80 Interview Questions on Python for Data Science

RG Follow

Aug 17, 2020 · 11 min read

· Python — 34 questions

· Pandas — 18 questions

· Visualization — 8 questions

· Data cleaning — 5 questions

· Machine learning — 15 questions

Python — 34 questions

1. How do we create numerical variables in python?

```
pi = 3.14159

diameter = 3
```

2. How do we perform calculations in python?

```
radius = diameter / 2

area = pi * radius * radius
```

3. Give an example of BODMAS in python?

```
(8–3) * (2 — (1 + 1))
```

> *The output is 0*

4. Give examples of list?

> *a = [1, 2, 3] → length of a : 3*
>
> *b = [1, [2, 3]] → length of b : 2*
>
> *c = [] → length of c : 0*
>
> *d = [1, 2, 3][1:] → length of d : 2*

5. How do we interchange the values of two lists?

> *a = [1, 2, 3]*
>
> *b = [3, 2, 1]*
>
> *b,a = a,b*

6. How do we extract values from list?

> *r = [“Mario”, “Bowser”, “Luigi”]*
>
> *r[0] → Mario*
>
> *r[-1] → Luigi*

7. How we create loops in python using list?

> *The following code returns the numbers from a list that are more than the threshold*
>
> *def elementwise_greater_than(L, thresh):*
>
> *res = []*
>
> *for ele in L:*
>
> *if ele > thresh: res.append()*
>
> *return res*
>
> *elementwise_greater_than([1, 2, 3, 4], 2)*

> *The output is [3, 4]*

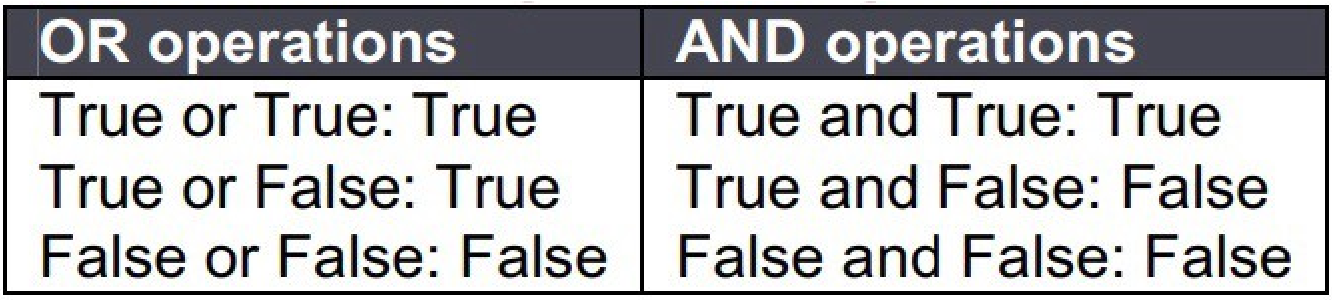
8. Give examples of String?

> *a = “” → length of a : 0*
>
> *b = “it’s ok” → length of b : 7*
>
> *c = ‘it\’s ok’ → length of c : 7*
>
> *d = “””hey””” → length of d : 3*
>
> *e = ‘\n’ → length of e : 1*

9. Give an example of Boolean?

> *A Boolean takes only 2 values: True and False*
>
> *0 < 1: True*
>
> *0 > 1: False*

10. How do we perform operations on Boolean?



OR operations	AND operations
True or True: True True or False: True False or False: False	True and True: True True and False: False False and False: False

11. What are function in python?

> *A function is a block of organized, reusable code that is used to perform a single, related action.*
>
> *def round_to_two_places(num):*
>
> *return round(num, 2)*
>
> *pi = round_to_two_places(3.14159)*

> *The output is 3.14*

12. Calculating remainder in python?

> *91 % 3*
>
> *The output is 1*

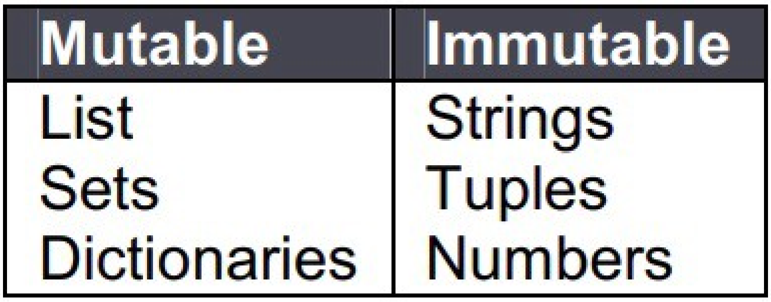
13. Who created python?

> *Python is an interpreted, high-level, general-purpose programming language.*
>
> *Python was created by Guido van Rossum*

14. When was python created?

> *Python was conceived in the late 1980s as a successor to the ABC language*
>
> *The first version was releases in 1991*
>
> *Python 2.0 was released in 2000*
>
> *Python 3.0 was released in 2008*

15. What are the built-in type does python provides?



Mutable	Immutable
List Sets Dictionaries	Strings Tuples Numbers

16. What is lambda in Python?

> *It is a single expression anonymous function used as inline function.*
>
> *x = lambda a : a + 10*
>
> *x(5)*
>
> *The output is 15*

17. What is pass in Python?

> *Pass means, no-operation Python statement.*
>
> *It is a place holder in compound statement, where nothing has to be written.*

18. What is slicing?

> *A mechanism to select a range of items from sequence types like list, tuple, strings etc. is known as slicing.*
>
> *x[1, 2, 3, 4, 5]*
>
> *x[0:2] → [1,2]*
>
> *x[2:] → [3,4,5]*

19. What is negative index in Python?

> *Python sequences can be index in positive and negative numbers.*
>
> *For positive index, 0 is the first index, 1 is the second index and so forth.*
>
> *For negative index, (-1) is the last index and (-2) is the second last index and so forth.*

20. How you can convert a number to a string?

> *In order to convert a number into a string, use the inbuilt function str().*
>
> *If you want a octal or hexadecimal representation, use the inbuilt function oct() or hex().*

21. What is range function?

> *The range() function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.*
>
> *x = range(6)*
>
> *for n in x:*
>
> *print(n)*
>
> *The output is 0, 1, 2, 3, 4, 5*

22. How do you generate random numbers in Python?

> ***Library:*** *import random*
>
> ***Syntax:*** *random.random()*
>
> ***Output:*** *Returns a random floating point number in the range [0,1)*

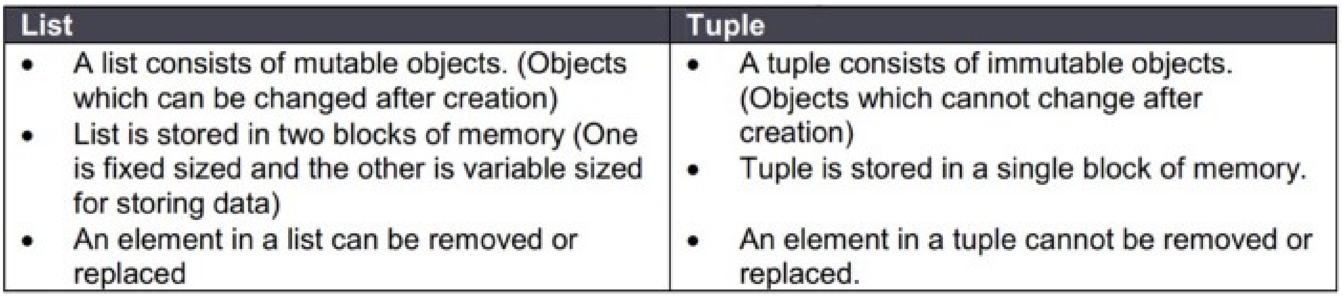
23. What is the difference between / and // operator in Python?

> *// is a Floor Division operator*
>
> *It is used for dividing two operands with the result as quotient showing only digits before the decimal point.*
>
> *10 / 3 = 3.33333*
>
> *10 // 3 = 3*

24. What is the use of the split function in Python?

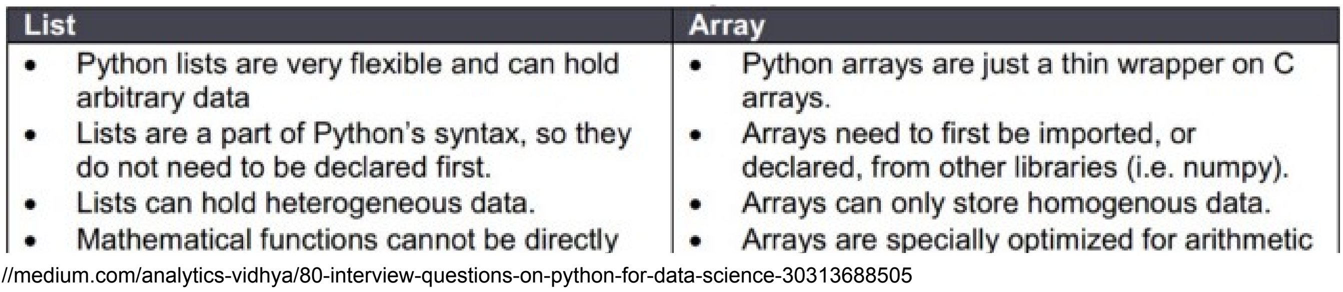
> *The use of the split function in Python is that it breaks a string into shorter strings using the defined separator.*
>
> *It gives a list of all words present in the string.*

25. What is the difference between a list and a tuple?

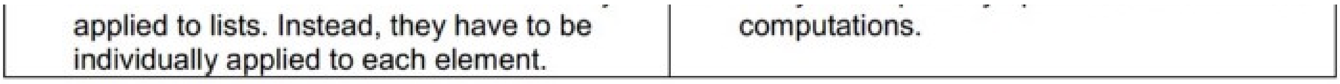


List	Tuple
• A list consists of mutable objects. (Objects which can be changed after creation) • List is stored in two blocks of memory (One is fixed sized and the other is variable sized for storing data) • An element in a list can be removed or replaced	• A tuple consists of immutable objects. (Objects which cannot change after creation) • Tuple is stored in a single block of memory. • An element in a tuple cannot be removed or replaced.

26. What is the difference between an array and a list?



List	Array
• Python lists are very flexible and can hold arbitrary data • Lists are a part of Python's syntax, so they do not need to be declared first. • Lists can hold heterogeneous data. • Mathematical functions cannot be directly	• Python arrays are just a thin wrapper on C arrays. • Arrays need to first be imported, or declared, from other libraries (i.e. numpy). • Arrays can only store homogenous data. • Arrays are specially optimized for arithmetic



applied to lists. Instead, they have to be individually applied to each element.	computations.

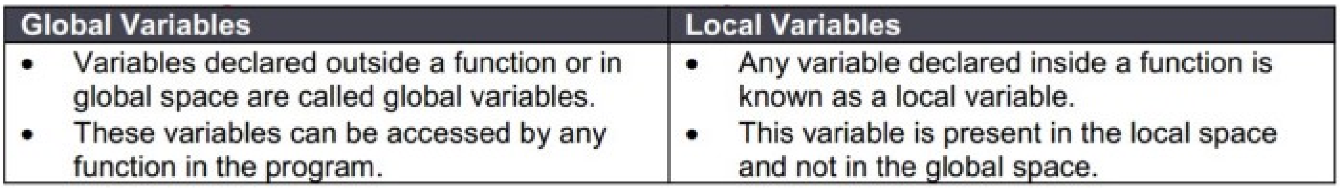
27. How would you convert a list to an array?

> *This is done using numpy.array().*
>
> *This function of the numpy library takes a list as an argument and returns an array that contains all the elements of the list.*

28. What are the advantages of NumPy arrays over Python lists?

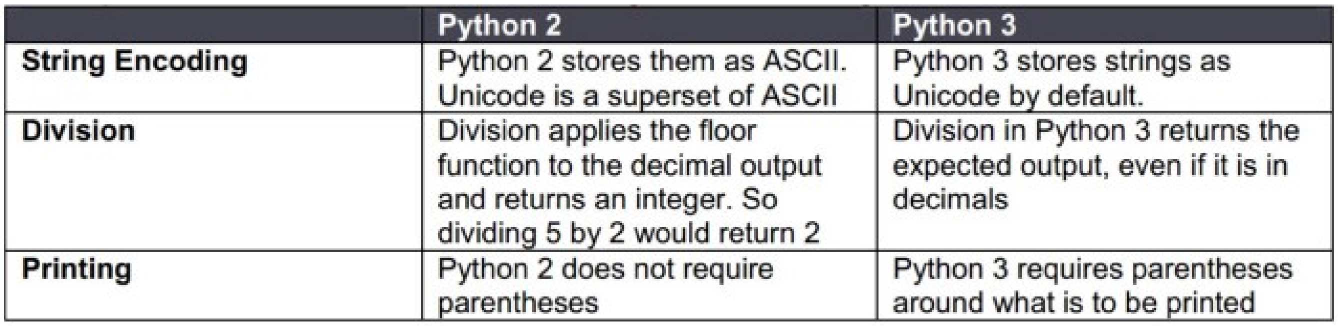
> *NumPy is more convenient.*
>
> *You get a lot of vector and matrix operations, which sometimes allow one to avoid unnecessary work.*
>
> *You get a lot built in functions with NumPy for fast searching, basic statistics, linear algebra, histograms, etc.*

29. What are global and local variables in Python?



Global Variables	Local Variables
• Variables declared outside a function or in global space are called global variables. • These variables can be accessed by any function in the program.	• Any variable declared inside a function is known as a local variable. • This variable is present in the local space and not in the global space.

30. Explain the differences between Python 2 and Python 3?



	Python 2	Python 3
String Encoding	Python 2 stores them as ASCII. Unicode is a superset of ASCII	Python 3 stores strings as Unicode by default.
Division	Division applies the floor function to the decimal output and returns an integer. So dividing 5 by 2 would return 2	Division in Python 3 returns the expected output, even if it is in decimals
Printing	Python 2 does not require parentheses	Python 3 requires parentheses around what is to be printed

31. What is dictionary comprehension in Python?

> *Dictionary comprehension is one way to create a dictionary in Python.*

> *It creates a dictionary by merging two sets of data which are in the form of either lists or arrays.*
>
> *rollNumbers =[122, 233, 353, 456]*
>
> *names = [‘alex’, ‘bob’, ‘can’, ‘don’]*
>
> *NewDictionary={ i:j for (i,j) in zip (rollNumbers,names)}*
>
> *The output is {(122, ‘alex’), (233, ‘bob’), (353, ‘can’), (456, ‘don’)*

32. How would you sort a dictionary in Python?

> ***Dictionary.keys() :*** *Returns only the keys in an arbitrary order.*
>
> ***Dictionary.values() :*** *Returns a list of values.*
>
> ***Dictionary.items() :*** *Returns all of the data as a list of key-value pairs.*
>
> ***Sorted():*** *This method takes one mandatory and two optional arguments*

33. How do you reverse a string in Python?

> *Stringname = ‘python’*
>
> *Stringname[::-1]*
>
> *The output is ‘nohtyp’*

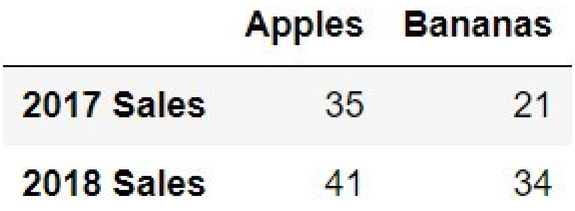
34. How do you check if a Python string contains another string?

> *“Python Programming”* ***contains*** *“Programming”*
>
> *The output is True*
>
> *“Python Programming”* ***contains*** *“Language”*
>
> *The output is False*

Pandas — 18 questions

35. How to create dataframe from list?

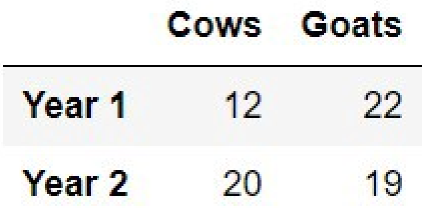
```
fruit_sales = pd.DataFrame([[35, 21], [41, 34]], columns=[‘Apples’, ‘Bananas’],index=[‘2017 Sales’, ‘2018 Sales’])
```



	Apples	Bananas
2017 Sales	35	21
2018 Sales	41	34

36. How to create dataframe from dictionary?

```
animals = pd.DataFrame({‘Cows’: [12, 20], ‘Goats’: [22, 19]}, index=[‘Year 1’, ‘Year 2’])
```



	Cows	Goats
Year 1	12	22
Year 2	20	19

37. How to import csv?

```
import pandas as pd

cr_data = pd.read_csv(“credit_risk_dataset.csv”)
```

38. How to export csv?

```
import pandas as pd

animals.to_csv(“cows_and_goats.csv”)
```

39. How do you select columns from dataframe?

> *Selecting the ‘description’ column from ‘reviews’ dataframe*

```
reviews[‘description’]
```

40. How do you select rows from dataframe?

> *Selecting the first row from ‘reviews’ dataframe*

> *reviews.iloc[0]*

41. How do you select both rows and columns from dataframe?

> *Selecting the first row of 'description' column from 'reviews' dataframe*
>
> *reviews['description'].iloc[0]*

42. How do you select rows based on indices?

> *Selecting rows 1, 2, 3, 5 and 8 from 'reviews' dataframe*
>
> *indices = [1, 2, 3, 5, 8]*
>
> *sample_reviews = reviews.loc[indices]*

43. How do you find the median value?

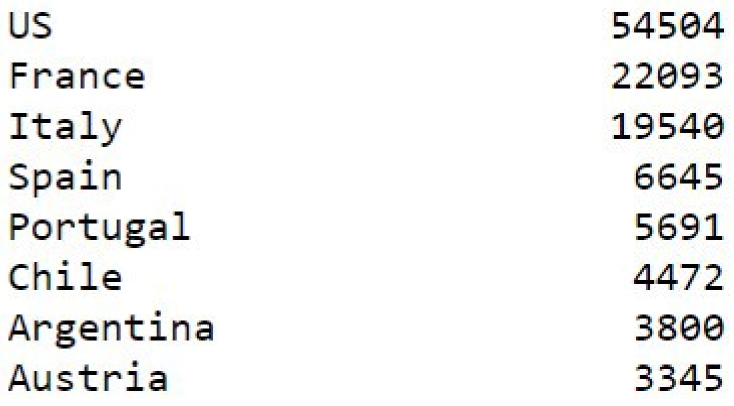
> *Finding the median of 'points' column from 'reviews' dataframe*
>
> *reviews['points'].median()*

44. How do you find the unique values?

> *Finding all the unique countries in 'country' column from 'reviews' dataframe*
>
> *reviews['country'].unique()*

45. How do you find count of unique values?

> *Finding the count of unique countries in 'country' column from 'reviews' dataframe*
>
> *reviews['country'].value_counts()*



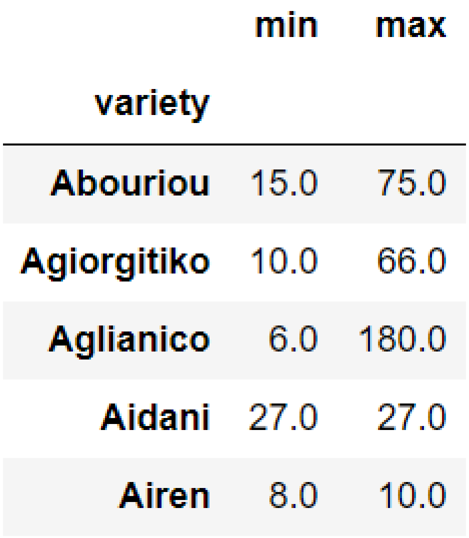
```
US                    54504
France                22093
Italy                 19540
Spain                  6645
Portugal               5691
Chile                  4472
Argentina              3800
Austria                3345
```

46. How do you group on a particular variable?

> *Find the count of 'taster_twitter_handle' column from 'reviews' dataframe*
>
> *reviews.groupby('taster_twitter_handle').size()*

47. How do you apply functions after grouping on a particular variable?

> *Find the min and max of 'price' for different 'variety' column from 'reviews' dataframe*
>
> *reviews.groupby('variety').['price'].agg([min, max])*



	min	max
variety		
Abouriou	15.0	75.0
Agiorgitiko	10.0	66.0
Aglianico	6.0	180.0
Aidani	27.0	27.0
Airen	8.0	10.0

48. How to get the data type of a particular variable?

> *Get the data type of 'points' column from 'reviews' dataframe*
>
> *reviews['points'].dtype*

49. How do you drop columns?

> *Dropping columns 'points' and 'country' from 'reviews' dataframe*
>
> *reviews.drop(['points', 'country'], axis=1, inplace=True)*

50. How do you keep columns?

> *Keeping columns 'points' and 'country' from 'reviews' dataframe*
>
> *reviews = reviews[['points', 'country']]*

51. How do you rename a column?

> *Rename 'region_1' as 'region' and 'region_2' as 'locale'*
>
> *reviews.rename(columns=dict(region_1='region', region_2='locale'))*

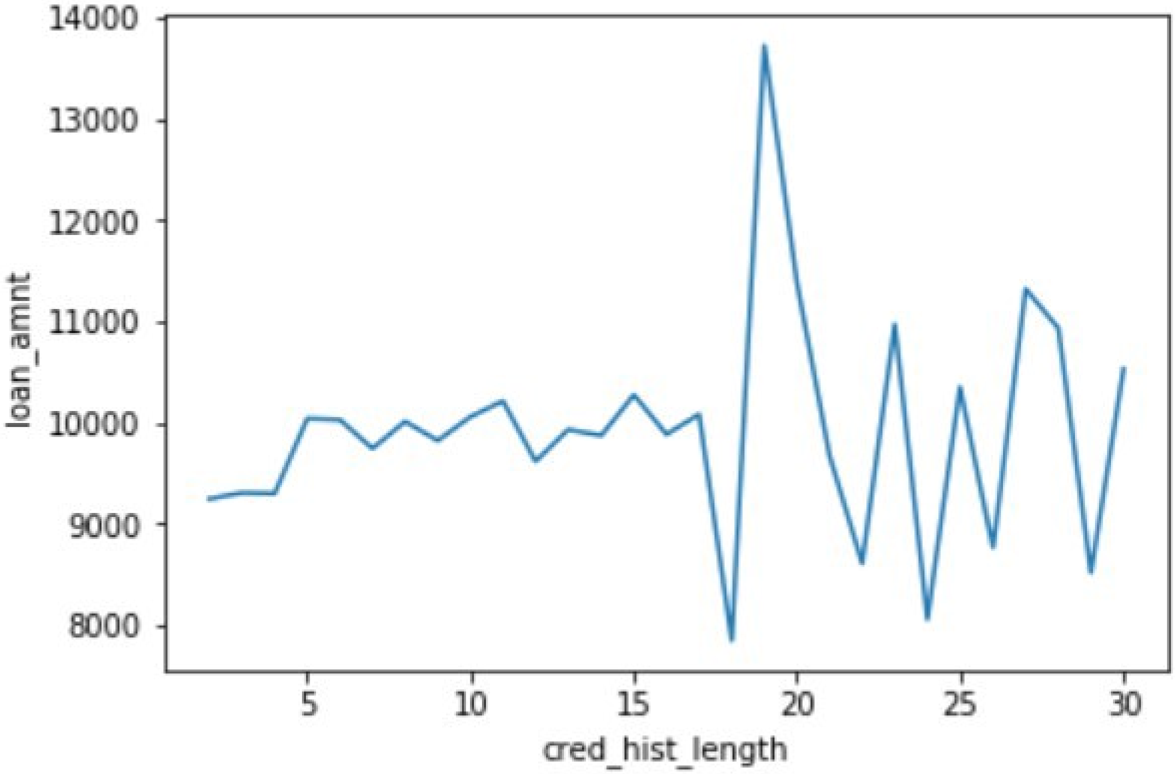
52. How do you sort a dataframe based on a variable?

> *Sorting 'region_1' in descending order*
>
> *reviews['region_1'].sort_values(ascending=False)*

Visualization — 8 questions

53. How do you plot a line chart?

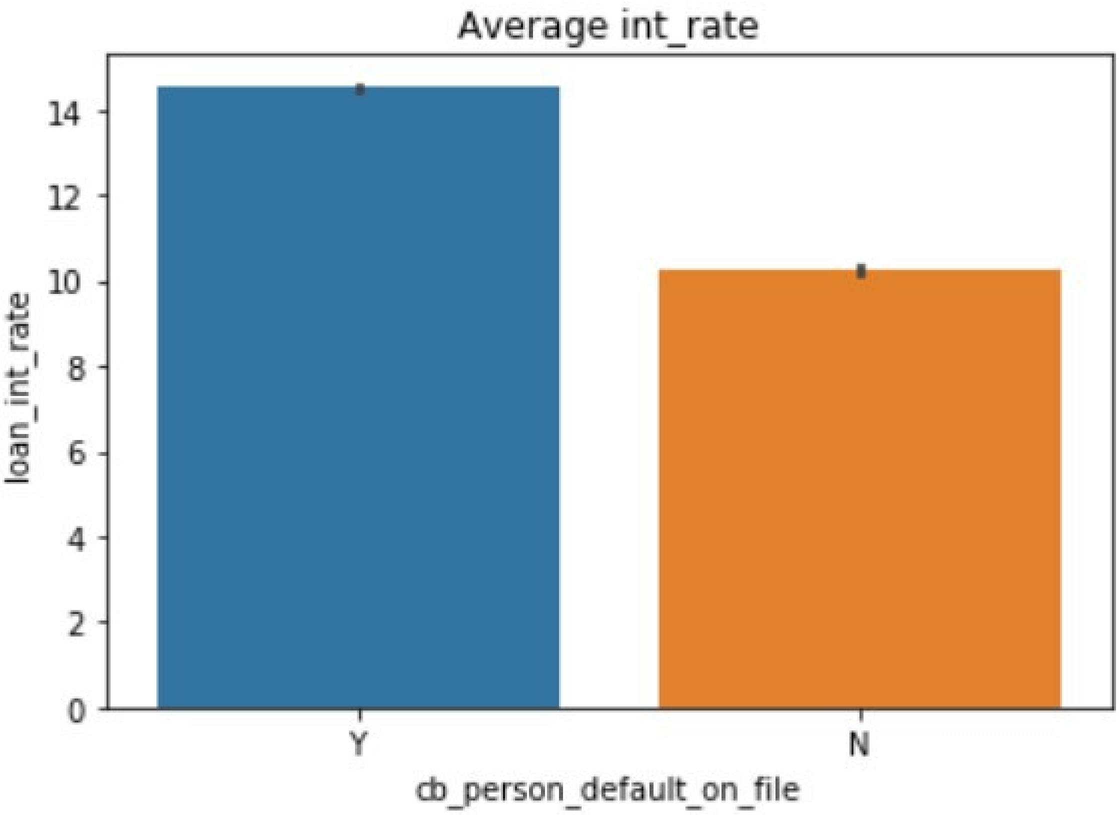
> *import seaborn as sns*
>
> *sns.lineplot(data=loan_amnt)*



54. How do you plot a bar chart?

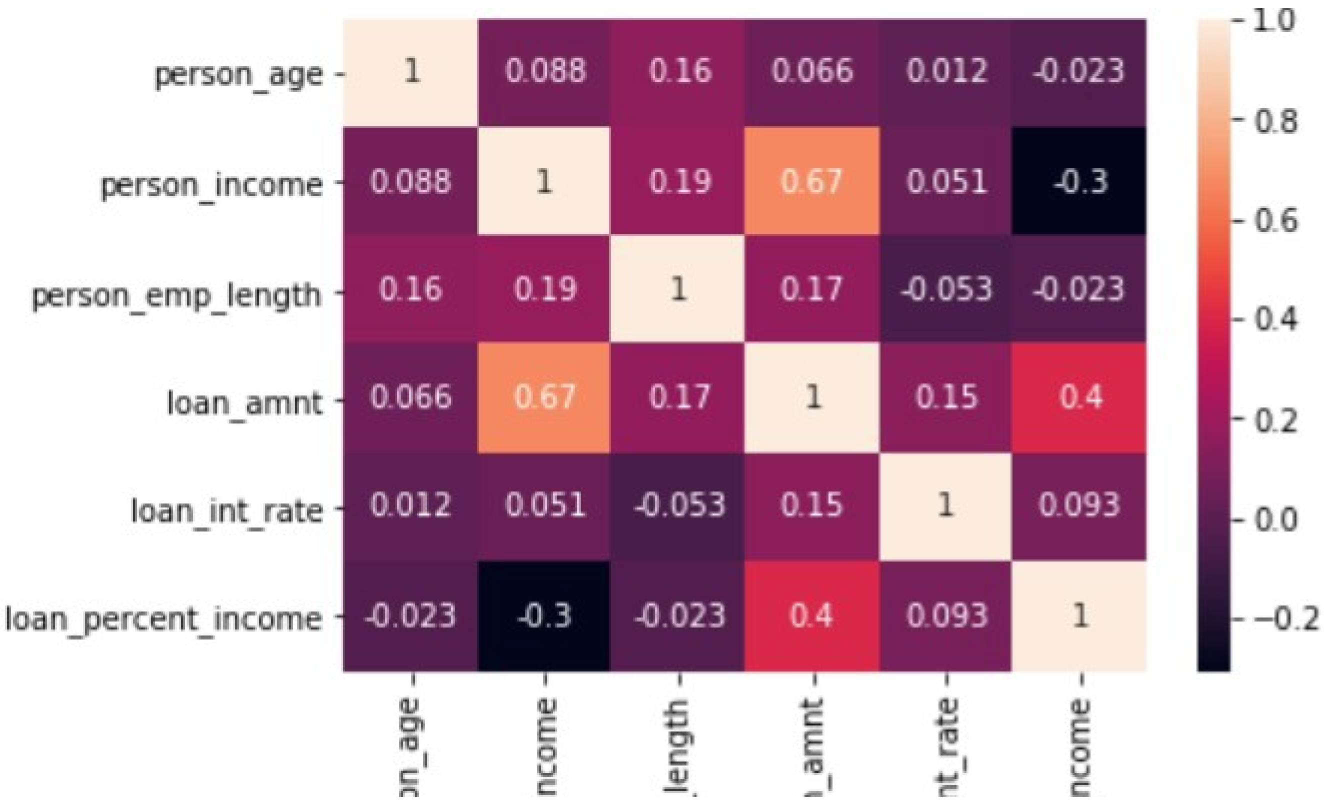
> *import seaborn as sns*

> *sns.barplot(x=cr_data['cb_person_default_on_file'], y=cr_data['loan_int_rate'])*



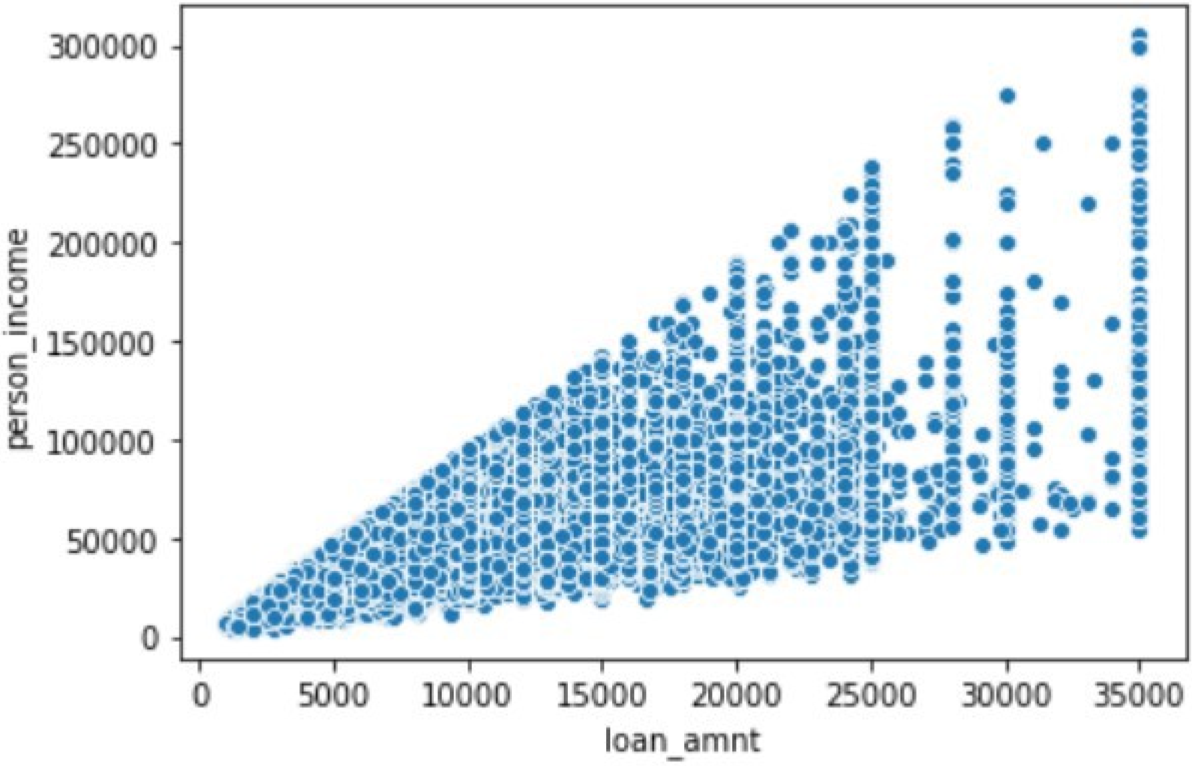
55. How do you plot heat map?

> *import seaborn as sns*
>
> *sns.heatmap(num_data.corr(), annot=True)*



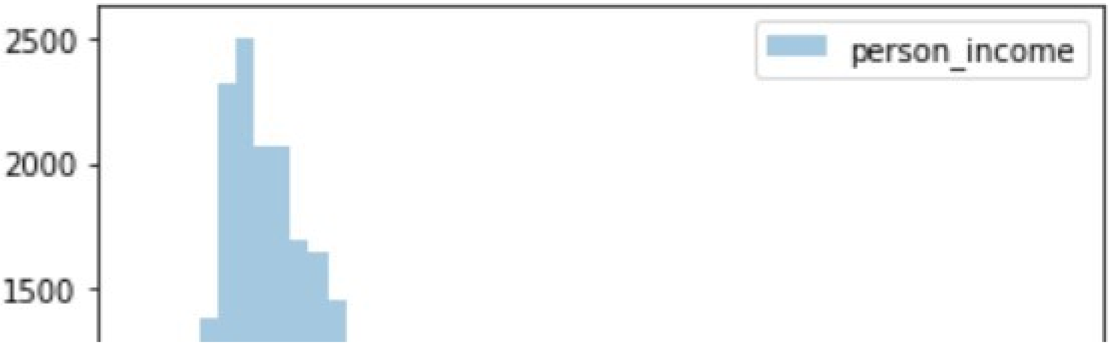
56. How do you plot scatter plot?

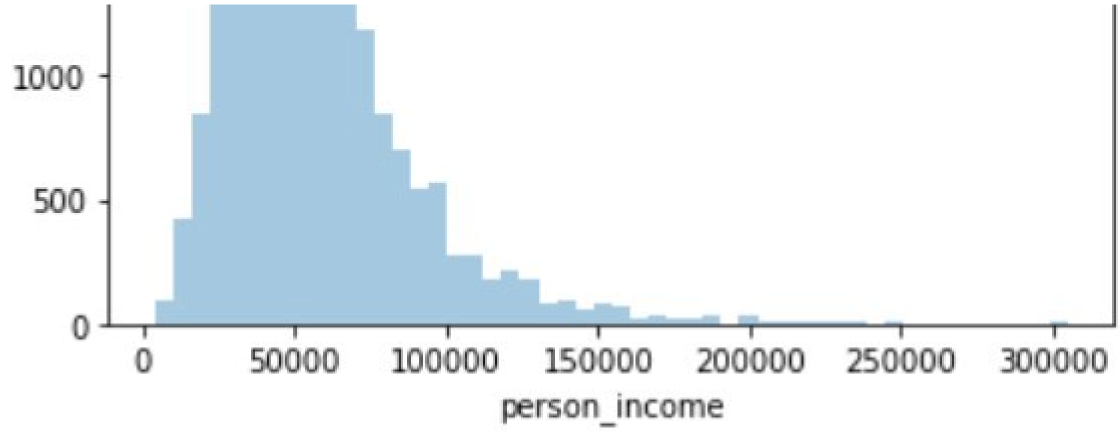
```
import seaborn as sns

sns.scatterplot(x=cr_data['loan_amnt'], y=cr_data['person_income'])
```



57. How do you plot distribution chart?

```
import seaborn as sns

sns.distplot(a=cr_data['person_income'], label="person_income", kde=False)
```





58. How do you add x-label and y-label to the chart?

> *import matplotlib.pyplot as plt*
>
> *plt.xlabel("cred_hist_length")*
>
> *plt.ylabel("loan_amnt")*

59. How do you add title to the chart?

> *import matplotlib.pyplot as plt*
>
> *plt.title("Average int_rate")*

60. How do you add legend to chart?

> *import matplotlib.pyplot as plt*
>
> *plt.legend()*

Data Cleaning — 5 questions

61. How do you identify missing values?

> *The function used to identify the missing value is through .isnull()*
>
> *The code below gives the total number of missing data points in the data frame*
>
> *missing_values_count = sf_permits.isnull().sum()*

62. How do you impute missing values value imputation?

> *Replace missing values with zero / mean*
>
> *df['income'].fillna(0)*
>
> *df['income'] = df['income'].fillna((df['income'].mean()))*

63. What is scaling of data?

> ***Scaling convert the data using the formula = (value — min value) / (max value — min value)***
>
> *from sklearn.preprocessing import MinMaxScaler*
>
> *scaler = MinMaxScaler()*
>
> *original_data = pd.DataFrame(kickstarters_2017['usd_goal_real'])*
>
> *scaled_data = pd.DataFrame(scaler.fit_transform(original_data))*
>
> ***Original data***
>
> *Minimum value: 0.01*
>
> *Maximum value: 166361390.71*
>
> ***Scaled data***
>
> *Minimum value: 0.0*
>
> *Maximum value: 1.0*

64. What is normalizing of data?

> ***Scaling convert the data using the formula = (value — mean) / standard deviation***
>
> *from sklearn.preprocessing import StandardScaler*
>
> *scaler = StandardScaler()*
>
> *original_data = pd.DataFrame(kickstarters_2017['usd_goal_real'])*
>
> *scaled_data = pd.DataFrame(scaler.fit_transform(original_data))*

> ***Original data***
>
> *Minimum value: 0.01*
>
> *Maximum value: 166361390.71*
>
> ***Scaled data***
>
> *Minimum value: -0.10*
>
> *Maximum value: 212.57*

65. How do you treat dates in python?

> *To convert dates from String to Date*
>
> *import datetime*
>
> *import pandas as pd*
>
> *df['Date_parsed'] = pd.to_datetime(df['Date'], format="%m/%d/%Y")*

Machine Learning — 15 questions

66. What is logistic regression?

> *Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.*

67. What is the syntax for logistic regression?

> ***Library:*** *sklearn.linear_model.LogisticRegression*
>
> ***Define model:*** *lr = LogisticRegression()*
>
> ***Fit model:*** *model = lr.fit(x, y)*
>
> ***Predictions:*** *pred = model.predict_proba(test)*

68. How do you split the data in train / test?

> ***Library:*** *sklearn.model_selection.train_test_split*
>
> ***Syntax:*** *X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)*

69. What is decision tree?

> *Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.*

70. What is the syntax for decision tree classifier?

> ***Library:*** *sklearn.tree.DecisionTreeClassifier*
>
> ***Define model:*** *dtc = DecisionTreeClassifier()*
>
> ***Fit model:*** *model = dtc.fit(x, y)*
>
> ***Predictions:*** *pred = model.predict_proba(test)*

71. What is random forest?

> *Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.*

72. What is the syntax for random forest classifier?

> ***Library:*** *sklearn.ensemble.RandomForestClassifier*
>
> ***Define model:*** *rfc = RandomForestClassifier()*
>
> ***Fit model:*** *model = rfc.fit(x, y)*
>
> ***Predictions:*** *pred = model.predict_proba(test)*

73. What is gradient boosting?

> *Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like*

> *other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.*

74. What is the syntax for gradient boosting classifier?

> ***Library:*** *sklearn.ensemble.GradientBoostingClassifier*
>
> ***Define model:*** *gbc = GradientBoostingClassifier()*
>
> ***Fit model:*** *model = gbc.fit(x, y)*
>
> ***Predictions:*** *pred = model.predict_proba(test)*

75. What is SVM?

> *Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.*

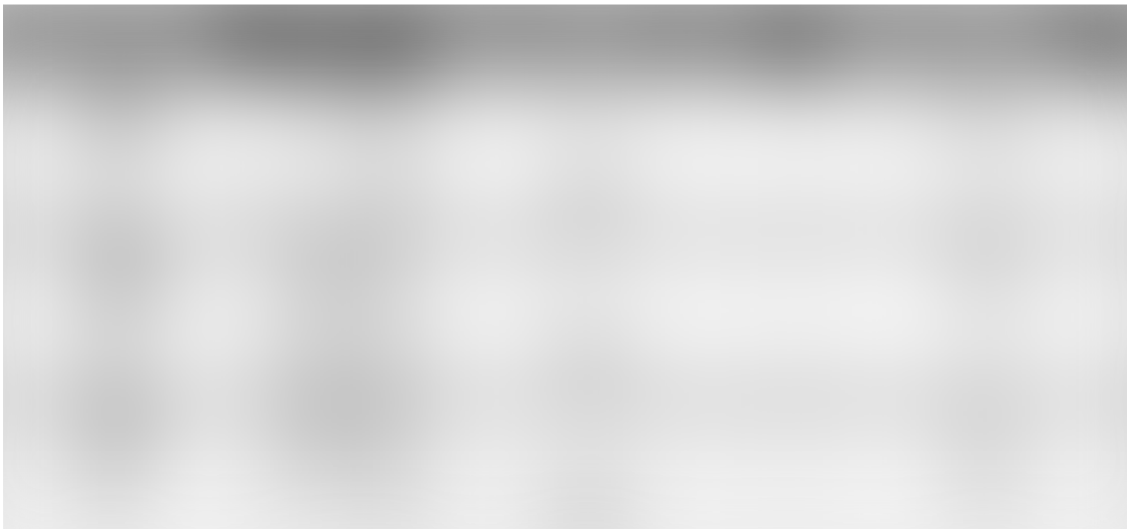
76. What is the difference between KNN and KMeans?

> ***KNN:***
>
> *Supervised classification algorithm*
>
> *Classifies new data points accordingly to the k number or the closest data points*
>
> ***KMeans:***
>
> *Unsupervised clustering algorithm*
>
> *Groups data into k number of clusters.*

77. How do you treat categorical variables?

> *Replace categorical variables with the average of target for each category*

> *One hot encoding*



78. How do you treat missing values?

> ***Drop rows having missing values***
>
> *DataFrame.dropna(axis=0, how='any', inplace=True)*
>
> ***Drop columns***
>
> *DataFrame.dropna(axis=1, how='any', inplace=True)*
>
> ***Replace missing values with zero / mean***
>
> *df['income'].fillna(0)*
>
> *df['income'] = df['income'].fillna((df['income'].mean()))*

79. How do you treat outliers?

> *Inter quartile range is used to identify the outliers.*
>
> *Q1 = df['income'].quantile(0.25)*

> *Q3 = df['income'].quantile(0.75)*
>
> *IQR = Q3 — Q1*
>
> *df = df[(df['income'] >= (Q1–1.5 * IQR)) & (df['income'] <= (Q3 + 1.5 * IQR))]*

80. What is bias / variance trade off?

> ***Definition***
>
> *The Bias-Variance Trade off is relevant for supervised machine learning, specifically for predictive modelling. It's a way to diagnose the performance of an algorithm by breaking down its prediction error.*
>
> ***Error from Bias***
>
> *Bias is the difference between your model's expected predictions and the true values.*
>
> *This is known as under-fitting.*
>
> *Does not improve with collecting more data points.*
>
> ***Error from Variance***
>
> *Variance refers to your algorithm's sensitivity to specific sets of training data.*
>
> *This is known as over-fitting.*
>
> *Improves with collecting more data points.*

Python Interview Data Science Machine Learning