

[Painel do utilizador](#)

As minhas unidades curriculares

[Introdução à análise de dados em Pyth](#)[Exame Final Presencial](#)[Final exam 22/6](#)**Início** quarta, 22 de junho de 2022 às 17:28**Estado** Prova submetida**Data de  
submissão:** quarta, 22 de junho de 2022 às 18:57**Tempo gasto** 1 hora 29 minutos**Nota** 17,5 de um máximo de 20,0 (88%)

## Pergunta 1

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Print the values from columns 'race' to 'salary' from rows 1, 4, and 6.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours		ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado					
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>		race	sex	weekhours	country	salary
	1	White	Male	13	United-States	40312
	4	Black	Female	40	Cuba	31375
	6	Black	Female	16	Jamaica	27656

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     print(df.loc[[1,4,6], 'race':'salary'])
```

	Teste	Esperado	Recebido																																																																	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';',parse_dates=['birthday']) solve()</pre>	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th></tr><tr><td>country</td><td>salary</td><td></td><td></td></tr><tr><td>1</td><td>White</td><td>Male</td><td>13</td></tr><tr><td>United-States</td><td>40312</td><td></td><td></td></tr><tr><td>4</td><td>Black</td><td>Female</td><td>40</td></tr><tr><td>Cuba</td><td>31375</td><td></td><td></td></tr><tr><td>6</td><td>Black</td><td>Female</td><td>16</td></tr><tr><td>Jamaica</td><td>27656</td><td></td><td></td></tr></table>		race	sex	weekhours	country	salary			1	White	Male	13	United-States	40312			4	Black	Female	40	Cuba	31375			6	Black	Female	16	Jamaica	27656			<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th></tr><tr><td>country</td><td>salary</td><td></td><td></td></tr><tr><td>1</td><td>White</td><td>Male</td><td>13</td></tr><tr><td>United-States</td><td>40312</td><td></td><td></td></tr><tr><td>4</td><td>Black</td><td>Female</td><td>40</td></tr><tr><td>Cuba</td><td>31375</td><td></td><td></td></tr><tr><td>6</td><td>Black</td><td>Female</td><td>16</td></tr><tr><td>Jamaica</td><td>27656</td><td></td><td></td></tr></table>		race	sex	weekhours	country	salary			1	White	Male	13	United-States	40312			4	Black	Female	40	Cuba	31375			6	Black	Female	16	Jamaica	27656			✓
	race	sex	weekhours																																																																	
country	salary																																																																			
1	White	Male	13																																																																	
United-States	40312																																																																			
4	Black	Female	40																																																																	
Cuba	31375																																																																			
6	Black	Female	16																																																																	
Jamaica	27656																																																																			
	race	sex	weekhours																																																																	
country	salary																																																																			
1	White	Male	13																																																																	
United-States	40312																																																																			
4	Black	Female	40																																																																	
Cuba	31375																																																																			
6	Black	Female	16																																																																	
Jamaica	27656																																																																			
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';',parse_dates=['birthday']) solve()</pre>	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th></tr><tr><td>country</td><td>salary</td><td></td><td></td></tr><tr><td>1</td><td>White</td><td>Female</td><td>24</td></tr><tr><td>United-States</td><td>35437</td><td></td><td></td></tr><tr><td>4</td><td>White</td><td>Male</td><td>40</td></tr><tr><td>Mexico</td><td>34875</td><td></td><td></td></tr><tr><td>6</td><td>White</td><td>Female</td><td>40</td></tr><tr><td>United-States</td><td>26468</td><td></td><td></td></tr></table>		race	sex	weekhours	country	salary			1	White	Female	24	United-States	35437			4	White	Male	40	Mexico	34875			6	White	Female	40	United-States	26468			<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th></tr><tr><td>country</td><td>salary</td><td></td><td></td></tr><tr><td>1</td><td>White</td><td>Female</td><td>24</td></tr><tr><td>United-States</td><td>35437</td><td></td><td></td></tr><tr><td>4</td><td>White</td><td>Male</td><td>40</td></tr><tr><td>Mexico</td><td>34875</td><td></td><td></td></tr><tr><td>6</td><td>White</td><td>Female</td><td>40</td></tr><tr><td>United-States</td><td>26468</td><td></td><td></td></tr></table>		race	sex	weekhours	country	salary			1	White	Female	24	United-States	35437			4	White	Male	40	Mexico	34875			6	White	Female	40	United-States	26468			✓
	race	sex	weekhours																																																																	
country	salary																																																																			
1	White	Female	24																																																																	
United-States	35437																																																																			
4	White	Male	40																																																																	
Mexico	34875																																																																			
6	White	Female	40																																																																	
United-States	26468																																																																			
	race	sex	weekhours																																																																	
country	salary																																																																			
1	White	Female	24																																																																	
United-States	35437																																																																			
4	White	Male	40																																																																	
Mexico	34875																																																																			
6	White	Female	40																																																																	
United-States	26468																																																																			



Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150



## Pergunta 2

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Calculate the minimum, maximum and the median of the 'age' for the persons with a 'salary' greater than or equal 150000 (rounded to one decimal place)

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import numpy as np import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>min      28.0 max      88.0 median   47.0 Name: age, dtype: float64</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     res = df.loc[ (df['salary'] >= 150000)]
3     f = res['age'].describe()
4     del f['count']
5     del f['std']
6     del f['25%']
7     del f['50%']
8     del f['75%']
9     del f['mean']
10    f['median'] = res['age'].median()
11    print(f.round(1))
```

	Teste	Esperado	Recebido	
✓	<pre>import numpy as np import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>min      28.0 max      88.0 median   47.0 Name: age, dtype: float64</pre>	<pre>min      28.0 max      88.0 median   47.0 Name: age, dtype: float64</pre>	✓
✓	<pre>import numpy as np import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>min      28.0 max      88.0 median   47.0 Name: age, dtype: float64</pre>	<pre>min      28.0 max      88.0 median   47.0 Name: age, dtype: float64</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150





## Pergunta 3

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Delete the rows from the dataframe where 'sex' is equal to 'Male' and 'age' greater than or equal 45.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) print(df.shape) solve() print(df.shape)</pre>	<pre>(10853, 12) (8790, 12)</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     i = df.loc[ (df['sex'] == 'Male') & (df['age'] >= 45)].index
3     df.drop(i,inplace=True)
4
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) print(df.shape) solve() print(df.shape)</pre>	<pre>(10853, 12) (8790, 12)</pre>	<pre>(10853, 12) (8790, 12)</pre>	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) print(df.shape) solve() print(df.shape)</pre>	<pre>(10853, 12) (8733, 12)</pre>	<pre>(10853, 12) (8733, 12)</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150



Pergunta 4

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
From the adults having a 'salary' greater than or equal 150000 what is the percentage of married people? Consider as married those who have a 'status' starting with 'Married'. (rounded to one decimal place)

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	83.9%

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     sal = df.loc[ (df['salary'] >= 150000)]
3     res = sal.loc[ df['status'].str.contains('Married-')]
4     per = round(100*len(res)/len(sal),1)
5     print(f'{per}%')
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	83.9%	83.9%	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()</pre>	85.5%	85.5%	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150

Pergunta 5

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
What is the minimum 'age' and what is the average 'salary' for people with that 'age'. (rounded to one decimal place)  
(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>Minimum age: 15 Average salary for people with that age: 23585.4</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     age = df['age'].min()
3     ave = df.loc[ df['age'] == age]
4     salary = round(ave['salary'].mean(),1)
5     print('Minimum age:',age)
6     print('Average salary for people with that age:',salary)
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>Minimum age: 15 Average salary for people with that age: 23585.4</pre>	<pre>Minimum age: 15 Average salary for people with that age: 23585.4</pre>	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>Minimum age: 15 Average salary for people with that age: 23491.1</pre>	<pre>Minimum age: 15 Average salary for people with that age: 23491.1</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150



## Pergunta 6

Correta Pontuou 1,150 de 1,150

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

How many people earn more than 150000. Of those people how many have an 'education' greater than or equal to 13.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	1358 991

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     df1 = df[df['salary'] > 150000]
3     print(len(df1))
4     df2 = df1[df1['education'] >= 13]
5     print(len(df2))
```

	Teste	Esperado	Recebido	
✓	import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	1358 991	1358 991	✓
✓	import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()	1393 1037	1393 1037	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,150/1,150

Pergunta 7

Não respondida Pontuou 0,000 de 1,250

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Create a function 'hif\_und(name)' that given a string returns a new string replacing the '-' by '\_'. Using the 'hif\_und(name)' function update the column 'status' replacing the '-' by '\_'

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) print(hif_und('very-high')) solve() print(df.loc[:, 'age': 'status'].head(3))</pre>	<pre>very_high age      workclass  education      status 0  37      State-gov      13      Never_married 1  48  Self-emp-not-inc      13  Married_civ_spouse 2  36      Private      9      Divorced</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

1 | |

Pergunta 8

Não respondida Pontuou 0,000 de 1,250

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Print the bottom three values of the number of birthdays per year.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	1939    2 1937    1 1934    1 Name: birthday, dtype: int64

Resposta: (regime de penalização: 0, 10, 20, ... %)

1 | |

## Pergunta 9

Correta Pontuou 1,250 de 1,250

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Insert a column named 'retirementdate' before column 'birthday' equal to the retirement date defined as the day the person turns 66 years old.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado			
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve() print(df.loc[:, 'salary': 'birthday'].head(3))</pre>		salary	retirementdate	birthday
	0	35843	2049-07-13	1983-07-13
	1	40312	2038-02-04	1972-02-04
	2	30687	2050-12-16	1984-12-16

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     ret = df['birthday'] + pd.DateOffset(years = 66)
3     df.insert(11, 'retirementdate', ret)
4
```

	Teste	Esperado	Recebido																																	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';',parse_dates= ['birthday']) solve() print(df.loc[:, 'salary': 'birthday'].head(3))</pre>	<table><tr><th></th><th>salary</th><th>retirementdate</th><th>birthday</th></tr><tr><td>0</td><td>35843</td><td>2049-07-13</td><td>1983-07-13</td></tr><tr><td>1</td><td>40312</td><td>2038-02-04</td><td>1972-02-04</td></tr><tr><td>2</td><td>30687</td><td>2050-12-16</td><td>1984-12-16</td></tr></table>		salary	retirementdate	birthday	0	35843	2049-07-13	1983-07-13	1	40312	2038-02-04	1972-02-04	2	30687	2050-12-16	1984-12-16	<table><tr><th></th><th>salary</th><th>retirementdate</th><th>birthday</th></tr><tr><td>0</td><td>35843</td><td>2049-07-13</td><td>1983-07-13</td></tr><tr><td>1</td><td>40312</td><td>2038-02-04</td><td>1972-02-04</td></tr><tr><td>2</td><td>30687</td><td>2050-12-16</td><td>1984-12-16</td></tr></table>		salary	retirementdate	birthday	0	35843	2049-07-13	1983-07-13	1	40312	2038-02-04	1972-02-04	2	30687	2050-12-16	1984-12-16	✓
	salary	retirementdate	birthday																																	
0	35843	2049-07-13	1983-07-13																																	
1	40312	2038-02-04	1972-02-04																																	
2	30687	2050-12-16	1984-12-16																																	
	salary	retirementdate	birthday																																	
0	35843	2049-07-13	1983-07-13																																	
1	40312	2038-02-04	1972-02-04																																	
2	30687	2050-12-16	1984-12-16																																	

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates= ['birthday']) solve() print(df.loc[:, 'salary': 'birthday'].head(3))</pre>	<pre>salary retirementdate birthday 0  31375      2060-12-18 1994-12-18 1  35437      2050-09-19 1984-09-19 2 150000      2040-06-17 1974-06-17</pre>	<pre>salary retirementdate birthday 0  31375      2060-12-18 1994-12-18 1  35437      2050-09-19 4-09-19 2 150000      2040-06-17 1974-06-17</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,250/1,250

Pergunta 10

Correta Pontuou 1,250 de 1,250

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Calculate the percentage of adults by 'status' (rounded to one decimal place) sorted in descending order.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>Married-civ-spouse    45.7 Never-married         33.0 Divorced              13.7 Separated              3.3 Widowed               2.9 Married-spouse-absent  1.3 Married-AF-spouse      0.1 Name: status, dtype: float64</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 |
2 | def solve():
3 |     res = round(100*(df['status'].value_counts()/len(df)),1)
4 |     res.sort_values(ascending = False, inplace = True)
5 |     print(res)
```

Teste	Esperado	Recebido	
-------	----------	----------	--

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	Married-civ-spouse 45.7 Never-married 33.0 Divorced 13.7 Separated 3.3 Widowed 2.9 Married-spouse-absent 1.3 Married-AF-spouse 0.1 Name: status, dtype: float64	Married-civ-spouse 45.7 Never-married 33.0 Divorced 13.7 Separated 3.3 Widowed 2.9 Married-spouse-absent 1.3 Married-AF-spouse 0.1 Name: status, dtype: float64	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	Married-civ-spouse 45.8 Never-married 32.8 Divorced 13.7 Widowed 3.3 Separated 3.0 Married-spouse-absent 1.4 Married-AF-spouse 0.1 Name: status, dtype: float64	Married-civ-spouse 45.8 Never-married 32.8 Divorced 13.7 Widowed 3.3 Separated 3.0 Married-spouse-absent 1.4 Married-AF-spouse 0.1 Name: status, dtype: float64	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,250/1,250

Pergunta 11

Correta Pontuou 1,250 de 1,250

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Print the columns 'race' to 'salary' for the rows where the 'salary' is greater than 240000 and race is 'Black'.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado					
import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()		race	sex	weekhours	country	salary
	2578	Black	Male	40	United-States	245416
	5406	Black	Male	50	United-States	312500
	6359	Black	Male	20	United-States	300000

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1
2 def solve():
3     res = df.loc[ (df['salary'] > 240000) & (df['race'] == 'Black')]
4     print(res.loc[:, 'race':'salary'])
```

	Teste	Esperado	Recebido																																																	
✓	import pandas as pd df = pd.read_csv('adult1.csv', sep=';',parse_dates=['birthday']) solve()	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th><th>country</th><th>salary</th></tr><tr><td>2578</td><td>Black</td><td>Male</td><td>40</td><td>United-States</td><td>245416</td></tr><tr><td>5406</td><td>Black</td><td>Male</td><td>50</td><td>United-States</td><td>312500</td></tr><tr><td>6359</td><td>Black</td><td>Male</td><td>20</td><td>United-States</td><td>300000</td></tr></table>		race	sex	weekhours	country	salary	2578	Black	Male	40	United-States	245416	5406	Black	Male	50	United-States	312500	6359	Black	Male	20	United-States	300000	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th><th>country</th><th>salary</th></tr><tr><td>2578</td><td>Black</td><td>Male</td><td>40</td><td>United-States</td><td>245416</td></tr><tr><td>5406</td><td>Black</td><td>Male</td><td>50</td><td>United-States</td><td>312500</td></tr><tr><td>6359</td><td>Black</td><td>Male</td><td>20</td><td>United-States</td><td>300000</td></tr></table>		race	sex	weekhours	country	salary	2578	Black	Male	40	United-States	245416	5406	Black	Male	50	United-States	312500	6359	Black	Male	20	United-States	300000	✓
	race	sex	weekhours	country	salary																																															
2578	Black	Male	40	United-States	245416																																															
5406	Black	Male	50	United-States	312500																																															
6359	Black	Male	20	United-States	300000																																															
	race	sex	weekhours	country	salary																																															
2578	Black	Male	40	United-States	245416																																															
5406	Black	Male	50	United-States	312500																																															
6359	Black	Male	20	United-States	300000																																															
✓	import pandas as pd df = pd.read_csv('adult2.csv', sep=';',parse_dates=['birthday']) solve()	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th><th>country</th><th>salary</th></tr><tr><td>9267</td><td>Black</td><td>Female</td><td>40</td><td>United-States</td><td>248333</td></tr></table>		race	sex	weekhours	country	salary	9267	Black	Female	40	United-States	248333	<table><tr><th></th><th>race</th><th>sex</th><th>weekhours</th><th>country</th><th>salary</th></tr><tr><td>9267</td><td>Black</td><td>Female</td><td>40</td><td>United-States</td><td>248333</td></tr></table>		race	sex	weekhours	country	salary	9267	Black	Female	40	United-States	248333	✓																								
	race	sex	weekhours	country	salary																																															
9267	Black	Female	40	United-States	248333																																															
	race	sex	weekhours	country	salary																																															
9267	Black	Female	40	United-States	248333																																															

Passou em todos os testes! ✓

Correta





Nota desta submissão: 1,250/1,250



## Pergunta 12

Correta Pontuou 1,370 de 1,370

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Calculate the number of persons in each of the following 'age' classes:

[0, 20[ - very\_young

[20, 40[ - young

[40, 60[ - middle\_age

[60, 80[ - old

[80, 100[ - very\_old

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>very_young    1055 young         5627 middle_age    3514 old           630 very_old       27 Name: age, dtype: int64</pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     df['age'] = pd.cut(df['age'], [0,20,40,60,80,100], right=False, labels=['very_young','young','middle_ag
3     res = df['age'].value_counts()
4     res.sort_index(ascending = True, inplace = True)
5     ##res = res.reset_index(drop = True)
6     print(res)
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>very_young    1055 young         5627 middle_age    3514 old           630 very_old       27 Name: age, dtype: int64</pre>	<pre>very_young    1055 young         5627 middle_age    3514 old           630 very_old       27 Name: age, dtype: int64</pre>	✓

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>very_young    988 young         5603 middle_age    3539 old           695 very_old       28 Name: age, dtype: .&gt;</pre>	<pre>very_young    988 young         5603 middle_age    3539 old           695 very_old       28 Name: age, dtype: int64</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,370/1,370

Pergunta 13

Correta Pontuou 1,370 de 1,370

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Calculate the average age by 'status' and 'sex' (rounded to one decimal place).

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado		
import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	sex	Female	Male
	status		
	Divorced	41.1	39.8
	Married-AF-spouse	22.0	32.3
	Married-civ-spouse	37.5	41.6
	Married-spouse-absent	37.5	39.4
	Never-married	26.2	26.3
	Separated	37.2	36.8
	Widowed	57.0	58.8

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 | def solve():  
2 |     pt = df.pivot_table( index = 'status', columns = 'sex', values = 'age', aggfunc = 'mean')  
3 |     print(pt.round(1))
```

	Teste	Esperado	Recebido	
--	-------	----------	----------	--

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>sex Female Male status Divorced 41.1 39.8 Married-AF-spouse 22.0 32.3 Married-civ-spouse 37.5 41.6 Married-spouse-absent 37.5 39.4 Never-married 26.2 26.3 Separated 37.2 36.8 Widowed 57.0 58.8</pre>	<pre>sex Female Male status Divorced     39.8 Married-AF-spouse 22.0 32.3 Married-civ-spouse 37.5 41.6 Married-spouse-absent 37.5 39.4 Never-married 26.2 26.3 Separated 37.2 36.8 Widowed 57.0 58.8</pre>	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates= ['birthday']) solve()</pre>	<pre>sex Female Male status Divorced 41.3 41.4 Married-AF-spouse 37.0 25.0 Married-civ-spouse 38.1 41.8 Married-spouse-absent 38.5 40.1 Never-married 26.2 26.4 Separated 37.6 37.7 Widowed 56.5 61.0</pre>	<pre>sex Female Male status Divorced 41.3 41.4 Married-AF-spouse 37.0 25.0 Married-civ-spouse 38.1 41.8 Married-spouse-absent 38.5 40.1 Never-married 26.2 26.4 Separated 37.6 37.7 Widowed 56.5 61.0</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,370/1,370

## Pergunta 14

Correta Pontuou 1,370 de 1,370

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Print the 'status', the 'race' and the value corresponding to the highest mean 'age' (rounded to one decimal place) by 'status' and 'race'.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	Widowed, White, 57.6

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     pt = df.pivot_table(index='race', columns='status', values='age', aggfunc='mean')
3     city = pt.max().idxmax()
4     product = pt[city].idxmax()
5     total = round(pt[city].max(), 1)
6     print(f"{city}, {product}, {total}")
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	Widowed, White, 57.6	Widowed, White, 57.6	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()</pre>	Widowed, Amer-Indian-Eskimo, 64.2	Widowed, Amer-Indian-Eskimo, 64.2	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,370/1,370

## Pergunta 15

Correta Pontuou 1,370 de 1,370

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:

Calculate the average salary for 'status': 'Never-married' and 'Married-civ-spouse' for 'race' 'White' (rounded to one decimal place).

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	>	ountry	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40		United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13		United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40		United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado
import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	<pre> race           White status Married-civ-spouse  88459.7 Never-married      34639.2 </pre>

Resposta: (regime de penalização: 0, 10, 20, ... %)

```

1 def solve():
2     df1 = df.loc[ ((df['status'] == 'Never-married') | (df['status'] == 'Married-civ-spouse')) & (df['race'] == 'White')]
3     pt = df1.pivot_table(index = 'status', columns = 'race', values = 'salary', aggfunc = 'mean')
4     print(pt.round(1))

```

	Teste	Esperado	Recebido	
✓	import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()	<pre> race White status Married-civ-spouse  88459.7 Never-married      34639.2 </pre>	<pre> race White status Married-civ-spouse  88459.7 Never-married      34639.2 </pre>	✓
✓	import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()	<pre> race White status Married-civ-spouse  88761.8 Never-married      35341.4 </pre>	<pre> race White status Married-civ-spouse  88761.8 Never-married      35341.4 </pre>	✓

Passou em todos os testes! ✓



Correta

Nota desta submissão: 1,370/1,370





Pergunta 16

Correta Pontuou 1,370 de 1,370

Using the Pandas library and dataframe 'df' create a function 'solve()' to answer the following:  
Print the columns 'education' to 'salary' of the rows where the 'education' is greater than 15 and the 'salary' is less than 35000 or the 'education' is less than 5 and the 'salary' is greater than 100000.

(1st test file: adult1.csv)

	age	workclass	education	status	occupation	relationship	race	sex	weekhours	country	salary	birthday
0	37	State-gov	13	Never-married	Adm-clerical	Not-in-family	White	Male	40	United-States	35843	1983-07-13
1	48	Self-emp-not-inc	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	13	United-States	40312	1972-02-04
2	36	Private	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	40	United-States	30687	1984-12-16

Por exemplo:

Teste	Resultado			
import pandas as pd df = pd.read_csv('adult1.csv', sep=';',parse_dates=[ 'birthday']) solve()	education	status	...	country
	salary			
	541	16	Never-married	United-States
	34500			
	574	16	Never-married	United-States
	33500			
	2284	16	Never-married	Canada
	34500			
	5072	4	Married-civ-spouse	United-States
	100833			
	6590	4	Married-spouse-absent	Italy
	114166			
	6964	16	Married-civ-spouse	India
	34500			
	7220	16	Never-married	United-States
	33500			
	8126	16	Never-married	United-States
	33500			
	8795	4	Married-civ-spouse	United-States
	100833			
	[9 rows x 9 columns]			

Resposta: (regime de penalização: 0, 10, 20, ... %)

```
1 def solve():
2     res = df.loc[ ( (df['education'] > 15) & (df['salary'] < 35000) ) | ((df['education'] < 5) & (df['salary'] > 100000)) ]
3     print(res.loc[:, 'education': 'salary'])
```

	Teste	Esperado	Recebido	
✓	<pre>import pandas as pd df = pd.read_csv('adult1.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>education status ... country salary 541      16      Never- married ... United-States 34500 574      16      Never- married ... United-States 33500 2284     16      Never- married ...      Canada 34500 5072     4      Married-civ- spouse ... United-States 100833 6590     4      Married-spouse- absent ...      Italy 114166 6964     16      Married-civ- spouse ...      India 34500 7220     16      Never- married ... United-States 33500 8126     16      Never- married ... United-States 33500 8795     4      Married-civ- spouse ... United-States 100833  [9 rows x 9 columns]</pre>	<pre>education status ... country salary 541      16      Never- married ... United-States 34500 574      16      Never- married &gt; . United-States 33500 2284     16      Never- married ...      Canada 34500 5072     4      Married-civ- spouse ... United-States 100833 6590     4      Married-spouse- absent ...      Italy 114166 6964     16      Married-civ- spouse ...      India 34500 7220     16      Never- married ... United-States 33500 8126     16      Never- married ... United-States 33500 8795     4      Married-civ- spouse ... United-States 100833  [9 rows x 9 columns]</pre>	✓
✓	<pre>import pandas as pd df = pd.read_csv('adult2.csv', sep=';', parse_dates=['birthday']) solve()</pre>	<pre>education status ... country salary 1672     4      Married-civ-spouse ... United-States 102500 1711     4      Married-civ-spouse ... United-States 103333 1935     16      Never-married ...      England 32000 5007     4      Married-civ-spouse ... United-States 101666 5038     4      Married-civ-spouse ... United-States 110833 5751     4      Married-civ-spouse ... United-States 110833 6003     4      Widowed ... United-States 109166 7630     16      Never-married ...      Germany 34000 8079     4      Married-civ-spouse ... United-States 102500 8318     16      Never-married ... United-States 34000 8854     4      Married-civ-spouse ... United-States 105000  [11 rows x 9 columns]</pre>	<pre>education status ... country salary 1672     4      Married-civ-spouse ... United-States 102500 1711     4      Married-civ-spouse ... United-States 103333 1935     16      Never-married ...      England 32000 5007     4      Married-civ-spouse ... United-States 101666 5038     4      Married-civ-spouse ... United-States 110833 5751     4      Married-civ-spouse ... United-States 110833 6003     4      Widowed ... United-States 109166 7630     16      Never-married ...      Germany 34000 8079     4      Married-civ-spouse ... United-States 102500 8318     16      Never-married ... United-States 34000 8854     4      Married-civ-spouse ... United-States 105000  [11 rows x 9 columns]</pre>	✓

Passou em todos os testes! ✓

Correta

Nota desta submissão: 1,370/1,370

◀ Final exam 20/6

Ir para...

