# Analise_Descritiva

April 14, 2021

---

# 1 CURSO DE ESTATÍSTICA - PARTE 1

---

## 1.1 Trabalho de Análise Descritiva de um Conjunto de Dados

Utilizando os conhecimentos adquiridos em nosso treinamento realize uma análise descritiva básica de um conjunto de dados retirados da Pesquisa Nacional por Amostra de Domicílios - 2015 do IBGE.

Vamos construir histogramas, calcular e avaliar medidas de tendência central, medidas separatrizes e de dispersão dos dados.

Siga o roteiro proposto e vá completando as células vazias. Procure pensar em mais informações interessantes que podem ser exploradas em nosso dataset.

# 2 DATASET DO PROJETO

---

### 2.0.1 Pesquisa Nacional por Amostra de Domicílios - 2015

A Pesquisa Nacional por Amostra de Domicílios - PNAD investiga anualmente, de forma permanente, características gerais da população, de educação, trabalho, rendimento e habitação e outras, com periodicidade variável, de acordo com as necessidades de informação para o país, como as características sobre migração, fecundidade, nupcialidade, saúde, segurança alimentar, entre outros temas. O levantamento dessas estatísticas constitui, ao longo dos 49 anos de realização da pesquisa, um importante instrumento para formulação, validação e avaliação de políticas orientadas para o desenvolvimento socioeconômico e a melhoria das condições de vida no Brasil.

### 2.0.2 Fonte dos Dados

https://ww2.ibge.gov.br/home/estatistica/populacao/trabalhoerendimento/pnad2015/microdados.shtm

### 2.0.3 Variáveis utilizadas

#### 2.0.4 Renda

---

Rendimento mensal do trabalho principal para pessoas de 10 anos ou mais de idade.

#### 2.0.5 Idade

---

Idade do morador na data de referência em anos.

#### 2.0.6 Altura (elaboração própria)

---

Altura do morador em metros.

#### 2.0.7 UF

---

| Código | Descrição |
|---|---|
| 11 | Rondônia |
| 12 | Acre |
| 13 | Amazonas |
| 14 | Roraima |
| 15 | Pará |
| 16 | Amapá |
| 17 | Tocantins |
| 21 | Maranhão |
| 22 | Piauí |
| 23 | Ceará |
| 24 | Rio Grande do Norte |
| 25 | Paraíba |
| 26 | Pernambuco |
| 27 | Alagoas |
| 28 | Sergipe |
| 29 | Bahia |
| 31 | Minas Gerais |
| 32 | Espírito Santo |
| 33 | Rio de Janeiro |
| 35 | São Paulo |
| 41 | Paraná |
| 42 | Santa Catarina |
| 43 | Rio Grande do Sul |
| 50 | Mato Grosso do Sul |
| 51 | Mato Grosso |
| 52 | Goiás |
| 53 | Distrito Federal |

#### 2.0.8 Sexo

| Código | Descrição |
|---|---|
| 0 | Masculino |
| 1 | Feminino |

#### 2.0.9 Anos de Estudo

| Código | Descrição |
|---|---|
| 1 | Sem instrução e menos de 1 ano |
| 2 | 1 ano |
| 3 | 2 anos |
| 4 | 3 anos |
| 5 | 4 anos |
| 6 | 5 anos |
| 7 | 6 anos |
| 8 | 7 anos |
| 9 | 8 anos |
| 10 | 9 anos |
| 11 | 10 anos |
| 12 | 11 anos |
| 13 | 12 anos |
| 14 | 13 anos |
| 15 | 14 anos |
| 16 | 15 anos ou mais |
| 17 | Não determinados |
| | Não aplicável |

#### 2.0.10 Cor

| Código | Descrição |
|---|---|
| 0 | Indígena |
| 2 | Branca |
| 4 | Preta |
| 6 | Amarela |
| 8 | Parda |
| 9 | Sem declaração |

**Observação**

Os seguintes tratamentos foram realizados nos dados originais: 1. Foram eliminados os registros onde a Renda era inválida (999 999 999 999); 2. Foram eliminados os registros onde a Renda era missing; 3. Foram considerados somente os registros das Pessoas de Referência de cada domicílio (responsável pelo domicílio).

---

---

### 2.0.11 Utilize a célula abaixo para importar as biblioteca que precisar para executar as tarefas

**Sugestões: pandas, numpy, seaborn**

[1]:
```
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

### 2.0.12 Importe o dataset e armazene o conteúdo em uma DataFrame

[2]:
```
df = pd.read_csv('dados/dados.csv', sep=',')
```

### 2.0.13 Visualize o conteúdo do DataFrame

[3]:
```
df.head(5)
```

[3]:

| | UF | Sexo | Idade | Cor | Anos de Estudo | Renda | Altura |
|---|---|---|---|---|---|---|---|
| 0 | 11 | 0 | 23 | 8 | 12 | 800 | 1.603808 |
| 1 | 11 | 1 | 23 | 2 | 12 | 1150 | 1.739790 |
| 2 | 11 | 1 | 35 | 8 | 15 | 880 | 1.760444 |
| 3 | 11 | 0 | 46 | 2 | 6 | 3500 | 1.783158 |
| 4 | 11 | 1 | 47 | 8 | 9 | 150 | 1.690631 |

### 2.0.14 Para avaliarmos o comportamento da variável RENDA vamos construir uma tabela de frequências considerando as seguintes classes em salários mínimos (SM)

**Descreva os pontos mais relevantes que você observa na tabela e no gráfico.** Classes de renda:

A Acima de 25 SM

B De 15 a 25 SM

C De 5 a 15 SM

D De 2 a 5 SM

E Até 2 SM

Para construir as classes de renda considere que o salário mínimo na época da pesquisa era de R$ 788,00.

**Siga os passos abaixo:**

### 2.0.15 1º Definir os intevalos das classes em reais (R$)

[4]:
```
v_max =df['Renda'].max()
v_min =df['Renda'].min()
SM = 788
```

[5]:
```
classes = [v_min, 2*SM , 5*SM , 15*SM, 25*SM, v_max]
classes
```

[5]:
```
[0, 1576, 3940, 11820, 19700, 200000]
```

### 2.0.16 2º Definir os labels das classes

[6]:
```
labels = ['E', 'D', 'C','B', 'A']
labels
```

[6]:
```
['E', 'D', 'C', 'B', 'A']
```

### 2.0.17 3º Construir a coluna de frequências

[7]:
```
# Com as classes e labels realizamos a relação de renda e faixa de classe.
cut = pd.cut(x=df['Renda'],
       bins= classes,
       labels=labels,
       include_lowest=True)
cut
```

[7]:
```
0        E
1        E
2        E
3        D
4        E
        ..
76835    E
76836    E
76837    E
76838    E
76839    E
Name: Renda, Length: 76840, dtype: category
Categories (5, object): ['E' < 'D' < 'C' < 'B' < 'A']
```

[8]:
```
frequencia = pd.value_counts(cut)
frequencia
```

[8]:
```
E    49755
D    18602
C     7241
B      822
A      420
Name: Renda, dtype: int64
```

#### 2.0.18 4º Construir a coluna de percentuais

[9]:
```
percentual = pd.value_counts(cut, normalize=True)
percentual.round(2)
```

[9]:
```
E    0.65
D    0.24
C    0.09
B    0.01
A    0.01
Name: Renda, dtype: float64
```

#### 2.0.19 5º Juntar as colunas de frequência e percentuais e ordenar as linhas de acordo com os labels das classes

[10]:
```
series = {'Frequência':frequencia ,'Distribuição %':percentual}
distribuicao = pd.DataFrame(series)
distribuicao.sort_index(ascending=True, inplace=True)
distribuicao
```

[10]:

| | Frequência | Distribuição % |
|---|---|---|
| E | 49755 | 0.647514 |
| D | 18602 | 0.242087 |
| C | 7241 | 0.094235 |
| B | 822 | 0.010698 |
| A | 420 | 0.005466 |

[11]:
```
df_labels = ['A > Acima 19700',
             'B > De 11820 a 19700',
             'C > De 3940 a 11820',
             'D > De 1576 a 3940',
             'E >  0 Até 1576',]
df_labels.reverse()
df_labels
```

[11]:
```
['E >  0 Até 1576',
 'D > De 1576 a 3940',
 'C > De 3940 a 11820',
 'B > De 11820 a 19700',
 'A > Acima 19700']
```

[12]:
```
# Criando faixas (cut) de acordo com o corte no bins
n = df.shape[0]
n
```

[12]:
```
76840
```

[13]:
```
# Criando faixas (cut) de acordo com o corte no bins
k = 1 + (10/3) * np.log10(n)
k = int(k.round(0))
k
```

[13]:
```
17
```

[14]:
```
# Criando faixas (cut) de acordo com o corte no bins
cut_bins = pd.value_counts(pd.cut(x=df['Renda'],
                  bins= 17,
                  include_lowest=True),
    sort = False
)
cut_bins
```

[14]:
```
(-200.001, 11764.706]       75594
(11764.706, 23529.412]       1022
(23529.412, 35294.118]        169
(35294.118, 47058.824]         19
(47058.824, 58823.529]         16
(58823.529, 70588.235]          5
(70588.235, 82352.941]          4
(82352.941, 94117.647]          1
(94117.647, 105882.353]         6
(105882.353, 117647.059]        0
(117647.059, 129411.765]        1
(129411.765, 141176.471]        0
(141176.471, 152941.176]        0
(152941.176, 164705.882]        0
(164705.882, 176470.588]        0
(176470.588, 188235.294]        0
(188235.294, 200000.0]          3
Name: Renda, dtype: int64
```

[15]:
```
distribuicao.index = df_labels
distribuicao.rename_axis('Faixas', axis= 'columns', inplace = True)
distribuicao
```

[15]:
```
Faixas                Frequência  Distribuição %
E >  0 Até 1576            49755        0.647514
D > De 1576 a 3940         18602        0.242087
C > De 3940 a 11820         7241        0.094235
```

```
B > De 11820 a 19700          822         0.010698
A > Acima 19700               420         0.005466
```

[16]:
```
per = distribuicao['Distribuição %'].round(4) * 100
per
```

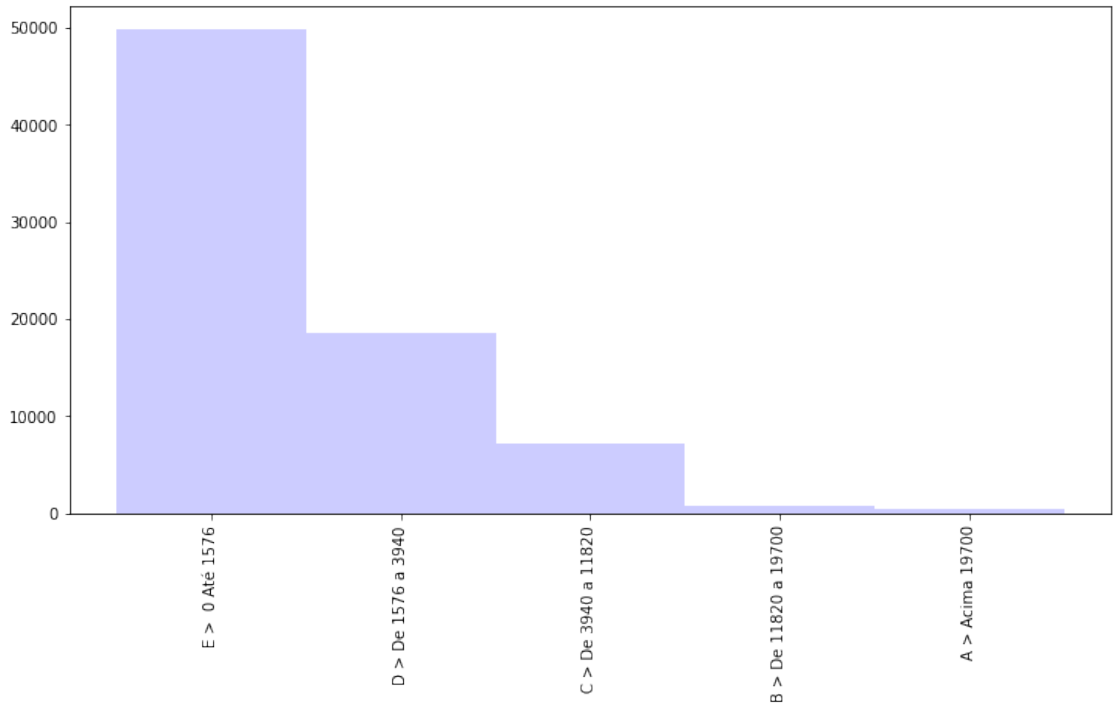
[16]:
```
E >  0 Até 1576          64.75
D > De 1576 a 3940       24.21
C > De 3940 a 11820       9.42
B > De 11820 a 19700      1.07
A > Acima 19700           0.55
Name: Distribuição %, dtype: float64
```

[17]:
```
per.sum()
```

[17]:
```
100.0
```

### 2.0.20 Construa um gráfico de barras para visualizar as informações da tabela de frequências acima

[18]:
```
bar_graf = distribuicao['Frequência'].plot.bar(width= 1, color='blue', alpha=0.
↪2, figsize=(12,6))
```



### 2.0.21 Conclusões

Podemos concluir que a população com maior renda representa o a menor quantidade de pessoas e maior riquezas em contraposição a maior quantidade de pessoas possuem as menores rendas.

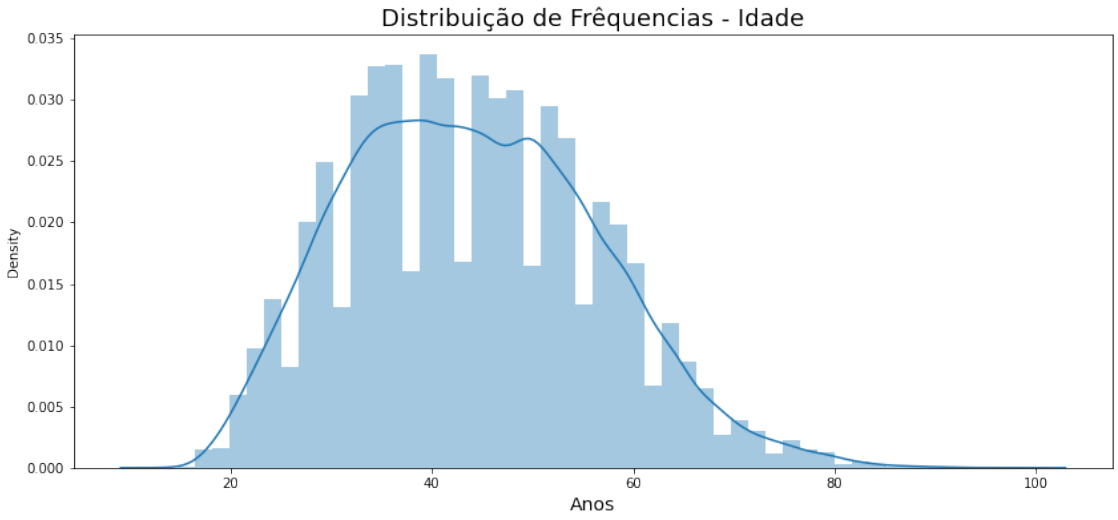
### 2.0.22 Crie um histograma para as variáveis QUANTITATIVAS de nosso dataset

**Descreva os pontos mais relevantes que você observa nos gráficos (assimetrias e seus tipos, possíveis causas para determinados comportamentos etc.)**

[19]:
```
histograma =  sns.distplot(df['Idade'])
histograma.figure.set_size_inches(14,6)
histograma.set_title('Distribuição de Frêquencias - Idade', fontsize=18)
histograma.set_xlabel('Anos', fontsize=14)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[19]:
```
Text(0.5, 0, 'Anos')
```

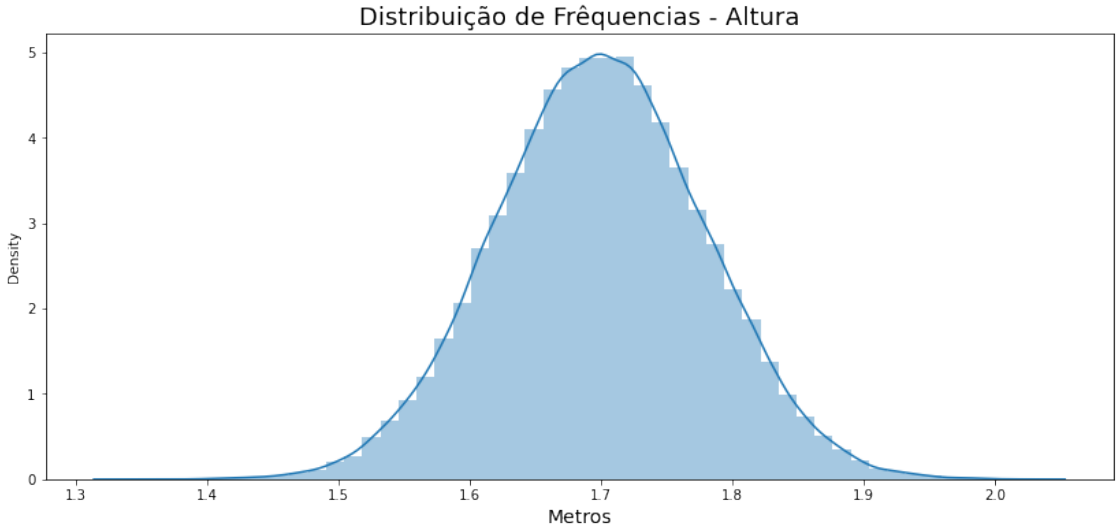



[20]:
```
histograma =  sns.distplot(df['Altura'])
histograma.figure.set_size_inches(14,6)
histograma.set_title('Distribuição de Frêquencias - Altura', fontsize=18)
histograma.set_xlabel('Metros', fontsize=14)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
```

```
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

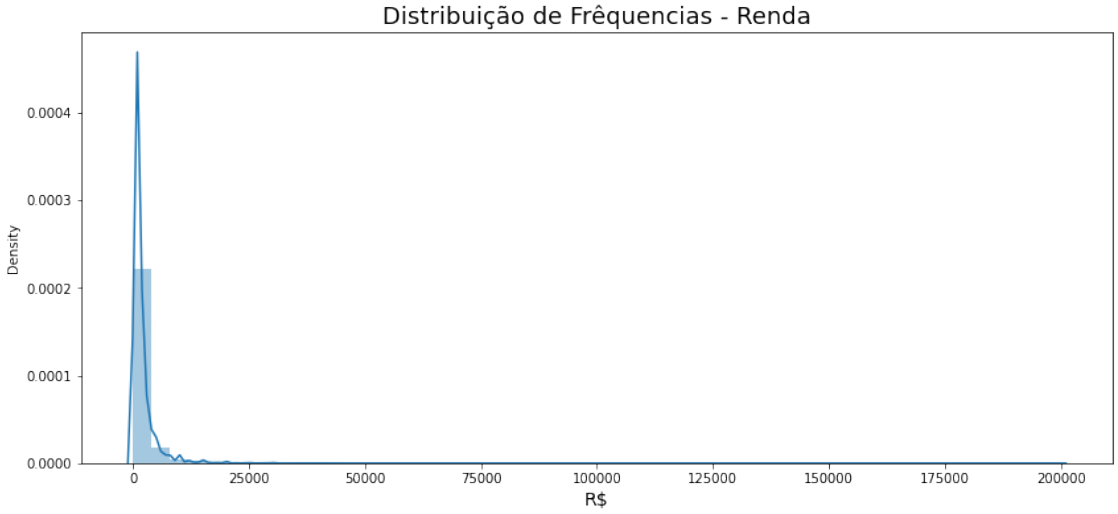
[20]:
```
Text(0.5, 0, 'Metros')
```




[21]:
```
histograma =  sns.distplot(df['Renda'])
histograma.figure.set_size_inches(14,6)
histograma.set_title('Distribuição de Frêquencias - Renda', fontsize=18)
histograma.set_xlabel('R$', fontsize=14)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[21]:
```
Text(0.5, 0, 'R$')
```




### 2.0.23 Conclusões

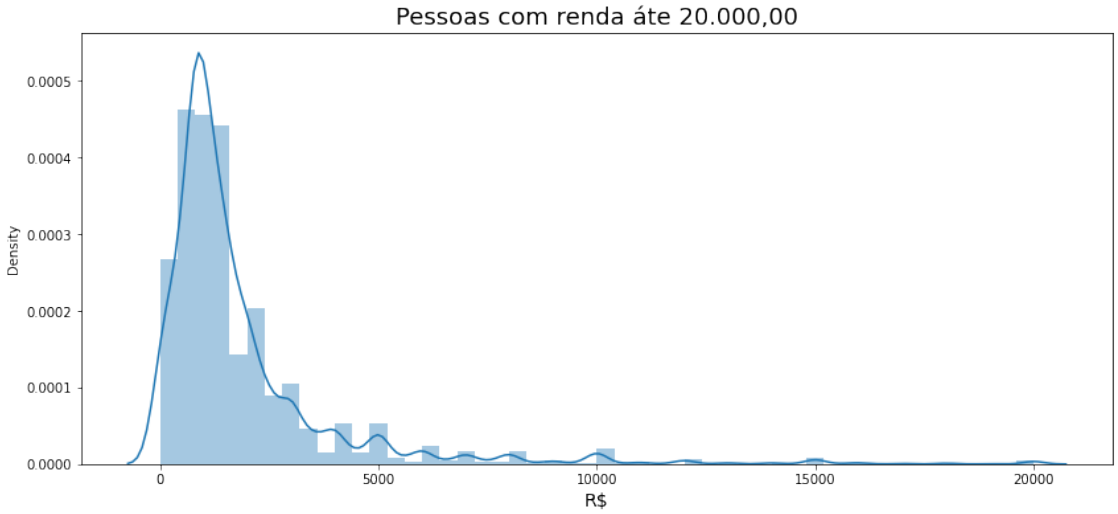
Podemo concluir que a idade está concetrada entre 20 a 60 anos, que a média das pessoas é de 1.70 e que a renda est concetrada em poucas pessoas.

### 2.0.24 Para a variável RENDA, construa um histograma somente com as informações das pessoas com rendimento até R$ 20.000,00

[22]:
```
histograma =  sns.distplot(df.query('Renda <= 20000')['Renda'])
histograma.figure.set_size_inches(14,6)
histograma.set_title('Pessoas com renda áte 20.000,00', fontsize=18)
histograma.set_xlabel('R$', fontsize=14)
```

```
C:\Users\alexsandro.ignacio\AppData\Local\Programs\Python\Python37\lib\site-
packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a
deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility)
or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[22]:
```
Text(0.5, 0, 'R$')
```




### 2.0.25 Construa uma tabela de frequências e uma com os percentuais do cruzando das variáveis SEXO e COR

**Avalie o resultado da tabela e escreva suas principais conclusões**

**Utilize os dicionários abaixo para renomear as linha e colunas das tabelas de frequências e dos gráficos em nosso projeto**

[23]:
```
sexo = {
    0: 'Masculino',
    1: 'Feminino'
}
cor = {
    0: 'Indígena',
    2: 'Branca',
    4: 'Preta',
    6: 'Amarela',
    8: 'Parda',
    9: 'Sem declaração'
}
anos_de_estudo = {
    1: 'Sem instrução e menos de 1 ano',
    2: '1 ano',
    3: '2 anos',
    4: '3 anos',
    5: '4 anos',
    6: '5 anos',
    7: '6 anos',
    8: '7 anos',
    9: '8 anos',
```

```
    10: '9 anos',
    11: '10 anos',
    12: '11 anos',
    13: '12 anos',
    14: '13 anos',
    15: '14 anos',
    16: '15 anos ou mais',
    17: 'Não determinados'
}
uf = {
    11: 'Rondônia',
    12: 'Acre',
    13: 'Amazonas',
    14: 'Roraima',
    15: 'Pará',
    16: 'Amapá',
    17: 'Tocantins',
    21: 'Maranhão',
    22: 'Piauí',
    23: 'Ceará',
    24: 'Rio Grande do Norte',
    25: 'Paraíba',
    26: 'Pernambuco',
    27: 'Alagoas',
    28: 'Sergipe',
    29: 'Bahia',
    31: 'Minas Gerais',
    32: 'Espírito Santo',
    33: 'Rio de Janeiro',
    35: 'São Paulo',
    41: 'Paraná',
    42: 'Santa Catarina',
    43: 'Rio Grande do Sul',
    50: 'Mato Grosso do Sul',
    51: 'Mato Grosso',
    52: 'Goiás',
    53: 'Distrito Federal'
}
```

[24]:
```
df.head(2)
```

[24]:

| | UF | Sexo | Idade | Cor | Anos de Estudo | Renda | Altura |
|---|---|---|---|---|---|---|---|
| 0 | 11 | 0 | 23 | 8 | 12 | 800 | 1.603808 |
| 1 | 11 | 1 | 23 | 2 | 12 | 1150 | 1.739790 |

[25]:
```
frequencia = pd.crosstab(df['Cor'], df['Sexo'])
frequencia.rename(columns=sexo,inplace=True)
```

```
frequencia.rename(index=cor,inplace=True)
frequencia
```

[25]:
| Sexo | Masculino | Feminino |
|---|---|---|
| Cor | | |
| Indígena | 256 | 101 |
| Branca | 22194 | 9621 |
| Preta | 5502 | 2889 |
| Amarela | 235 | 117 |
| Parda | 25063 | 10862 |

```
[26]: percentual = pd.crosstab(df['Cor'], df['Sexo'], normalize=True)
percentual.rename(columns=sexo,inplace=True)
percentual.rename(index=cor,inplace=True)
percentual
```

[26]:
| Sexo | Masculino | Feminino |
|---|---|---|
| Cor | | |
| Indígena | 0.003332 | 0.001314 |
| Branca | 0.288834 | 0.125208 |
| Preta | 0.071603 | 0.037598 |
| Amarela | 0.003058 | 0.001523 |
| Parda | 0.326171 | 0.141359 |

#### 2.0.26 Conclusões

Podemos indentificar que a população masculina é maior que a população feminina.

### 2.1 Realize, para a variável RENDA, uma análise descritiva com as ferramentas que aprendemos em nosso treinamento

#### 2.1.1 Obtenha a média aritimética

```
[27]: media = df['Renda'].mean()
media
```

[27]: 2000.3831988547631

#### 2.1.2 Obtenha a mediana

```
[28]: mediana = df['Renda'].median()
mediana
```

[28]: 1200.0

### 2.1.3 Obtenha a moda

[29]:
```
moda = df['Renda'].mode()
moda[0]
```

[29]:
```
788
```

### 2.1.4 Obtenha o desvio médio absoluto

[30]:
```
dma = df['Renda'].mad()
dma
```

[30]:
```
1526.4951371638058
```

### 2.1.5 Obtenha a variância

[31]:
```
var = df['Renda'].var()
var
```

[31]:
```
11044906.006217021
```

### 2.1.6 Obtenha o desvio-padrão

[32]:
```
desv_padrao = df['Renda'].std()
desv_padrao
```

[32]:
```
3323.3877303464037
```

### 2.1.7 Obtenha a média, mediana e valor máximo da variável RENDA segundo SEXO e COR

**Destaque os pontos mais importante que você observa nas tabulações**

**O parâmento aggfunc da função crosstab() pode receber uma lista de funções. Exemplo: aggfunc = {'mean', 'median', 'max'}**

[33]:
```
renda_sexo_e_cor = pd.crosstab(df['Cor'],
                               df['Sexo'],
                               values=df['Renda'],␣
 ↪aggfunc={'mean','median','max'})
renda_sexo_e_cor.rename(index=cor, inplace=True)
renda_sexo_e_cor.rename(columns=sexo, inplace=True)

renda_sexo_e_cor
```

[33]:

| | max | | mean | | median | |
|---|---|---|---|---|---|---|
| Sexo | Masculino | Feminino | Masculino | Feminino | Masculino | Feminino |
| Cor | | | | | | |
| Indígena | 10000.0 | 120000.0 | 1081.710938 | 2464.386139 | 797.5 | 788.0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Branca | 200000.0 | 100000.0 | 2925.744435 | 2109.866750 | 1700.0 | 1200.0 |
| Preta | 50000.0 | 23000.0 | 1603.861687 | 1134.596400 | 1200.0 | 800.0 |
| Amarela | 50000.0 | 20000.0 | 4758.251064 | 3027.341880 | 2800.0 | 1500.0 |
| Parda | 100000.0 | 30000.0 | 1659.577425 | 1176.758516 | 1200.0 | 800.0 |

#### 2.1.8 Conclusões

A renda do homem branco é maior que as demais.

#### 2.1.9 Obtenha as medidas de dispersão da variável RENDA segundo SEXO e COR

**Destaque os pontos mais importante que você observa nas tabulações**

**O parâmento aggfunc da função crosstab() pode receber uma lista de funções. Exemplo: aggfunc = {'mad', 'var', 'std'}**

[34]:
```
renda_sexo_e_cor = pd.crosstab(df['Cor'],
                               df['Sexo'],
                               values=df['Renda'], aggfunc={'mad', 'var',␣
↪'std'})
renda_sexo_e_cor.rename(index=cor, inplace=True)
renda_sexo_e_cor.rename(columns=sexo, inplace=True)

renda_sexo_e_cor.round(2)
```

[34]:

| | mad | | std | | var | |
|---|---|---|---|---|---|---|
| Sexo | Masculino | Feminino | Masculino | Feminino | Masculino | Feminino |
| Cor | | | | | | |
| Indígena | 798.91 | 3007.89 | 1204.09 | 11957.50 | 1449841.13 | 1.429818e+08 |
| Branca | 2261.01 | 1670.97 | 4750.79 | 3251.01 | 22570023.41 | 1.056909e+07 |
| Preta | 975.60 | 705.45 | 1936.31 | 1349.80 | 3749293.59 | 1.821960e+06 |
| Amarela | 3709.60 | 2549.15 | 5740.82 | 3731.17 | 32957069.62 | 1.392166e+07 |
| Parda | 1125.83 | 811.58 | 2312.09 | 1596.23 | 5345747.15 | 2.547960e+06 |

#### 2.1.10 Conclusões

A maior variância se encontra na homem pardo.

#### 2.1.11 Construa um box plot da variável RENDA segundo SEXO e COR

**É possível verificar algum comportamento diferenciado no rendimento entre os grupos de pessoas analisados? Avalie o gráfico e destaque os pontos mais importantes.**

**1º - Utilize somente as informações de pessoas com renda abaixo de R$ 10.000**

**2º - Para incluir uma terceira variável na construção de um boxplot utilize o parâmetro hue e indique a variável que quer incluir na subdivisão.**

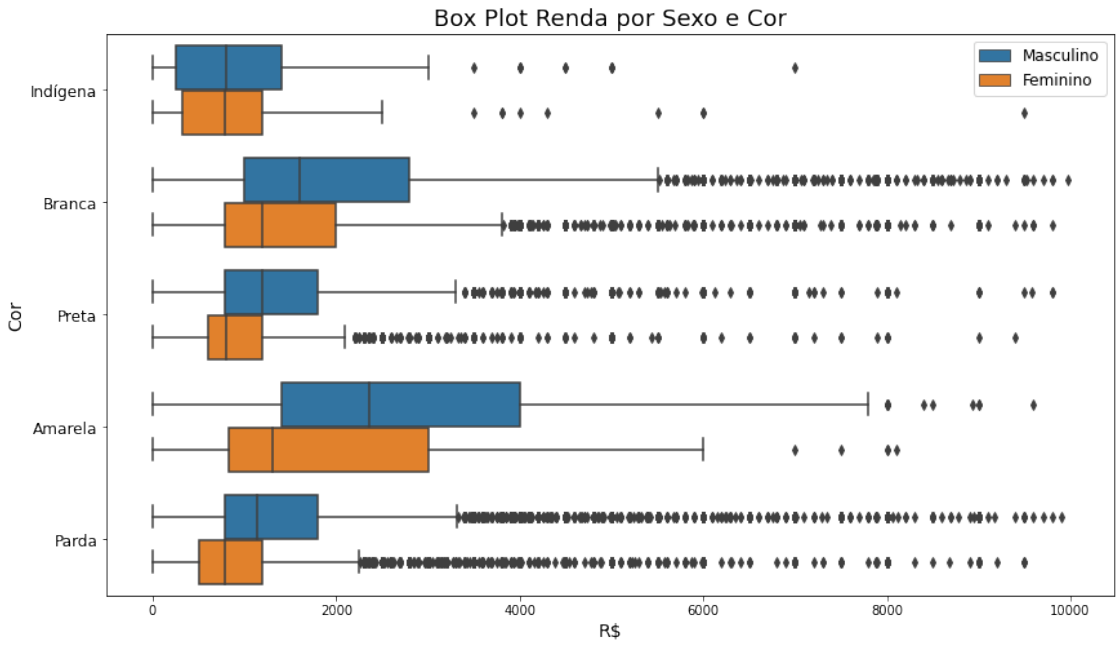
**Mais informações: https://seaborn.pydata.org/generated/seaborn.boxplot.html**

[35]:
```
boxplot = sns.boxplot(x= 'Renda', y='Cor', hue='Sexo',data = df.query('Renda <␣
↪10000'), orient='h')
boxplot.figure.set_size_inches(14, 8) # Tamanho
boxplot.set_title('Box Plot Renda por Sexo e Cor',fontsize=18) # Título
boxplot.set_xlabel('R$',fontsize=14) # Etiqueta do Eixo X
boxplot.set_ylabel('Cor',fontsize=14) # Etiqueta do Eixo Y

boxplot.
↪set_yticklabels(['Indígena','Branca','Preta','Amarela','Parda'],fontsize=12)

# Configurando Legenda
handles, _ = boxplot.get_legend_handles_labels()
boxplot.legend(handles,['Masculino', 'Feminino'], fontsize=12)
boxplot
```

[35]:
```
<AxesSubplot:title={'center':'Box Plot Renda por Sexo e Cor'}, xlabel='R$',
ylabel='Cor'>
```




### 2.1.12 Conclusões

Podemos concluír que a deferença entre sexos existe entre todas as raças, porém podemos observar que na raça índigena a essa diferença é muito pouca.

# 3 DESAFIO

### 3.0.1 Qual percentual de pessoas de nosso dataset ganham um salário mínimo (R$ 788,00) ou menos?

**Utilize a função percentileofscore() do scipy para realizar estas análises.**

**Mais informações: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.percentileofsco**

[36]:
```
df.head(2)
```

[36]:

| | UF | Sexo | Idade | Cor | Anos de Estudo | Renda | Altura |
|---|---|---|---|---|---|---|---|
| 0 | 11 | 0 | 23 | 8 | 12 | 800 | 1.603808 |
| 1 | 11 | 1 | 23 | 2 | 12 | 1150 | 1.739790 |

[37]:
```
from scipy import stats
```

[38]:
```
percentual = stats.percentileofscore(df['Renda'], 788, kind='weak')
print("{0:.2f}%".format(percentual))
```

```
28.87%
```

### 3.0.2 Qual o valor máximo ganho por 99% das pessoas de nosso dataset?

**Utilize o método quantile() do pandas para realizar estas análises.**

[39]:
```
valor = df['Renda'].quantile(.99)
print('R$ {0:.2f}'.format(valor))
print(f'R$ {valor:.2f}')
```

```
R$ 15000.00
R$ 15000.00
```

### 3.0.3 Obtenha a média, mediana, valor máximo e desvio-padrão da variável RENDA segundo ANOS DE ESTUDO e SEXO

**Destaque os pontos mais importante que você observa nas tabulações**

**O parâmento aggfunc da função crosstab() pode receber uma lista de funções. Exemplo: aggfunc = [‘mean’, ‘median’, ‘max’, ‘std’]**

[40]:
```
renda_sexo_e_anos = pd.crosstab(df['Anos de Estudo'],
                                df['Sexo'],
                                values=df['Renda'], aggfunc={'mean', 'median',␣
 ↪'max', 'std'})
renda_sexo_e_anos.rename(index=anos_de_estudo, inplace=True)
renda_sexo_e_anos.rename(columns=sexo, inplace=True)

renda_sexo_e_anos.round(2)
```

[40]:

| | max | | mean | |
|---|---|---|---|---|
| Sexo | Masculino | Feminino | Masculino | Feminino |
| Anos de Estudo | | | | |
| Sem instrução e menos de 1 ano | 30000.0 | 10000.0 | 799.49 | 516.20 |

| | | | | |
|---|---|---|---|---|
| 1 ano | 30000.0 | 2000.0 | 895.63 | 492.77 |
| 2 anos | 40000.0 | 4000.0 | 931.18 | 529.91 |
| 3 anos | 80000.0 | 3500.0 | 1109.20 | 546.85 |
| 4 anos | 50000.0 | 10000.0 | 1302.33 | 704.28 |
| 5 anos | 35000.0 | 8000.0 | 1338.65 | 781.39 |
| 6 anos | 25000.0 | 6000.0 | 1448.88 | 833.73 |
| 7 anos | 40000.0 | 9000.0 | 1465.50 | 830.75 |
| 8 anos | 30000.0 | 18000.0 | 1639.40 | 933.62 |
| 9 anos | 60000.0 | 20000.0 | 1508.04 | 868.02 |
| 10 anos | 45000.0 | 6000.0 | 1731.27 | 925.92 |
| 11 anos | 200000.0 | 100000.0 | 2117.06 | 1286.79 |
| 12 anos | 30000.0 | 120000.0 | 2470.33 | 1682.31 |
| 13 anos | 25000.0 | 20000.0 | 3195.10 | 1911.73 |
| 14 anos | 50000.0 | 20000.0 | 3706.62 | 2226.46 |
| 15 anos ou mais | 200000.0 | 100000.0 | 6134.28 | 3899.51 |
| Não determinados | 7000.0 | 3000.0 | 1295.76 | 798.17 |

| | median | | std | |
|---|---|---|---|---|
| Sexo | Masculino | Feminino | Masculino | Feminino |
| Anos de Estudo | | | | |
| Sem instrução e menos de 1 ano | 700.0 | 390.0 | 1023.90 | 639.31 |
| 1 ano | 788.0 | 400.0 | 1331.95 | 425.29 |
| 2 anos | 788.0 | 450.0 | 1435.17 | 498.23 |
| 3 anos | 800.0 | 500.0 | 2143.80 | 424.12 |
| 4 anos | 1000.0 | 788.0 | 1419.82 | 629.55 |
| 5 anos | 1045.0 | 788.0 | 1484.65 | 635.78 |
| 6 anos | 1200.0 | 788.0 | 1476.63 | 574.55 |
| 7 anos | 1200.0 | 788.0 | 1419.71 | 602.04 |
| 8 anos | 1300.0 | 800.0 | 1515.58 | 896.78 |
| 9 anos | 1200.0 | 788.0 | 2137.66 | 973.22 |
| 10 anos | 1218.0 | 800.0 | 2078.61 | 620.61 |
| 11 anos | 1500.0 | 1000.0 | 2676.54 | 1819.04 |
| 12 anos | 1800.0 | 1200.0 | 2268.08 | 4851.83 |
| 13 anos | 2400.0 | 1300.0 | 2797.12 | 2053.79 |
| 14 anos | 2500.0 | 1600.0 | 3987.21 | 2064.08 |
| 15 anos ou mais | 4000.0 | 2800.0 | 7447.61 | 4212.77 |
| Não determinados | 1200.0 | 788.0 | 979.65 | 459.99 |

#### 3.0.4 Construa um box plot da variável RENDA segundo ANOS DE ESTUDO e SEXO

**É possível verificar algum comportamento diferenciado no rendimento entre os grupos de pessoas analisados? Avalie o gráfico e destaque os pontos mais importantes.**

**1º - Utilize somente as informações de pessoas com renda abaixo de R$ 10.000**

**2º - Utilize a variável IDADE para identificar se a desigualdade se verifica para pessoas de mesma idade. Exemplo: data=dados.query('Renda < 10000 and Idade == 40') ou data=dados.query('Renda < 10000 and Idade == 50')**

**3º - Para incluir uma terceira variável na construção de um boxplot utilize o parâmetro hue e indique a variável que quer incluir na subdivisão.**

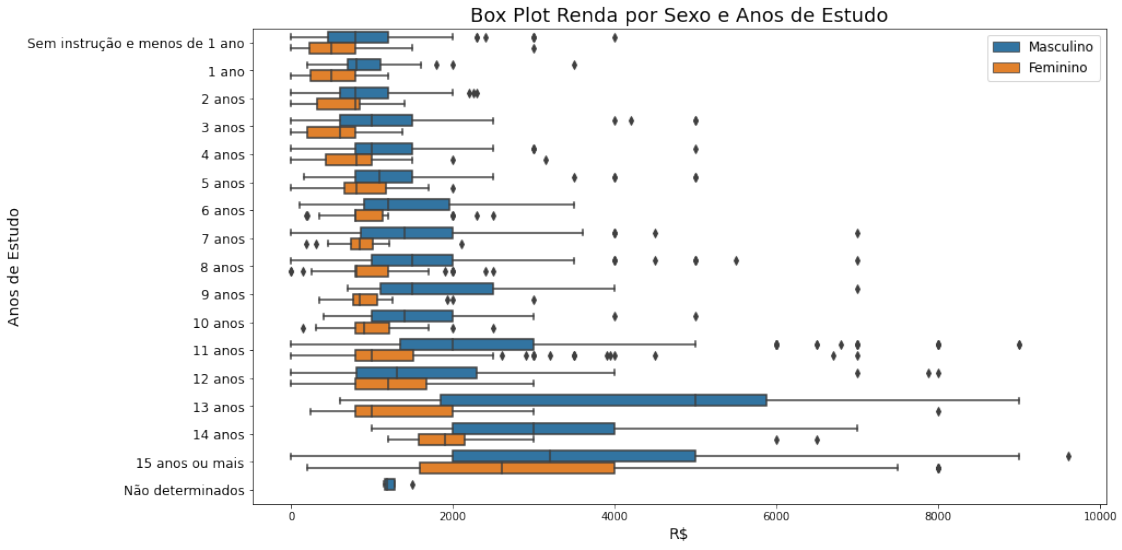
**Mais informações: https://seaborn.pydata.org/generated/seaborn.boxplot.html**

[41]:
```
boxplot = sns.boxplot(x= 'Renda', y='Anos de Estudo', hue='Sexo',data = df.
↪query('Renda < 10000 and Idade ==50'), orient='h')
boxplot.figure.set_size_inches(14, 8) # Tamanho
boxplot.set_title('Box Plot Renda por Sexo e Anos de Estudo',fontsize=18) #␣
↪Título
boxplot.set_xlabel('R$',fontsize=14) # Etiqueta do Eixo X
boxplot.set_ylabel('Anos de Estudo',fontsize=14) # Etiqueta do Eixo Y

boxplot.set_yticklabels([key for key in anos_de_estudo.values()],fontsize=12)

# Configurando Legenda
handles, _ = boxplot.get_legend_handles_labels()
boxplot.legend(handles,['Masculino', 'Feminino'], fontsize=12)
boxplot
```

[41]:
```
<AxesSubplot:title={'center':'Box Plot Renda por Sexo e Anos de Estudo'},
xlabel='R$', ylabel='Anos de Estudo'>
```




### 3.0.5 Conclusões

podemos concluír que a renda aumanta conforme os anos de estudo.

### 3.0.6 Obtenha a média, mediana, valor máximo e desvio-padrão da variável RENDA segundo as UNIDADES DA FEDERAÇÃO

**Destaque os pontos mais importante que você observa nas tabulações**

**Utilize o método groupby() do pandas juntamente com o método agg() para contruir a tabulação. O método agg() pode receber um dicionário especificando qual coluna do DataFrame deve ser utilizada e qual lista de funções estatísticas queremos obter, por exemplo: dados.groupby(['UF']).agg({'Renda': ['mean', 'median', 'max', 'std']})**

[42]:
```
renda_uf = df.groupby(['UF']).agg({'Renda': ['mean', 'median', 'max']})
renda_uf.rename(index=uf)
```

[42]:

| | Renda | | |
|---|---|---|---|
| | mean | median | max |
| UF | | | |
| Rondônia | 1789.761223 | 1200 | 50000 |
| Acre | 1506.091782 | 900 | 30000 |
| Amazonas | 1445.130100 | 900 | 22000 |
| Roraima | 1783.588889 | 1000 | 20000 |
| Pará | 1399.076871 | 850 | 50000 |
| Amapá | 1861.353516 | 1200 | 15580 |
| Tocantins | 1771.094946 | 1000 | 60000 |
| Maranhão | 1019.432009 | 700 | 30000 |
| Piauí | 1074.550784 | 750 | 40000 |
| Ceará | 1255.403692 | 789 | 25000 |
| Rio Grande do Norte | 1344.721480 | 800 | 15500 |
| Paraíba | 1293.370487 | 788 | 30000 |
| Pernambuco | 1527.079319 | 900 | 50000 |
| Alagoas | 1144.552602 | 788 | 11000 |
| Sergipe | 1109.111111 | 788 | 16000 |
| Bahia | 1429.645094 | 800 | 200000 |
| Minas Gerais | 2056.432084 | 1200 | 100000 |
| Espírito Santo | 2026.383852 | 1274 | 100000 |
| Rio de Janeiro | 2496.403168 | 1400 | 200000 |
| São Paulo | 2638.104986 | 1600 | 80000 |
| Paraná | 2493.870753 | 1500 | 200000 |
| Santa Catarina | 2470.854945 | 1800 | 80000 |
| Rio Grande do Sul | 2315.158336 | 1500 | 35000 |
| Mato Grosso do Sul | 2262.604167 | 1500 | 42000 |
| Mato Grosso | 2130.652778 | 1500 | 35000 |
| Goiás | 1994.580794 | 1500 | 30000 |
| Distrito Federal | 4241.954722 | 2000 | 100000 |

### 3.0.7 Construa um box plot da variável RENDA segundo as UNIDADES DA FEDERAÇÃO

**É possível verificar algum comportamento diferenciado no rendimento entre os grupos analisados? Avalie o gráfico e destaque os pontos mais importantes.**

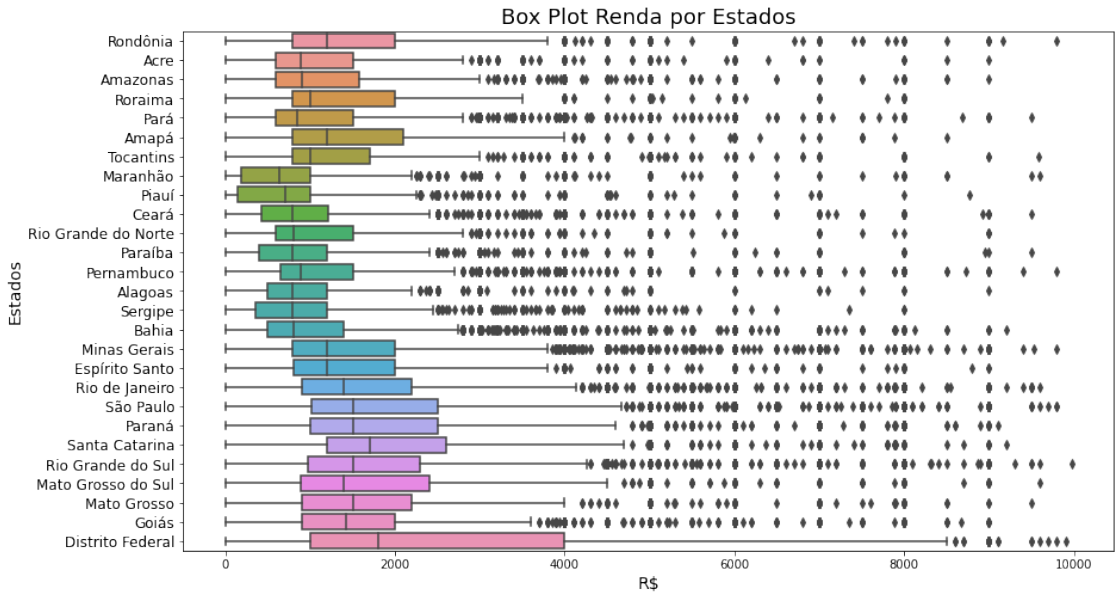
**1º - Utilize somente as informações de pessoas com renda abaixo de R$ 10.000**

[43]:
```
boxplot = sns.boxplot(x= 'Renda', y='UF', data = df.query('Renda < 10000'),␣
↪orient='h')
boxplot.figure.set_size_inches(14, 8) # Tamanho
boxplot.set_title('Box Plot Renda por Estados',fontsize=18) # Título
boxplot.set_xlabel('R$',fontsize=14) # Etiqueta do Eixo X
boxplot.set_ylabel('Estados',fontsize=14) # Etiqueta do Eixo Y

boxplot.set_yticklabels([key for key in uf.values()],fontsize=12)

boxplot
```

[43]:
```
<AxesSubplot:title={'center':'Box Plot Renda por Estados'}, xlabel='R$',
ylabel='Estados'>
```



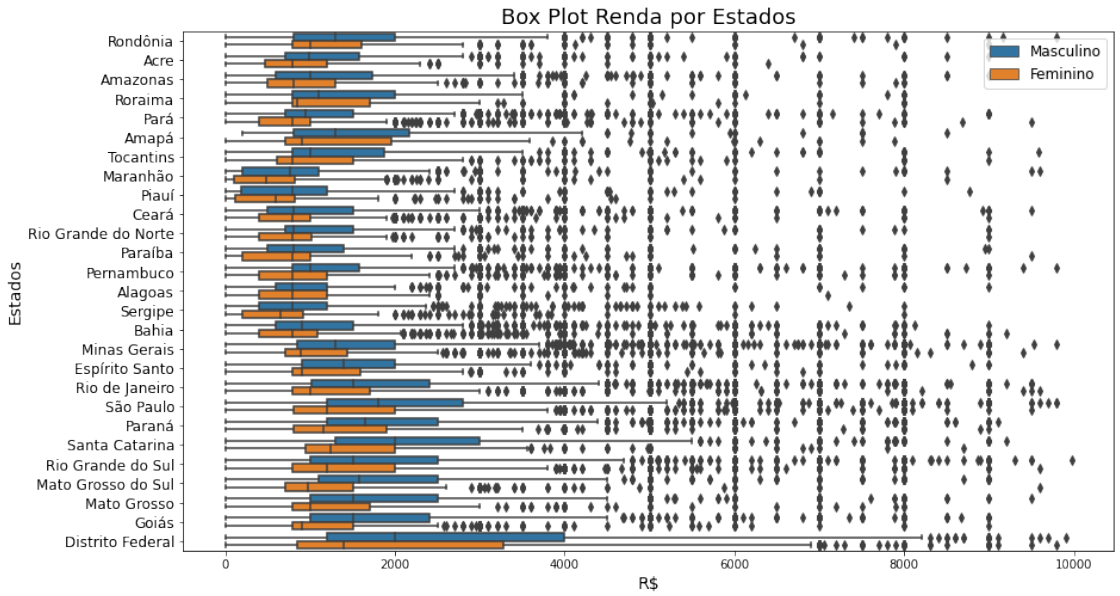

[44]:
```
boxplot = sns.boxplot(x= 'Renda', hue='Sexo',  y='UF', data = df.query('Renda <␣
↪10000'), orient='h')
boxplot.figure.set_size_inches(14, 8) # Tamanho
boxplot.set_title('Box Plot Renda por Estados',fontsize=18) # Título
boxplot.set_xlabel('R$',fontsize=14) # Etiqueta do Eixo X
boxplot.set_ylabel('Estados',fontsize=14) # Etiqueta do Eixo Y

boxplot.set_yticklabels([key for key in uf.values()],fontsize=12)

# Configurando Legenda
handles, _ = boxplot.get_legend_handles_labels()
boxplot.legend(handles,['Masculino', 'Feminino'], fontsize=12)
```

```
boxplot
```

[44]: <AxesSubplot:title={'center':'Box Plot Renda por Estados'}, xlabel='R$', ylabel='Estados'>




### 3.0.8 Conclusões

Podemos identificar que as maiores rendas estão no distrito federal.