proxy/application_1651

```
SparkSession available as 'spark'.

Collecting IPython
  Downloading https://files.pythonhosted.org/packages/e0/fe/9ebd702978bd9730bcabba366e98b53db955c5a7dc78d4e51f7514f08c2/ipython-7.33.0-p3-none-any.whl
```

```
(https://files.pythonhosted.org/packages/e0/fe/9ebd7029b78d9730bcab
a366e98b53db955c5a7dc78d4e51f7514f08c2/ipython-7.33.0-py3-none-any.w
l) (793kB)
Collecting matplotlib-inline (from IPython)
  Downloading https://files.pythonhosted.org/packages/a6/2d/2230afd5
0c70074e80fd06857ba2bdc5f10c055bd9125665fe276fadb67/matplotlib_inlin
-0.1.3-py3-none-any.whl
(https://files.pythonhosted.org/packages/a6/2d/2230afd570c70074e80fd
6857ba2bdc5f10c055bd9125665fe276fadb67/matplotlib_inline-0.1.3-py3-n
ne-any.whl)
Requirement already satisfied: setuptools>=18.5 in /mnt/tmp/16513755
9793-0/lib/python3.7/site-packages (from IPython)
Collecting pexpect>4.3; sys_platform != "win32" (from IPython)
  Downloading https://files.pythonhosted.org/packages/39/7b/88dbb785
81c28a102619d46423cb853b46dbccc70d3ac362d99773a78ce/pexpect-4.8.0-py

.py3-none-any.whl
(https://files.pythonhosted.org/packages/39/7b/88dbb785881c28a102619
46423cb853b46dbccc70d3ac362d99773a78ce/pexpect-4.8.0-py2.py3-none-an
.whl) (59kB)
Collecting decorator (from IPython)
  Downloading https://files.pythonhosted.org/packages/d5/50/83c593b0
763e1161326b3b8c6686f0f4b0f24d5526546bee538c89837d6/decorator-5.1.1-
y3-none-any.whl
(https://files.pythonhosted.org/packages/d5/50/83c593b07763e1161326b
b8c6686f0f4b0f24d5526546bee538c89837d6/decorator-5.1.1-py3-none-any.
hl)
Collecting traitlets>=4.2 (from IPython)
  Downloading https://files.pythonhosted.org/packages/37/46/be8a3c03
bd3673f4800fa7f46eda972dfa2990089a51ec5dd0a26ed33e9/traitlets-5.1.1-
y3-none-any.whl
(https://files.pythonhosted.org/packages/37/46/be8a3c030bd3673f4800f
7f46eda972dfa2990089a51ec5dd0a26ed33e9/traitlets-5.1.1-py3-none-any.
hl) (102kB)
Collecting jedi>=0.16 (from IPython)
  Downloading https://files.pythonhosted.org/packages/b3/0e/836f12ec
0075161e365131f13f5758451645af75c2becf61c6351ecec39/jedi-0.18.1-py2.
y3-none-any.whl
(https://files.pythonhosted.org/packages/b3/0e/836f12ec50075161e3651
1f13f5758451645af75c2becf61c6351ecec39/jedi-0.18.1-py2.py3-none-any.
hl) (1.6MB)
Collecting prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 (from IPytho
)
  Downloading https://files.pythonhosted.org/packages/3f/2d/dcb44d69
388ca2ee1a4a4d3c204ab66b36975c0d5166781eaeeff76b882/prompt_toolkit-3
0.29-py3-none-any.whl
(https://files.pythonhosted.org/packages/3f/2d/dcb44d69f388ca2ee1a4a
d3c204ab66b36975c0d5166781eaeeff76b882/prompt_toolkit-3.0.29-py3-non
-any.whl) (381kB)
Collecting pickleshare (from IPython)
  Downloading https://files.pythonhosted.org/packages/9a/41/220f49aa
```

```
a88bc6fa6cba8d05ecf24676326156c23b991e80b3f2fc24c77/pickleshare-0.7.
-py2.py3-none-any.whl
(https://files.pythonhosted.org/packages/9a/41/220f49aaea88bc6fa6cba
d05ecf24676326156c23b991e80b3f2fc24c77/pickleshare-0.7.5-py2.py3-non
-any.whl)
Collecting backcall (from IPython)
  Downloading https://files.pythonhosted.org/packages/4c/1c/ff6546b6
12603d8dd1070aa3c3d273ad4c07f5771689a7b69a550e8c951/backcall-0.2.0-p
2.py3-none-any.whl
(https://files.pythonhosted.org/packages/4c/1c/ff6546b6c12603d8dd107
aa3c3d273ad4c07f5771689a7b69a550e8c951/backcall-0.2.0-py2.py3-none-a
y.whl)
Collecting pygments (from IPython)
  Downloading https://files.pythonhosted.org/packages/5c/8e/1d901795
034297fffa336c72e693a5b51bbf85141b24a763882cf1977b5/Pygments-2.12.0-
y3-none-any.whl
(https://files.pythonhosted.org/packages/5c/8e/1d9017950034297fffa33
c72e693a5b51bbf85141b24a763882cf1977b5/Pygments-2.12.0-py3-none-any.
hl) (1.1MB)
Collecting ptyprocess>=0.5 (from pexpect>4.3; sys_platform != "win32
->IPython)
  Downloading https://files.pythonhosted.org/packages/22/a6/85889725
d0deac81a172289110f31629fc4cee19b6f01283303e18c8db3/ptyprocess-0.7.0
py2.py3-none-any.whl
(https://files.pythonhosted.org/packages/22/a6/858897256d0deac81a172
89110f31629fc4cee19b6f01283303e18c8db3/ptyprocess-0.7.0-py2.py3-none
any.whl)
Collecting parso<0.9.0,>=0.8.0 (from jedi>=0.16->IPython)
  Downloading https://files.pythonhosted.org/packages/05/63/8011bd08
4111858f79d2b09aad86638490d62fbf881c44e434a6dfca87b/parso-0.8.3-py2.
y3-none-any.whl
(https://files.pythonhosted.org/packages/05/63/8011bd08a4111858f79d2
09aad86638490d62fbf881c44e434a6dfca87b/parso-0.8.3-py2.py3-none-any.
hl) (100kB)
Collecting wcwidth (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.
->IPython)
  Downloading https://files.pythonhosted.org/packages/59/7c/e39aca59
badaf1b78e8f547c807b04dae603a433d3e7a7e04d67f2ef3e5/wcwidth-0.2.5-py
.py3-none-any.whl
(https://files.pythonhosted.org/packages/59/7c/e39aca596badaf1b78e8f
47c807b04dae603a433d3e7a7e04d67f2ef3e5/wcwidth-0.2.5-py2.py3-none-an
.whl)
Installing collected packages: traitlets, matplotlib-inline, ptyproc
ss, pexpect, decorator, parso, jedi, wcwidth, prompt-toolkit, pickle
hare, backcall, pygments, IPython
Successfully installed IPython-7.33.0 backcall-0.2.0 decorator-5.1.1
jedi-0.18.1 matplotlib-inline-0.1.3 parso-0.8.3 pexpect-4.8.0 pickle
hare-0.7.5 prompt-toolkit-3.0.29 ptyprocess-0.7.0 pygments-2.12.0 tr
itlets-5.1.1 wcwidth-0.2.5
```

In [3]:
```
sc.install_pypi_package( "matplotlib==3.2.1" )
sc.install_pypi_package( "pandas==1.0.3" )
#sc.install_pypi_package("seaborn==0.11.2")
```

▸ Spark Job Progress

```
Collecting matplotlib==3.2.1
  Downloading https://files.pythonhosted.org/packages/b2/c2/71fcf957710f3ba1f09088b35776a799ba7dd95f7c2b195ec800933b276b/matplotlib-3.2.1-cp37-cp37m-manylinux1_x86_64.whl (https://files.pythonhosted.org/packages/b2/c2/71fcf957710f3ba1f09088b35776a799ba7dd95f7c2b195ec800933b276b/matplotlib-3.2.1-cp37-cp37m-manylinux1_x86_64.whl) (12.4MB)
Collecting python-dateutil>=2.1 (from matplotlib==3.2.1)
  Downloading https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.py3-none-any.whl (https://files.pythonhosted.org/packages/36/7a/87837f39d0296e723bb9b62bbb257d0355c7f6128853c78955f57342a56d/python_dateutil-2.8.2-py2.py3-none-any.whl) (247kB)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 (from matplotlib==3.2.1)
  Downloading https://files.pythonhosted.org/packages/d9/41/d9cfb4410589805cd787f8a82cddd13142d9bf7449d12adf2d05a4a7d633/pyparsing-3.0.8-py3-none-any.whl (https://files.pythonhosted.org/packages/d9/41/d9cfb4410589805cd787f8a82cddd13142d9bf7449d12adf2d05a4a7d633/pyparsing-3.0.8-py3-none-any.whl) (98kB)
Collecting cycler>=0.10 (from matplotlib==3.2.1)
  Downloading https://files.pythonhosted.org/packages/5c/f9/695d6bedebd747e5eb0fe8fad57b72fdf25411273a39791cde838d5a8f51/cycler-0.11.0-py3-none-any.whl (https://files.pythonhosted.org/packages/5c/f9/695d6bedebd747e5eb0fe8fad57b72fdf25411273a39791cde838d5a8f51/cycler-0.11.0-py3-none-any.whl)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib64/python3.7/site-packages (from matplotlib==3.2.1)
Collecting kiwisolver>=1.0.1 (from matplotlib==3.2.1)
  Downloading https://files.pythonhosted.org/packages/51/50/9a9a94afa26c50fc5d9127272737806990aa698c7a1c220b8e5075e70304/kiwisolver-1.4.2-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl (https://files.pythonhosted.org/packages/51/50/9a9a94afa26c50fc5d9127272737806990aa698c7a1c220b8e5075e70304/kiwisolver-1.4.2-cp37-cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl) (1.1MB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil>=2.1->matplotlib==3.2.1)
Collecting typing-extensions; python_version < "3.8" (from kiwisolver>=1.0.1->matplotlib==3.2.1)
```

```
  Downloading https://files.pythonhosted.org/packages/75/e1/932e06004
039dd670c9d5e1df0cd606bf46e29a28e65d5bb28e894ea29c9/typing_extensions
-4.2.0-py3-none-any.whl
(https://files.pythonhosted.org/packages/75/e1/932e06004039dd670c9d5e
1df0cd606bf46e29a28e65d5bb28e894ea29c9/typing_extensions-4.2.0-py3-no
ne-any.whl)
Installing collected packages: python-dateutil, pyparsing, cycler, ty
ping-extensions, kiwisolver, matplotlib
Successfully installed cycler-0.11.0 kiwisolver-1.4.2 matplotlib-3.2.
1 pyparsing-3.0.8 python-dateutil-2.8.2 typing-extensions-4.2.0

Collecting pandas==1.0.3
  Downloading https://files.pythonhosted.org/packages/4a/6a/94b219b8e
a0f2d580169e85ed1edc0163743f55aaeca8a44c2e8fc1e344e/pandas-1.0.3-cp37
-cp37m-manylinux1_x86_64.whl
(https://files.pythonhosted.org/packages/4a/6a/94b219b8ea0f2d580169e8
5ed1edc0163743f55aaeca8a44c2e8fc1e344e/pandas-1.0.3-cp37-cp37m-manyli
nux1_x86_64.whl) (10.0MB)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3
.7/site-packages (from pandas==1.0.3)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib64/pyth
on3.7/site-packages (from pandas==1.0.3)
Requirement already satisfied: python-dateutil>=2.6.1 in /mnt/tmp/165
1375529793-0/lib/python3.7/site-packages (from pandas==1.0.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/s
ite-packages (from python-dateutil>=2.6.1->pandas==1.0.3)
Installing collected packages: pandas
Successfully installed pandas-1.0.3
```

In [4]:
```
sc.install_pypi_package("scipy==1.7.1")
sc.install_pypi_package( "seaborn==0.11.2" )
#import seaborn as sns
```

▸ Spark Job Progress

```
Collecting scipy==1.7.1
  Downloading https://files.pythonhosted.org/packages/b5/6b/8bc0b61eb
f824f8c3979a31368bbe38dd247590049a994ab0ed077cb56dc/scipy-1.7.1-cp37-
cp37m-manylinux_2_5_x86_64.manylinux1_x86_64.whl
(https://files.pythonhosted.org/packages/b5/6b/8bc0b61ebf824f8c3979a3
1368bbe38dd247590049a994ab0ed077cb56dc/scipy-1.7.1-cp37-cp37m-manylin
ux_2_5_x86_64.manylinux1_x86_64.whl) (28.5MB)
Requirement already satisfied: numpy<1.23.0,>=1.16.5 in /usr/local/li
b64/python3.7/site-packages (from scipy==1.7.1)
Installing collected packages: scipy
Successfully installed scipy-1.7.1
```

```
Collecting seaborn==0.11.2
  Downloading https://files.pythonhosted.org/packages/10/5b/0479d7d845b5ba410ca702ffcd7f2cd95a14a4dfff1fde2637802b258b9b/seaborn-0.11.2-py3-none-any.whl (292kB)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib64/python3.7/site-packages (from seaborn==0.11.2)
Requirement already satisfied: scipy>=1.0 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from seaborn==0.11.2)
Requirement already satisfied: matplotlib>=2.2 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from seaborn==0.11.2)
Requirement already satisfied: pandas>=0.23 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from seaborn==0.11.2)
Requirement already satisfied: python-dateutil>=2.1 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn==0.11.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn==0.11.2)
Requirement already satisfied: cycler>=0.10 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn==0.11.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn==0.11.2)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/site-packages (from pandas>=0.23->seaborn==0.11.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-packages (from python-dateutil>=2.1->matplotlib>=2.2->seaborn==0.11.2)
Requirement already satisfied: typing-extensions; python_version < "3.8" in /mnt/tmp/1651375529793-0/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplotlib>=2.2->seaborn==0.11.2)
Installing collected packages: seaborn
Successfully installed seaborn-0.11.2
```

Importing

Now, import the installed packages from the previous block below.

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

In [6]:
```
%matplotlib inline
```

In [7]:
```
sc.list_packages() #we are just trying to make seaborn work  here
```

```
Package                    Version
-------------------------- ---------
backcall                   0.2.0
beautifulsoup4             4.9.1
boto                       2.49.0
click                      7.1.2
cycler                     0.11.0
decorator                  5.1.1
ipython                    7.33.0
jedi                       0.18.1
jmespath                   0.10.0
joblib                     0.16.0
kiwisolver                 1.4.2
lxml                       4.5.2
matplotlib                 3.2.1
matplotlib-inline          0.1.3
mysqlclient                1.4.2
nltk                       3.5
nose                       1.3.4
numpy                      1.16.5
pandas                     1.0.3
parso                      0.8.3
pexpect                    4.8.0
pickleshare                0.7.5
pip                        9.0.1
prompt-toolkit             3.0.29
ptyprocess                 0.7.0
py-dateutil                2.2
Pygments                   2.12.0
pyparsing                  3.0.8
python-dateutil            2.8.2
python37-sagemaker-pyspark 1.4.0
pytz                       2020.1
PyYAML                     5.3.1
regex                      2020.7.14
scipy                      1.7.1
seaborn                    0.11.2
setuptools                 28.8.0
six                        1.13.0
soupsieve                  1.9.5
tqdm                       4.48.2
traitlets                  5.1.1
typing-extensions          4.2.0
wcwidth                    0.2.5
wheel                      0.29.0
windmill                   1.6
```

Loading Data

We are finally ready to load data. Using spark load the data from S3 into a dataframe object that we can manipulate further down in our analysis.

In [8]:
```
business_data  = spark.read.json( 's3://yelp-reviews-dataset/yelp_acad
```

▸ Spark Job Progress

In [9]:
```
business_data.show(10)
```

▸ Spark Job Progress

```
+--------------------+--------------------+--------------------+-----
--------------+-------------+--------------------+-------+---------
-+-----------+--------------------+----------+-----------+-----+--
---+
|             address|          attributes|         business_id|
categories|          city|               hours|is_open|  latitude|
longitude|                name|postal_code|review_count|stars|state|
+--------------------+--------------------+--------------------+-----
--------------+-------------+--------------------+-------+---------
-+-----------+--------------------+----------+-----------+-----+--
---+
|1616 Chapala St, ...|[,,,,,,,,,,, True...|Pns2l4eNsfO8kk83d...|Docto
rs, Traditio...| Santa Barbara|                null|      0|34.426678
7|-119.7111968|Abby Rappoport, L...|     93101|          7|  5.0|
CA|
|87 Grasso Plaza S...|[,,,,,,,,, True,,...|mpf3x-BjTdTEA3yCZ...|Shipp
ing Centers,...|        Affton|[8:0-18:30, 0:0-0...|      1| 38.55112
6|  -90.335695|       The UPS Store|     63123|         15|  3.0|
MO|
|5255 E Broadway Blvd|[,,,,,,, True,, T...|tUFrWirKiKi_TAnsV...|Depar
tment Stores...|        Tucson|[8:0-23:0, 8:0-22...|      0| 32.22323
6| -110.880452|              Target|     85711|         22|  3.5|
AZ|
|         935 Race St|[,, u'none',,,,, ...|MTSW4McQd7CbVtyjq...|Resta
urants, Food...|  Philadelphia|[7:0-21:0, 7:0-20...|      1|39.955505
2| -75.1555641|  St Honore Pastries|     19107|         80|  4.0|
PA|
|       101 Walnut St|[,,,,,,, True,, T...|mWMc6_wTdE0EUBKIG...|Brewp
ubs, Breweri...|    Green Lane|[12:0-22:0,, 12:0...|      1|40.338182
7| -75.4716585|Perkiomen Valley ...|     18054|         13|  4.5|
```

```
PA|
|       615 S Main St|[,, u'none', None...|CF33F8-E6oudUQ46H...|Burge
rs, Fast Foo...|  Ashland City|[9:0-0:0, 0:0-0:0...|      1| 36.26959
3|  -87.058943|      Sonic Drive-In|      37015|           6|  2.0|
TN|
|8522 Eager Road, ...|[,,,,,,,, True,, T...|n_0UpQx1hsNbnPUSl...|Sport
ing Goods, F...|     Brentwood|[10:0-18:0, 0:0-0...|      1| 38.62769
5|  -90.340465|     Famous Footwear|      63144|          13|  2.5|
MO|
|  400 Pasadena Ave S|                null|qkRM_2X51Yqxk3btl...|Synag
ogues, Relig...|St. Petersburg|[9:0-17:0, 9:0-17...|      1|  27.7665
9|  -82.732983|      Temple Beth-El|      33707|           5|  3.5|
FL|
|   8025 Mackenzie Rd|[,, u'full_bar', ...|k0hlBqXX-Bt0vf1op...|Pubs,
Restaurants...|        Affton|               null|      0|38.5651648
| -90.3210868|Tsevi's Pub And G...|      63123|          19|  3.0|
MO|
| 2312 Dickerson Pike|[,, u'none',,,,,,,...|bBDDEgkFA1Otx9Lfe...|Ice C
ream & Froze...|     Nashville|[6:0-16:0, 0:0-0:...|      1|36.208102
4| -86.7681696|      Sonic Drive-In|      37207|          10|  1.5|
TN|
+--------------------+--------------------+--------------------+-----
---------------+-------------+--------------------+-------+---------
-+------------+--------------------+----------+-----------+-----+--
---+
only showing top 10 rows
```

Overview of Data

Display the number of rows and columns in our dataset.

In [10]:
```
columns=len(business_data.columns)
rows=business_data.count()
print('Number of columns in Business table:  '+str(columns))
print('Number of rows in Business table:  '+str(rows))
```

▸ Spark Job Progress

```
Number of columns in Business table:  14
Number of rows in Business table:  150346
```

Display the DataFrame schema below.

In [11]:
```

```

```
business_data.printSchema()
```

```
root
 |-- address: string (nullable = true)
 |-- attributes: struct (nullable = true)
 |    |-- AcceptsInsurance: string (nullable = true)
 |    |-- AgesAllowed: string (nullable = true)
 |    |-- Alcohol: string (nullable = true)
 |    |-- Ambience: string (nullable = true)
 |    |-- BYOB: string (nullable = true)
 |    |-- BYOBCorkage: string (nullable = true)
 |    |-- BestNights: string (nullable = true)
 |    |-- BikeParking: string (nullable = true)
 |    |-- BusinessAcceptsBitcoin: string (nullable = true)
 |    |-- BusinessAcceptsCreditCards: string (nullable = true)
 |    |-- BusinessParking: string (nullable = true)
 |    |-- ByAppointmentOnly: string (nullable = true)
 |    |-- Caters: string (nullable = true)
 |    |-- CoatCheck: string (nullable = true)
 |    |-- Corkage: string (nullable = true)
 |    |-- DietaryRestrictions: string (nullable = true)
 |    |-- DogsAllowed: string (nullable = true)
 |    |-- DriveThru: string (nullable = true)
 |    |-- GoodForDancing: string (nullable = true)
 |    |-- GoodForKids: string (nullable = true)
 |    |-- GoodForMeal: string (nullable = true)
 |    |-- HairSpecializesIn: string (nullable = true)
 |    |-- HappyHour: string (nullable = true)
 |    |-- HasTV: string (nullable = true)
 |    |-- Music: string (nullable = true)
 |    |-- NoiseLevel: string (nullable = true)
 |    |-- Open24Hours: string (nullable = true)
 |    |-- OutdoorSeating: string (nullable = true)
 |    |-- RestaurantsAttire: string (nullable = true)
 |    |-- RestaurantsCounterService: string (nullable = true)
 |    |-- RestaurantsDelivery: string (nullable = true)
 |    |-- RestaurantsGoodForGroups: string (nullable = true)
 |    |-- RestaurantsPriceRange2: string (nullable = true)
 |    |-- RestaurantsReservations: string (nullable = true)
 |    |-- RestaurantsTableService: string (nullable = true)
 |    |-- RestaurantsTakeOut: string (nullable = true)
 |    |-- Smoking: string (nullable = true)
 |    |-- WheelchairAccessible: string (nullable = true)
 |    |-- WiFi: string (nullable = true)
 |-- business_id: string (nullable = true)
 |-- categories: string (nullable = true)
 |-- city: string (nullable = true)
 |-- hours: struct (nullable = true)
 |    |-- Friday: string (nullable = true)
```

```
 |    |-- Monday: string (nullable = true)
 |    |-- Saturday: string (nullable = true)
 |    |-- Sunday: string (nullable = true)
 |    |-- Thursday: string (nullable = true)
 |    |-- Tuesday: string (nullable = true)
 |    |-- Wednesday: string (nullable = true)
 |-- is_open: long (nullable = true)
 |-- latitude: double (nullable = true)
 |-- longitude: double (nullable = true)
 |-- name: string (nullable = true)
 |-- postal_code: string (nullable = true)
 |-- review_count: long (nullable = true)
 |-- stars: double (nullable = true)
 |-- state: string (nullable = true)
```

Display the first 5 rows with the following columns:

- business_id
- name
- city
- state
- categories

In [12]:
```
business_data.select(['business_id','name','city','state','categories'
```

▸ Spark Job Progress

```
+--------------------+--------------------+------------+-----+------
--------------+
|         business_id|                name|        city|state|
categories|
+--------------------+--------------------+------------+-----+------
--------------+
|Pns2l4eNsfO8kk83d...|Abby Rappoport, L...|Santa Barbara|   CA|Doctor
s, Traditio...|
|mpf3x-BjTdTEA3yCZ...|       The UPS Store|       Affton|   MO|Shippi
ng Centers,...|
|tUFrWirKiKi_TAnsV...|              Target|       Tucson|   AZ|Depart
ment Stores...|
|MTSW4McQd7CbVtyjq...|  St Honore Pastries| Philadelphia|   PA|Restau
rants, Food...|
|mWMc6_wTdE0EUBKIG...|Perkiomen Valley ...|   Green Lane|   PA|Brewpu
bs, Breweri...|
+--------------------+--------------------+------------+-----+------
--------------+
only showing top 5 rows
```

Analyzing Categories

Let's now answer this question: **how many unique categories are represented in this dataset?**

Essentially, we have the categories per business as a list - this is useful to quickly see what each business might be represented as but it is difficult to easily answer questions such as:

- How many businesses are categorized as `Active Life` , for instance
- What are the top 20 most popular categories available?

Association Table

We need to "break out" these categories from the business ids? One common approach to take is to build an association table mapping a single business id multiple times to each distinct category.

For instance, given the following:

business_id	categories
abcd123	a,b,c

We would like to derive something like:

business_id	category
abcd123	a
abcd123	b
abcd123	c

What this does is allow us to then perform a myriad of rollups and other analysis on this association table which can aid us in answering the questions asked above.

Implement the code necessary to derive the table described from your original yelp dataframe.

In [13]:
```
from pyspark.sql.functions import split, explode
```

In [14]:
```
business_data.select('business_id','categories').show(5) #to display h
```

Spark Job Progress

```
+--------------------+--------------------+
|         business_id|          categories|
+--------------------+--------------------+
|Pns2l4eNsfO8kk83d...|Doctors, Traditio...|
|mpf3x-BjTdTEA3yCZ...|Shipping Centers,...|
|tUFrWirKiKi_TAnsV...|Department Stores...|
|MTSW4McQd7CbVtyjq...|Restaurants, Food...|
|mWMc6_wTdE0EUBKIG...|Brewpubs, Breweri...|
+--------------------+--------------------+
only showing top 5 rows
```

Display the first 5 rows of your association table below.

In [15]:
```
#we will split the categories coloumn so that it maintains atomicity w

asso_tab = business_data.select(business_data.business_id, explode(spl
                                        .alias('category'))

asso_tab.show(5)
```

Spark Job Progress

```
+--------------------+--------------------+
|         business_id|            category|
+--------------------+--------------------+
|Pns2l4eNsfO8kk83d...|             Doctors|
|Pns2l4eNsfO8kk83d...|Traditional Chine...|
|Pns2l4eNsfO8kk83d...|Naturopathic/Holi...|
|Pns2l4eNsfO8kk83d...|         Acupuncture|
|Pns2l4eNsfO8kk83d...|    Health & Medical|
+--------------------+--------------------+
only showing top 5 rows
```

Total Unique Categories

Finally, we are ready to answer the question: **what is the total number of unique categories available?**

Below, implement the code necessary to calculate this figure.

In [16]:
```
asso_tab.select('category').distinct().count()
```

▸ Spark Job Progress

1311

Top Categories By Business

Now let's find the top categories in this dataset by rolling up categories.

Counts of Businesses / Category

So now, let's unroll our distinct count a bit and display the per count value of businesses per category.

The expected output should be:

category	count
a	15
b	2
c	45

Or something to that effect.

In [17]:
```
asso_tab.groupby('category').count().show(20)
```

▸ Spark Job Progress

```
+--------------------+-----+
|            category|count|
+--------------------+-----+
|      Paddleboarding|   98|
|      Dermatologists|  336|
|         Hobby Shops|  552|
|          Bubble Tea|  477|
|             Embassy|    3|
|             Tanning|  667|
|            Handyman|  356|
|      Aerial Fitness|   19|
|             Falafel|  103|
|        Summer Camps|  232|
|       Outlet Stores|  182|
|     Clothing Rental|   37|
|      Sporting Goods| 1662|
|     Cooking Schools|   76|
|  Lactation Services|   27|
|Ski & Snowboard S...|   40|
|             Museums|  413|
|              Doulas|   31|
|                Food|27781|
|         Halotherapy|   23|
+--------------------+-----+
only showing top 20 rows
```

Bar Chart of Top Categories

With this data available, let us now build a barchart of the top 20 categories.

HINT: don't forget about the matplotlib magic!

```
%matplot plt
```

If you want, you can also use seaborn library

In [18]:
```
temp = asso_tab.groupby('category')\
                    .count()\
                    .orderBy(['count'], ascending = False)
temp.show(5)
```

▸ Spark Job Progress

```
+-------------+-----+
|     category|count|
+-------------+-----+
|  Restaurants|52268|
|         Food|27781|
|     Shopping|24395|
|Home Services|14356|
|Beauty & Spas|14292|
+-------------+-----+
only showing top 5 rows
```

In [19]:
```
df=temp.toPandas()\
    .head(20)\
    .sort_values(ascending = True, by='count')

df.head(20)
```

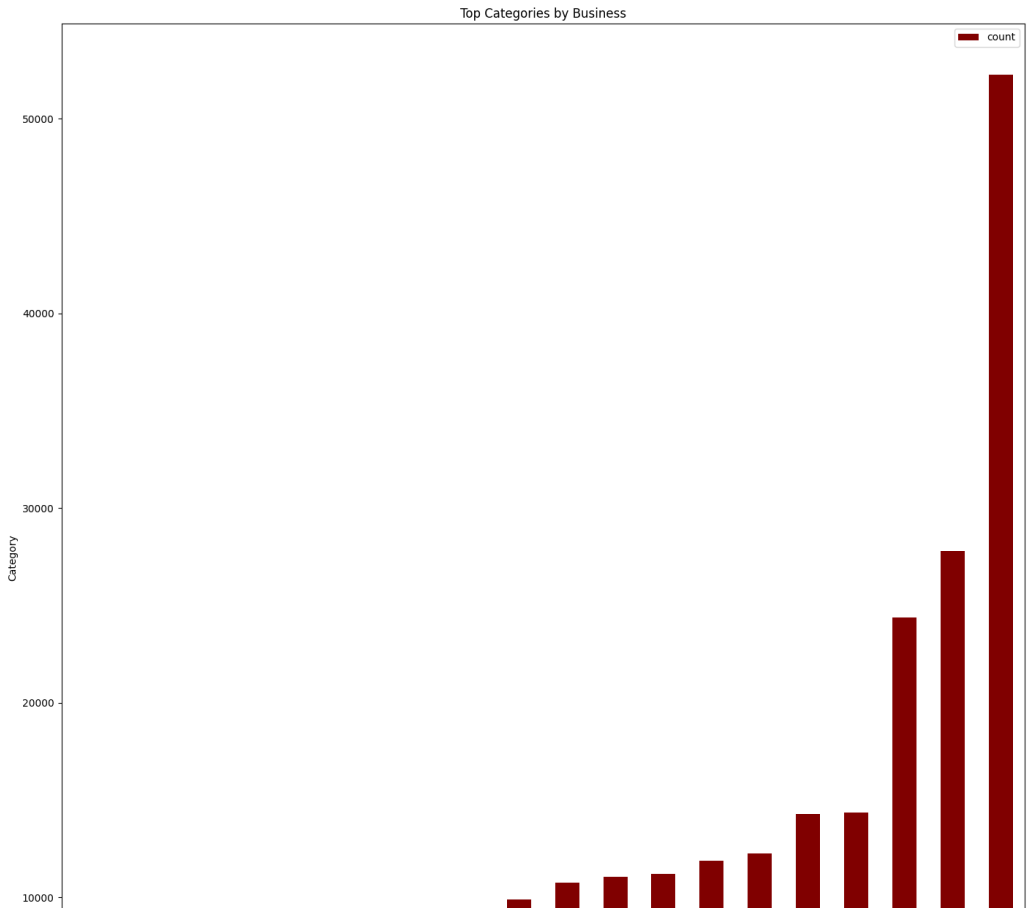
Spark Job Progress

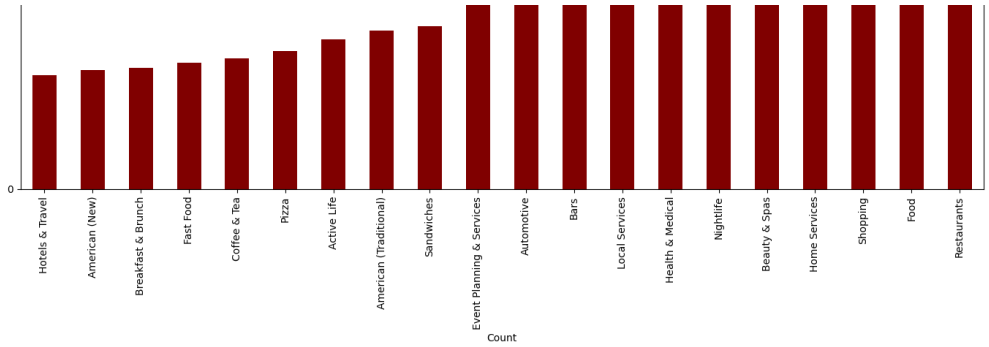
	category	count
19	Hotels & Travel	5857
18	American (New)	6097
17	Breakfast & Brunch	6239
16	Fast Food	6472
15	Coffee & Tea	6703
14	Pizza	7093
13	Active Life	7687
12	American (Traditional)	8139
11	Sandwiches	8366
10	Event Planning & Services	9895
9	Automotive	10773
8	Bars	11065
7	Local Services	11198
6	Health & Medical	11890
5	Nightlife	12281
4	Beauty & Spas	14292
3	Home Services	14356
2	Shopping	24395
1	Food	27781
0	Restaurants	52268

In [20]:
```

```

```
temp.toPandas()\
    .head(20)\
    .sort_values(ascending = True, by='count')\
    .plot.bar(y = 'count',
              x = 'category',
              rot=90,
              title = 'Top Categories by Business',
              legend = True,
              figsize=(17,19),
              color = 'maroon')\
    .set(xlabel="Count", ylabel = 'Category')\


%matplot plt
```

▸ Spark Job Progress





Loading User Data

Begin by loading the user data set from S3 and printing schema to determine what data is available. `s3://cis9760-yelpdataset/yelp-light/*review.json`

In [21]:
```
review_data = spark.read.json('s3://yelp-reviews-dataset/yelp_academic
user_data = spark.read.json('s3://yelp-reviews-dataset/yelp_academic_d
```

▸ Spark Job Progress

In [22]:
```
review_data.printSchema()
```

```
root
 |-- business_id: string (nullable = true)
 |-- cool: long (nullable = true)
 |-- date: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- review_id: string (nullable = true)
 |-- stars: double (nullable = true)
 |-- text: string (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
```

Let's begin by listing the `business_id` and `stars` columns together for the user reviews data.

In [23]:
```
user_data.printSchema()
```

```
root
 |-- average_stars: double (nullable = true)
 |-- compliment_cool: long (nullable = true)
 |-- compliment_cute: long (nullable = true)
 |-- compliment_funny: long (nullable = true)
 |-- compliment_hot: long (nullable = true)
 |-- compliment_list: long (nullable = true)
 |-- compliment_more: long (nullable = true)
 |-- compliment_note: long (nullable = true)
 |-- compliment_photos: long (nullable = true)
 |-- compliment_plain: long (nullable = true)
 |-- compliment_profile: long (nullable = true)
 |-- compliment_writer: long (nullable = true)
 |-- cool: long (nullable = true)
 |-- elite: string (nullable = true)
 |-- fans: long (nullable = true)
 |-- friends: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- name: string (nullable = true)
 |-- review_count: long (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
 |-- yelping_since: string (nullable = true)
```

In [24]:
```
review_data.createOrReplaceTempView("stars")
output = spark.sql('select business_id, stars from stars')
output.show(5)
```

▸ Spark Job Progress

```
+--------------------+-----+
|         business_id|stars|
+--------------------+-----+
|XQfwVwDr-v0ZS3_Cb...|  3.0|
|7ATYjTIgM3jUlt4UM...|  5.0|
|YjUWPpI6HXG530lwP...|  3.0|
|kxX2SOes4o-D3ZQBk...|  5.0|
|e4Vwtrqf-wpJfwesg...|  4.0|
+--------------------+-----+
only showing top 5 rows
```

Now, let's aggregate along the `stars` column to get a resultant dataframe that displays *average stars* per business as accumulated by users who **took the time to submit a written review**.

In [25]:
```
avg_aggstars = spark.sql('select business_id, avg(stars) as avgstars f
avg_aggstars.createOrReplaceTempView("reviews")
avg_aggstars.show(5)
```

▸ Spark Job Progress

```
+--------------------+------------------+
|         business_id|          avgstars|
+--------------------+------------------+
|zJErbOQMKX-MwHs_u...|2.9279279279279278|
|RZ-FNTXvqHKngyLGD...|2.8823529411764706|
|HSzSGdcNaU7heQe0N...|3.3333333333333335|
|skW4boArIApRw9DXK...|2.3947368421052633|
|I0053JmJ5DEFUWSJ8...|2.3956043956043955|
+--------------------+------------------+
only showing top 5 rows
```

Now the fun part - let's join our two dataframes (reviews and business data) by `business_id` .

In [26]:
```
#output = spark.sql(
#'''
#SELECT rev.*, bus.stars, bus.name, bus.city, bus.state
 #                       from business as bus
  #                      left join reviews as rev
   #                     on bus.business_id = rev.business_id''')
#output.createOrReplaceTempView("joinedOutput")
```

In [27]:
```
reviews_only= avg_aggstars.select("business_id","avgstars")
business_only= business_data.select("business_id","name","city","state
reviews_business= reviews_only.join(business_only, reviews_only.busine
reviews_business.select("name","city","state","stars","avgstars").show
```

▸ Spark Job Progress

```
+--------------------+-----------+-----+-----+------------------+
|                name|       city|state|stars|          avgstars|
+--------------------+-----------+-----+-----+------------------+
|Philadelphia Marr...|Philadelphia|   PA|  3.0|2.9279279279279278|
|Gaetano's of West...| West Berlin|   NJ|  3.0|2.8823529411764706|
|Gillane's Bar & G...|    Ardmore|   PA|  3.0|3.3333333333333335|
|Champps Penn's La...|Philadelphia|   PA|  2.5|2.3947368421052633|
|Golden Corral Buf...|     Tucson|   AZ|  2.5|2.3956043956043955|
|  Swiss Watch Center|      Tampa|   FL|  3.5| 3.357142857142857|
|  NJ Weedman's Joint|    Trenton|   NJ|  4.0| 4.232558139534884|
|       A Able Movers|     Tucson|   AZ|  2.0|             1.875|
|       Numchok Wilai|   Edmonton|   AB|  4.5|               4.3|
|             Safeway|     Sparks|   NV|  3.0|2.8117647058823527|
+--------------------+-----------+-----+-----+------------------+
only showing top 10 rows
```

In [28]:
```
reviews_business.createOrReplaceTempView("reviews_business")
```

Let's see a few of these:

Compute a new dataframe that calculates what we will call the *skew* (for lack of a better word) between the avg stars accumulated from written reviews and the *actual* star rating of a business (ie: the average of stars given by reviewers who wrote an actual review **and** reviewers who just provided a star rating).

The formula you can use is something like:

```
(row['avg(stars)'] - row['stars']) / row['stars']
```

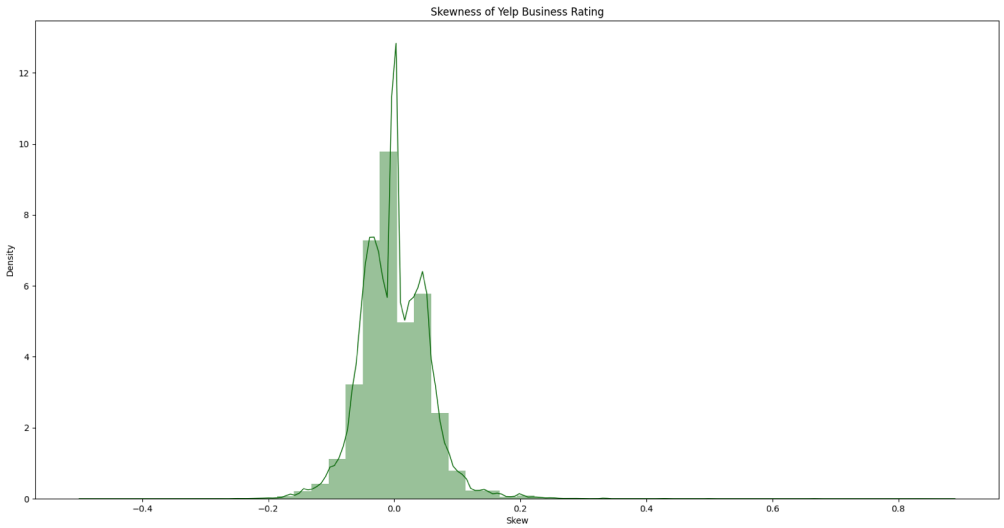
If the **skew** is negative, we can interpret that to be: reviewers who left a written response were more dissatisfied than normal. If **skew** is positive, we can interpret that to be: reviewers who left a written response were more satisfied than normal.

In [29]:
```
skew_df = spark.sql("select (avgstars-stars)/stars as skew from reviews
```

And finally, graph it!

In [30]:
```
skew_pdf = skew_df.toPandas()

plt.figure(figsize=(20,10))
sns.distplot(skew_pdf['skew'],
             hist=True,
             kde=True,
             bins=50,
             color = 'darkgreen',
             kde_kws={'linewidth':1}).\
    set(title = "Skewness of Yelp Business Rating",
        xlabel = 'Skew')

#plt.xlabel('skew')
#fig.gca()
#fig.set_title=("skewness")
%matplot plt
```

Spark Job Progress



So, do Yelp (written) Reviews skew negative? Does this analysis actually prove anything? Expound on implications / interpretations of this graph.

IMPLICATIONS

Here, we can see that the graph is positively skewed to a slight degree and the graph has a longer tail on the right , this means that there is a greater number of people that have given a negative written review. These could be complains from unsatisfied customers about the restaurant service or ambience or price etc.

Should the Elite be Trusted?

How accurate or close are the ratings of an "elite" user (check Users table schema) vs the actual business rating? `s3://cis9760-yelpdataset/yelp-light/*user.json`

Feel free to use any and all methodologies at your disposal. You must render one visualization in your analysis and interpret your findings.

In [31]:
```
user_data.printSchema()
```

```
root
 |-- average_stars: double (nullable = true)
 |-- compliment_cool: long (nullable = true)
 |-- compliment_cute: long (nullable = true)
 |-- compliment_funny: long (nullable = true)
 |-- compliment_hot: long (nullable = true)
 |-- compliment_list: long (nullable = true)
 |-- compliment_more: long (nullable = true)
 |-- compliment_note: long (nullable = true)
 |-- compliment_photos: long (nullable = true)
 |-- compliment_plain: long (nullable = true)
 |-- compliment_profile: long (nullable = true)
 |-- compliment_writer: long (nullable = true)
 |-- cool: long (nullable = true)
 |-- elite: string (nullable = true)
 |-- fans: long (nullable = true)
 |-- friends: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- name: string (nullable = true)
 |-- review_count: long (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
 |-- yelping_since: string (nullable = true)
```

In [32]:
```
elite_join = review_data.join(avg_aggstars, review_data.business_id ==
elite_join.show(5)
```

▸ Spark Job Progress

```
+--------------------+----+--------------------+-----+--------------
----+-----+--------------------+------+--------------------+---------
----------+----------------+
|         business_id|cool|                date|funny|           revie
w_id|stars|                text|useful|             user_id|
business_id|        avgstars|
+--------------------+----+--------------------+-----+--------------
----+-----+--------------------+------+--------------------+---------
----------+----------------+
|--gJkxbsiSIwsQKbi...|   3|2018-11-10 13:11:10|    1|4ww8UA2ZTwGmilBJ
N...|  4.0|I followed my lon...|     5|fen9BWC39ul9SJZfQ...|--gJkxbsi
SIwsQKbi...|4.833333333333333|
|--gJkxbsiSIwsQKbi...|   0|2019-01-10 02:51:06|    0|4SPOoOr1ZZWGm-1O
m...|  5.0|Amber is the best...|     0|L4q5nCwMaHhXCeSJz...|--gJkxbsi
SIwsQKbi...|4.833333333333333|
|--gJkxbsiSIwsQKbi...|   0|2019-04-18 18:34:40|    0|CSlZvn9wPq6kIahb
c...|  5.0|Gina Marotti in L...|     0|XGmxkw2Zbunt5u2ZD...|--gJkxbsi
SIwsQKbi...|4.833333333333333|
|--gJkxbsiSIwsQKbi...|   1|2018-12-10 21:36:47|    0|gMGm7d8b8pXwi1Bz
l...|  5.0|Irnela Sunj ("Nel...|     1|8txdIkqyhrSxZ4RMY...|--gJkxbsi
SIwsQKbi...|4.833333333333333|
|--gJkxbsiSIwsQKbi...|   0|2019-01-10 02:49:46|    0|SL93b9QthJJGb2LA
w...|  5.0|I saw Amber the o...|     0|TOmK6TTeQMfktNR__...|--gJkxbsi
SIwsQKbi...|4.833333333333333|
+--------------------+----+--------------------+-----+--------------
----+-----+--------------------+------+--------------------+---------
----------+----------------+
only showing top 5 rows
```

In [33]:
```
#now join this with user data

eluser_review=user_data.join(elite_join, on="user_id", how="inner")

elite_laundey=eluser_review.select('avgstars','stars', 'elite','review
elite_laundey.show(5)

#eluser_review.show(5)
```

▸ Spark Job Progress

```
+------------------+-----+-----+------------+
|          avgstars|stars|elite|review_count|
+------------------+-----+-----+------------+
|3.8430717863105177|  5.0|     |          11|
| 4.625498007968128|  5.0|     |          11|
|3.8030821917808217|  2.0|     |          11|
|              3.05|  1.0|     |           9|
|             3.652|  2.0|     |          50|
+------------------+-----+-----+------------+
only showing top 5 rows
```

In [34]:
```
import pyspark.sql.functions as F
review_skew = elite_laundey.withColumn("skew", F.round((F.col('avgstar

review_skew.show(5)
```

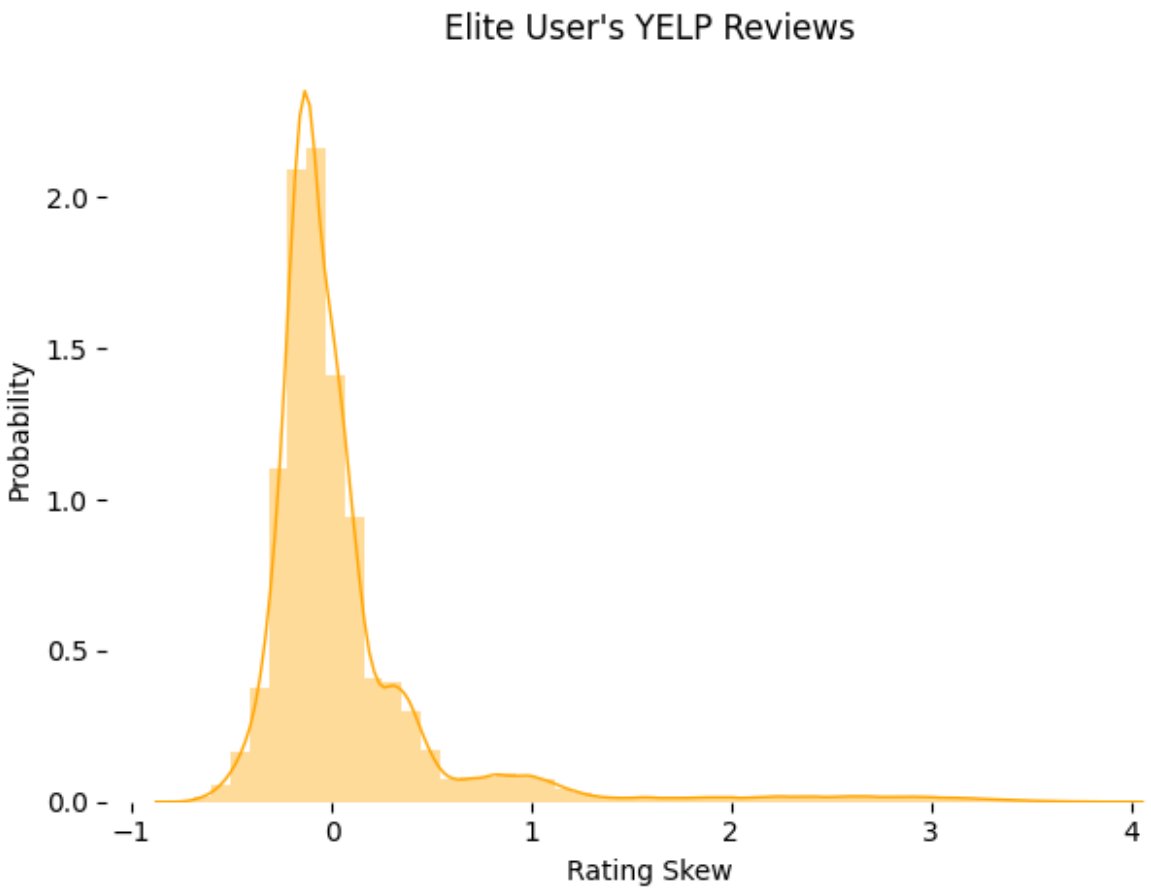
▸ Spark Job Progress

```
+------------------+-----+-----+------------+-----+
|          avgstars|stars|elite|review_count| skew|
+------------------+-----+-----+------------+-----+
|3.8430717863105177|  5.0|     |          11|-0.23|
|3.8030821917808217|  2.0|     |          11|  0.9|
| 4.625498007968128|  5.0|     |          11|-0.07|
|              3.05|  1.0|     |           9| 2.05|
|             3.652|  2.0|     |          50| 0.83|
+------------------+-----+-----+------------+-----+
only showing top 5 rows
```

In [35]:

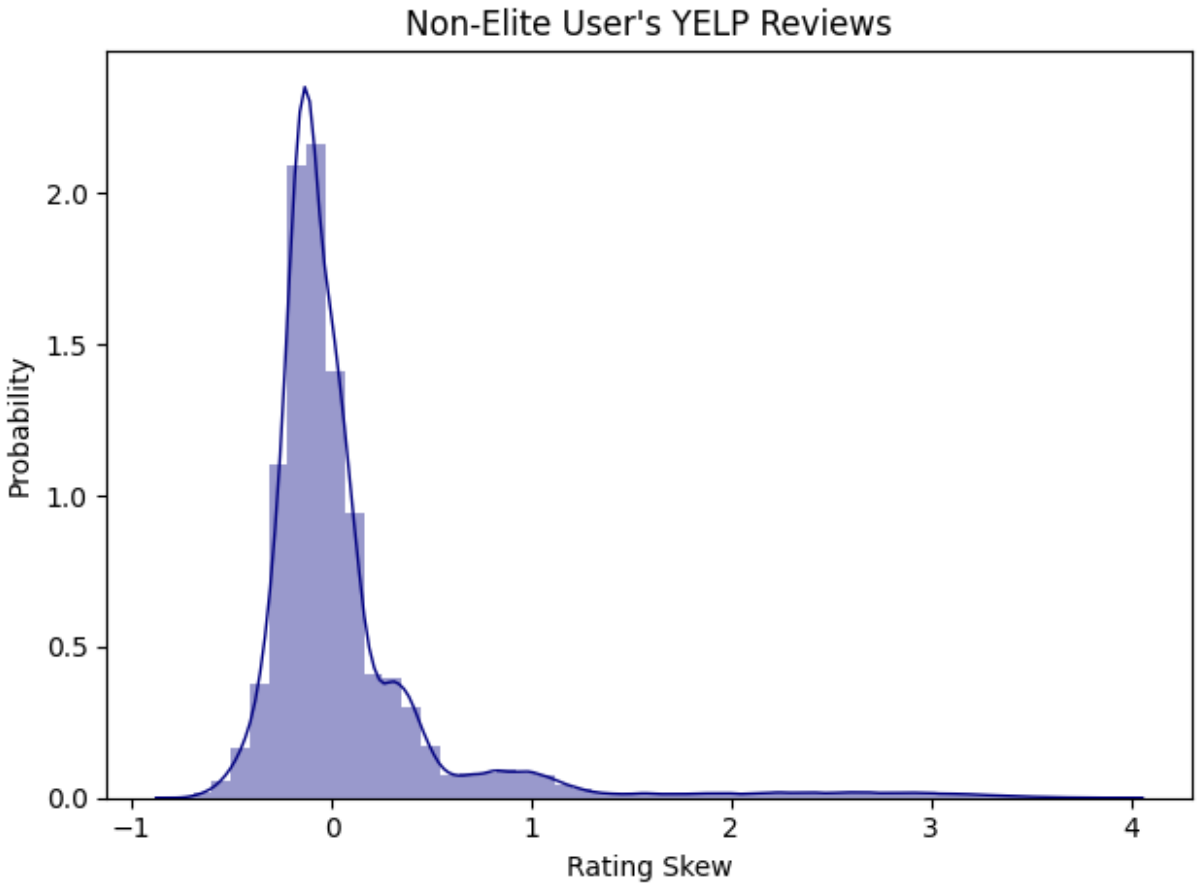
```
elite_review_skew=review_skew.filter(F.col('elite')!='')

result_pdf = elite_review_skew.toPandas()
plt.figure()
sns.distplot(result_pdf['skew'],
             hist=True,
             kde=True,
             bins=50,
             color = 'orange',
             kde_kws={'linewidth':1})

plt.title('Elite User\'s YELP Reviews')
plt.xlabel('Rating Skew')
plt.ylabel('Probability')
plt.tight_layout()
plt.box(False)

%matplot plt
```

▸ Spark Job Progress



In [36]:
```
nonelite_review_skew=review_skew.filter(F.col('elite')=='')

result_pdf = elite_review_skew.toPandas()
plt.figure()
sns.distplot(result_pdf['skew'],
             hist=True,
             kde=True,
             bins=50,
             color = 'navy',
             kde_kws={'linewidth':1})

plt.title('Non-Elite User\'s YELP Reviews')
plt.xlabel('Rating Skew')
plt.ylabel('Probability')
plt.tight_layout()
plt.box(True)

%matplot plt
```

▸ Spark Job Progress



From both the above graphs , it can be clearly gathered that by looking at the skewness of the elite vs non-elite there is'nt significant difference between the two, so it is safe to say that elite users don't have a significant impact on ratings.