All the stuff you need to know (Math)_Class3-Completed

February 18, 2021

[10]:
```
import numpy as np
import pandas as pd
from scipy import stats

from plotnine import *
```

1 Together

1.1 Distributions

A distribution is a curve (although sometimes it's pretty straight) that shows how common or uncommon different values are. For example, this is a normal distribution with mean = 0 and standard deviation = 1. Which values are relatively common under this distribution? Uncommon?

What about this one?

1.2 Function Optimization

In our lecture we talked about derivatives, and that we often want to *minimize* functions when doing Data Scienc e + Machine Learning. While I won't dive into ALL the calculus now (if this kind of thing excites you, you should take CPSC 393!) I want to talk about some of the ideas behind minimizing functions.

or

or

2 In Your Groups

2.1 Logarithms

Use your new pandas skills to add a column, `logX` to the dataframe DF that contains the log (`np.log()`) of the column `X`.

Then run the pre-written code (no need to change anything) to plot the log function.

What range of values can the log() function take in? What range of values can the log function spit out? What happens to values < 1 when you log() them? What about values > 1?

2.1.1 Answer

log() can take in values from 0 to infinity, and spits out values between -inf and inf.

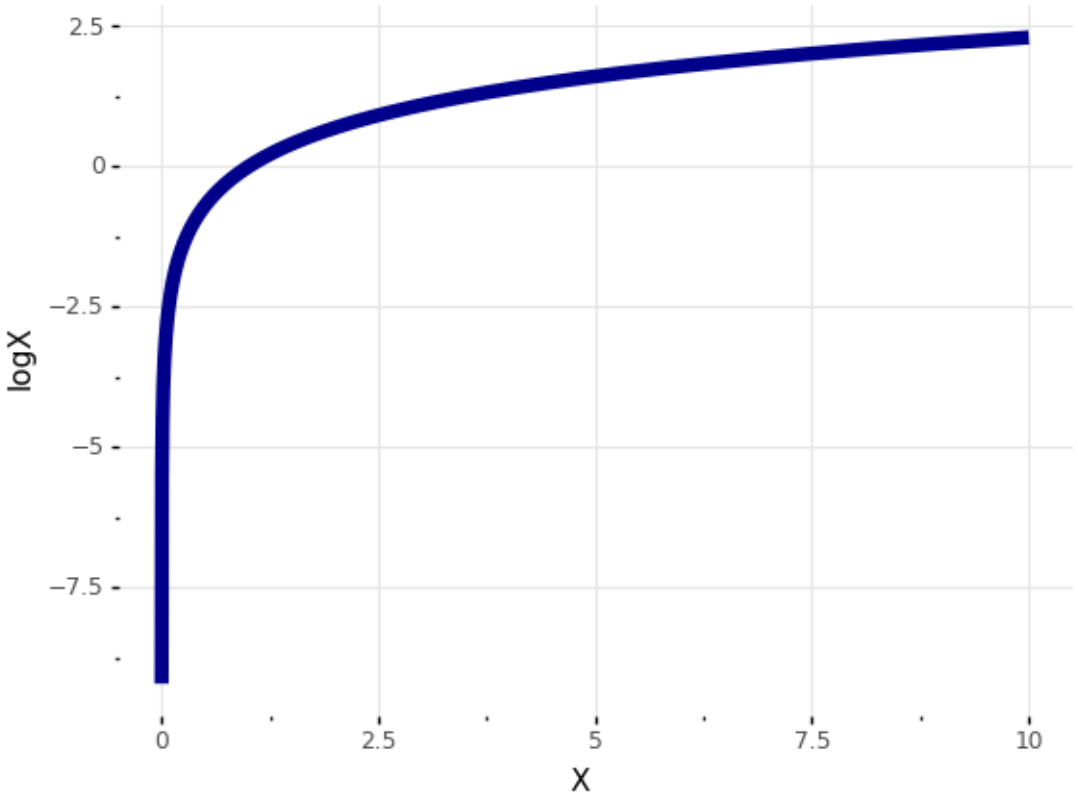
The log of values greater than 1 are positive, the log of values less than 1 are negative.

[11]:
```
DF = pd.DataFrame({"X": np.linspace(0.0001,10, 10000)})



### YOUR CODE HERE ###

DF["logX"] = np.log(DF.X)

DF.head() ### /YOUR CODE HERE ###
```

[11]:

	X	logX
0	0.0001	-9.210340
1	0.0011	-6.812363
2	0.0021	-6.165732
3	0.0031	-5.776266
4	0.0041	-5.496680

[12]:
```
# DONT CHANGE, JUST RUN

(ggplot(DF, aes(x = "X", y = "logX")) +
 geom_line(color = "darkblue", size = 3) +
 theme_bw() +
theme(panel_border = element_blank(),
     panel_grid_minor = element_blank()))
```



[12]: <ggplot: (8784881921362)>

2.2 Data Types

In your lecture, you learned about different types of data/variables we could have. Go to our course github and click on the *Data* folder. Get the raw URL for the **Beyonce_data.csv** dataset and load it in using `pd.read_csv()`. Store your dataframe in the variable `bey`, and print the head of the dataframe.

[13]:
```
bey = pd.read_csv("https://raw.githubusercontent.com/cmparlettpelleriti/
↪CPSC392ParlettPelleriti/master/Data/Beyonce_data.csv")

bey.head()
```

[13]:

	Unnamed: 0	artist_name	danceability	energy	key	loudness	mode	\
0	1	Beyoncé	0.386	0.28800	1	-18.513	1	
1	2	Beyoncé	0.484	0.36300	5	-8.094	0	
2	3	Beyoncé	0.537	0.24700	2	-17.750	1	
3	4	Beyoncé	0.672	0.69600	4	-6.693	0	
4	5	Beyoncé	0.000	0.00515	9	-22.612	0	

```
   speechiness  acousticness  instrumentalness  liveness  valence  \
0       0.0602         0.533           0.01670    0.1410    0.399
1       0.0368         0.645           0.00000    0.1250    0.201
2       0.0793         0.199           0.00001    0.4230    0.170
3       0.1770         0.200           0.02750    0.0736    0.642
4       0.0000         0.524           0.95000    0.1140    0.000

   duration_ms                    track_name
0        43850   balance (mufasa interlude)
1       226479                        BIGGER
2        46566  the stars (mufasa interlude)
3       162353            FIND YOUR WAY BACK
4        13853   uncle scar (scar interlude)
```

What types are all the variables?

- Categorical:
  - nominal: `artist_name`,
  - ordinal: `key?`
  - interval: `key?`
- Numeric: `danceability`, `energy`, `speechiness`, `acousticness`, `instrumentalness`, `liveness`, `valence`, `duration`
- Boolean: `mode`
- Text: `track_name`

If a variable is Categorical, how do you decide if it's nominal, ordinal, or interval? Give an example of each.

2.2.1 Answer

Nominal variables have no order (like Red, Green, Blue), Ordinal variables DO have a specific order (like small, medium, and large drinks) but the difference between each successive category may not be the same. Interval variables have an order AND have the same difference between each successive category (for example, age is technically interval because we do not say we are 48.234234239482934 years old, we just say we are 48).

2.3 Probabilities and Conditional Probabilities

Remember that in general, probabilities are

Given this information, and the dataframe `voters`, calculate the probability of:

- being a registered voter
- being a vegetarian AND a registered voter

[14]:
```
registered = ['no', 'yes', 'yes', 'no', 'no', 'yes', 'yes', 'yes', 'no', 'no',
       'yes', 'no', 'no', 'no', 'yes', 'yes', 'no', 'no', 'no', 'yes']

diet = ['Meat_Eater', 'Vegetarian', 'Vegetarian', 'Vegan', 'Vegetarian',
       'Vegetarian', 'Vegetarian', 'Vegetarian', 'Meat_Eater', 'Vegan',
```

```
       'Vegan', 'Vegetarian', 'Meat_Eater', 'Meat_Eater', 'Vegan',
       'Vegan', 'Vegetarian', 'Meat_Eater', 'Meat_Eater', 'Vegetarian']

voters = pd.DataFrame({"RegisteredToVote": registered, "Diet": diet})

voters

# P(reg)

sum(voters.RegisteredToVote == "yes")/voters.shape[0]

# P(veg and reg)
sum((voters.RegisteredToVote == "yes") & (voters.Diet == "Vegetarian"))/voters.
 ↪shape[0]
```

[14]: 0.3

Conditional probabilities are just probabilities where the total events are *restricted* by some kind of information.

For example: P(**Vegetarian | registered to vote**) (in words we'd say this as "the Probability of being Vegetarian *given* that you are registered to vote") means that we want to know the probability of being Vegetarian when ONLY looking at registered voters. This means that the denominator of our probability will only count registered voters.

There are 9 registered voters in our data frame, and out of those 9, 6 are Vegetarian. So P(**Vegetarian | registered to vote**) $= 6/9$.

Using the data frame `booksRead` below which indicates the responses from 25 people about which books they had read in the past 5 years, calculate (using code or by hand) the following probabilites: - P(read Tale of Two Cities) - P(read the Bible) - P(read What to Expect When You're Expecting | read Tale of Two Cities) - P(read What to Expect When You're Expecting | read Tale of Two Cities **AND** the Bible) - P(read How to Win Friends and Influence People | did not read LOTR) - P(read LOTR **AND** Tale of Two Cities)

[15]:
```
taleOfTwoCities = ['yes', 'yes', 'yes', 'no', 'yes', 'yes', 'no', 'yes', 'yes',␣
 ↪'no',
       'yes', 'no', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes',
       'yes', 'yes', 'yes', 'yes', 'yes', 'yes']

bible = ['yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes',
       'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes', 'yes',
       'yes', 'yes', 'yes', 'no', 'yes', 'yes', 'yes']

howToWinFriendsAndInfluencePeople = ['yes', 'no', 'no', 'no', 'no', 'no',␣
 ↪'yes', 'yes', 'no', 'no',
       'yes', 'no', 'no', 'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no',
       'no', 'no', 'no', 'yes']
```

```
whatToExpectWhenYoureExpecting = ['yes', 'yes', 'yes', 'no', 'yes', 'yes',␣
↪'yes', 'no', 'no', 'yes',
       'no', 'no', 'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'no', 'no',
       'no', 'yes', 'no', 'yes']

LOTR = ['yes', 'yes', 'no', 'no', 'no', 'no', 'no', 'yes', 'no', 'no',
       'no', 'no', 'no', 'yes', 'yes', 'no', 'no', 'no', 'no', 'no',
       'yes', 'yes', 'no', 'yes', 'no']

booksRead = pd.DataFrame({"taleOfTwoCities": taleOfTwoCities,
                          "bible": bible,
                          "howToWinFriendsAndInfluencePeople":␣
↪howToWinFriendsAndInfluencePeople,
                          "whatToExpectWhenYoureExpecting":␣
↪whatToExpectWhenYoureExpecting,
                          "LOTR": LOTR})

booksRead
```

[15]:

	taleOfTwoCities	bible	howToWinFriendsAndInfluencePeople	\
0	yes	yes	yes	
1	yes	yes	no	
2	yes	yes	no	
3	no	yes	no	
4	yes	yes	no	
5	yes	yes	no	
6	no	yes	yes	
7	yes	yes	yes	
8	yes	yes	no	
9	no	yes	no	
10	yes	yes	yes	
11	no	yes	no	
12	yes	yes	no	
13	yes	yes	no	
14	yes	yes	no	
15	yes	yes	yes	
16	yes	yes	no	
17	yes	yes	no	
18	yes	yes	no	
19	yes	yes	no	
20	yes	yes	no	
21	yes	no	no	
22	yes	yes	no	
23	yes	yes	no	
24	yes	yes	yes	

whatToExpectWhenYoureExpecting	LOTR

```
0                                yes  yes
1                                yes  yes
2                                yes   no
3                                 no   no
4                                yes   no
5                                yes   no
6                                yes   no
7                                 no  yes
8                                 no   no
9                                yes   no
10                                no   no
11                                no   no
12                               yes   no
13                                no  yes
14                                no  yes
15                                no   no
16                                no   no
17                                no   no
18                                no   no
19                                no   no
20                                no  yes
21                                no  yes
22                               yes   no
23                                no  yes
24                               yes   no
```

[16]:
```
# - P(read Tale of Two Cities)
(booksRead.taleOfTwoCities == "yes").mean()

# - P(read the Bible)
(booksRead.bible == "yes").mean()

# - P(read What to Expect When You're Expecting **|** read Tale of Two Cities)
sum((booksRead.whatToExpectWhenYoureExpecting == "yes") &(booksRead.
 ↪taleOfTwoCities == "yes"))/ sum((booksRead.taleOfTwoCities == "yes"))

# - P(read What to Expect When You're Expecting **|** read Tale of Two Cities␣
 ↪**AND** the Bible)
ttB = booksRead.loc[(booksRead.taleOfTwoCities == "yes") & (booksRead.bible ==␣
 ↪"yes")]
(ttB.whatToExpectWhenYoureExpecting == "yes").mean()

# - P(read How to Win Friends and Influence People **|** did not read LOTR)
lotrNo = booksRead.loc[(booksRead.LOTR == "no")]
(lotrNo.howToWinFriendsAndInfluencePeople == "yes").mean()

# - P(read LOTR **AND** Tale of Two Cities)
```

```
((booksRead.LOTR == "yes") & (booksRead.taleOfTwoCities == "yes")).mean()
```

[16]: 0.32

2.4 Odds

Odds are the probability of something happening, divided by the probabilit of it not happening. What are the **Odds** of the following events:

- The odds of Bob scoring a goal during a soccer game if P(Bob scoring a goal during a soccer game) = 0.2
- The odds of flipping a heads on a fair coin if P(head) = 0.5
- The odds of your professor showing up in a Dinosaur costume today if P(professor showing up in a Dinosaur costume) = 0.7
- The odds of NOT winning the lottery if P(winning the lottery) = 0.0000001

If my odds of ordering pizza tonight are 3, what is the probability that I order pizza? If I increase my odds by 10x and my odds are now 30, what is the probability that I order pizza?

[17]:
```
# - The odds of Bob scoring a goal during a soccer game if P(Bob scoring a goal␣
 ↪during a soccer game) = 0.2
0.2/0.8
# - The odds of flipping a heads on a fair coin if P(head) = 0.5
0.5/0.5
# - The odds of your professor showing up in a Dinosaur costume today if␣
 ↪P(professor showing up in a Dinosaur costume) = 0.7
0.7/0.3
# - The odds of NOT winning the lottery if P(winning the lottery) = 0.0000001
(1-0.0000001)/0.0000001

# If my odds of ordering pizza tonight are 3, what is the probability that I␣
 ↪order pizza?
3/4

# If I increase my odds by 10x and my odds are now 30, what is the probability␣
 ↪that I order pizza?
30/31
```

[17]: 0.967741935483871