In [1]:

```
from IPython.display import HTML

HTML('''<script>
code_show=true;
function code_toggle() {
 if (code_show){
 $('div.input').hide();
 } else {
 $('div.input').show();
 }
 code_show = !code_show
}
$( document ).ready(code_toggle);
</script>''')
```

Out[1]:

In [3]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:

```
sns.set_theme(style="ticks", palette="pastel")
```

Dataframe

In [7]:

```
df.head()
```

Out[7]:

	finess	mois	annee	sexe	ghm2	GHS	age	duree	supp_rea	supp_si	...	population_region	Libel GHI
0	10007987	6	2012	1	20Z042	7268	36	9	0	0	...	5 542 094	Ethylism ave dépendanc niveau
1	10007987	10	2015	1	20Z042	7268	39	8	0	0	...	5 542 094	Ethylism ave dépendanc niveau
2	10007987	7	2015	1	20Z042	7268	50	10	0	0	...	5 542 094	Ethylism ave dépendanc niveau
3	10780054	1	2015	1	20Z042	7268	71	11	0	0	...	5 542 094	Ethylism ave dépendanc niveau
4	10780054	7	2017	1	20Z042	7282	21	14	0	0	...	5 542 094	Ethylism ave dépendanc niveau

5 rows × 45 columns

In [8]:
```
df_c = df[df['Libellé GHM']=='Hémodialyse, en séances']
```

In [11]:
```
df_c
```

Out[11]:

	finess	mois	annee	sexe	ghm2	GHS	age	duree	supp_rea	supp_si	...	population_region	Li
2014555	10780054	5	2012	2	28Z04Z	9605	48	0	0	0	...	5 542 094	H
2014556	10780054	6	2012	2	28Z04Z	9605	48	0	0	0	...	5 542 094	H
2014557	10780054	6	2012	2	28Z04Z	9605	48	0	0	0	...	5 542 094	H
2014558	10780054	6	2012	2	28Z04Z	9605	48	0	0	0	...	5 542 094	H
2014559	10780054	4	2013	2	28Z04Z	9605	49	0	0	0	...	5 542 094	H
...	...	...	...	...	...	...	...	...	...	...	...	...	
2341751	970421004	1	2011	1	28Z04Z	9617	16	0	0	0	...	868 846	H
2341752	970421004	8	2011	2	28Z04Z	9605	23	0	0	0	...	868 846	H
2341753	970421004	1	2011	1	28Z04Z	9617	10	0	0	0	...	868 846	H
2341754	970421004	1	2011	1	28Z04Z	9617	16	0	0	0	...	868 846	H
2341755	970421004	6	2011	2	28Z04Z	9617	11	0	0	0	...	868 846	H

327201 rows × 45 columns

In [12]:
```
df_c['cost']
```

Out[12]:
```
2014555    318.771289
2014556    318.771289
2014557    318.771289
2014558    318.771289
2014559    318.771289
              ...
2341751    318.771300
2341752    318.771300
2341753    318.771300
2341754    318.771300
```

```
2341755    318.771300
Name: cost, Length: 327201, dtype: float64
```

Cost

In [18]:

```
df_c['cost'].describe()
```

Out[18]:

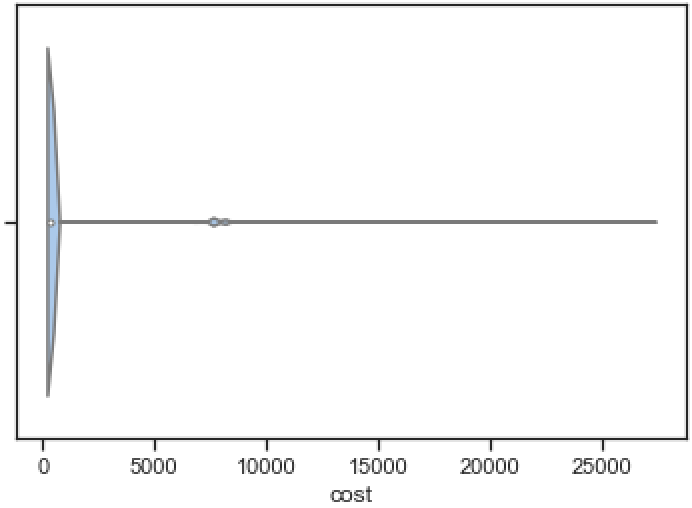
```
count    327201.000000
mean        407.665070
std         826.636061
min         318.771289
25%         318.771289
50%         318.771289
75%         318.771289
max       27245.843871
Name: cost, dtype: float64
```

In [14]:

```
sns.violinplot(data=df_c,x='cost')
```

Out[14]:

```
<AxesSubplot:xlabel='cost'>
```



In [15]:

```
df_c[df_c['cost'] > (5000)]['cost']
```

Out[15]:

```
2017192    7934.508989
2017194    8206.499621
2017195    8206.499621
2017197    7934.508989
2017198    8206.499621
              ...
2341463    7662.518356
2341470    7662.518356
2341471    7662.518356
2341472    7390.527724
2341473    6846.546460
Name: cost, Length: 3672, dtype: float64
```

In [39]:

```
df_c_trim = df_c[(df_c['cost']<((600)))&(df_c['cost']>300)]
```

In [40]:

```
df_c_trim['cost'].describe()
```

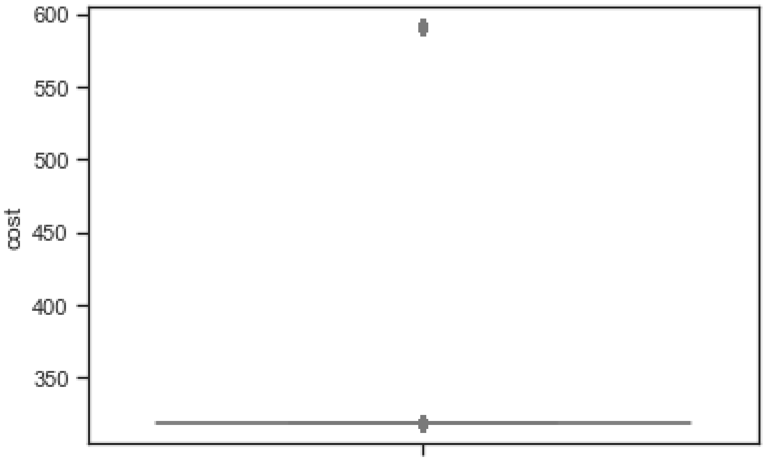
Out[40]:

```
count    323052.000000
mean        318.815070
std           3.450525
min         318.771289
25%         318.771289
50%         318.771289
75%         318.771289
```

```
max          590.761921
Name: cost, dtype: float64
```

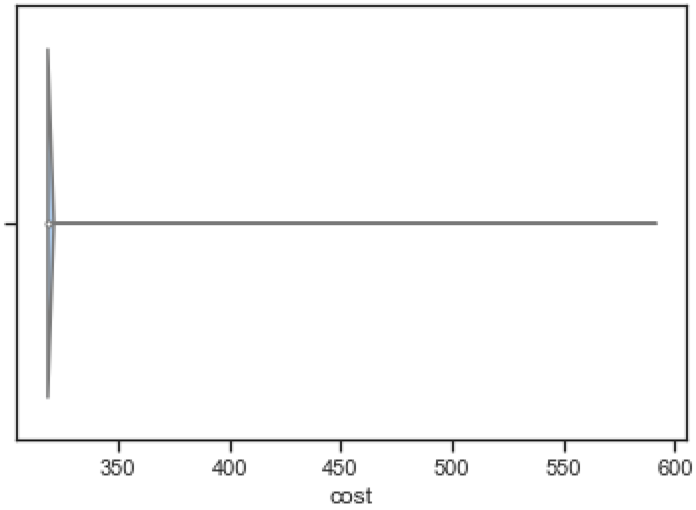
In [41]:
```
sns.boxplot(data=df_c_trim,y='cost')
```

Out[41]:
```
<AxesSubplot:ylabel='cost'>
```



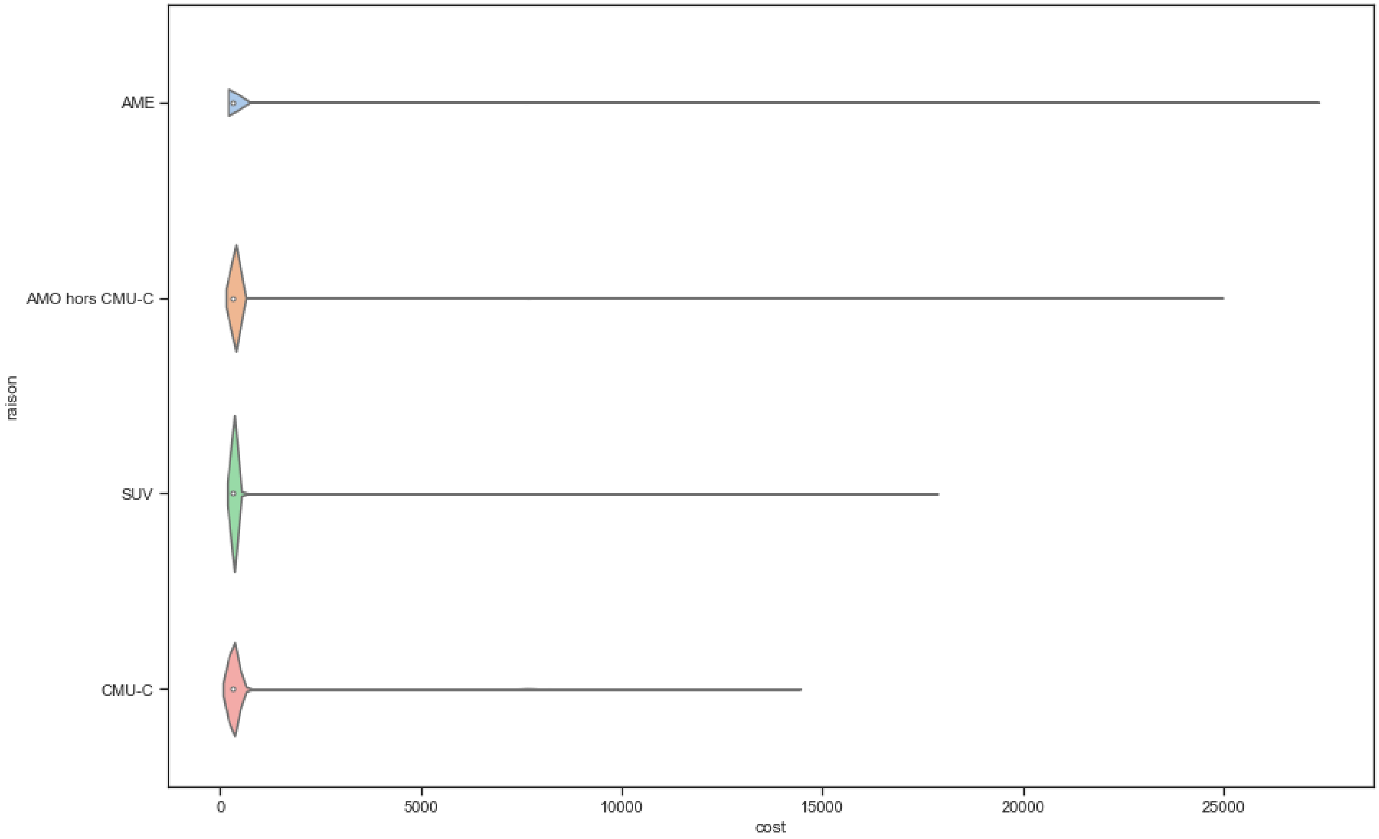
In [42]:
```
sns.violinplot(data=df_c_trim,x='cost')
```

Out[42]:
```
<AxesSubplot:xlabel='cost'>
```



In [43]:
```
plt.figure(figsize=(15,10))
sns.violinplot(data=df_c,x='cost',y='raison')
```

Out[43]:
```
<AxesSubplot:xlabel='cost', ylabel='raison'>
```

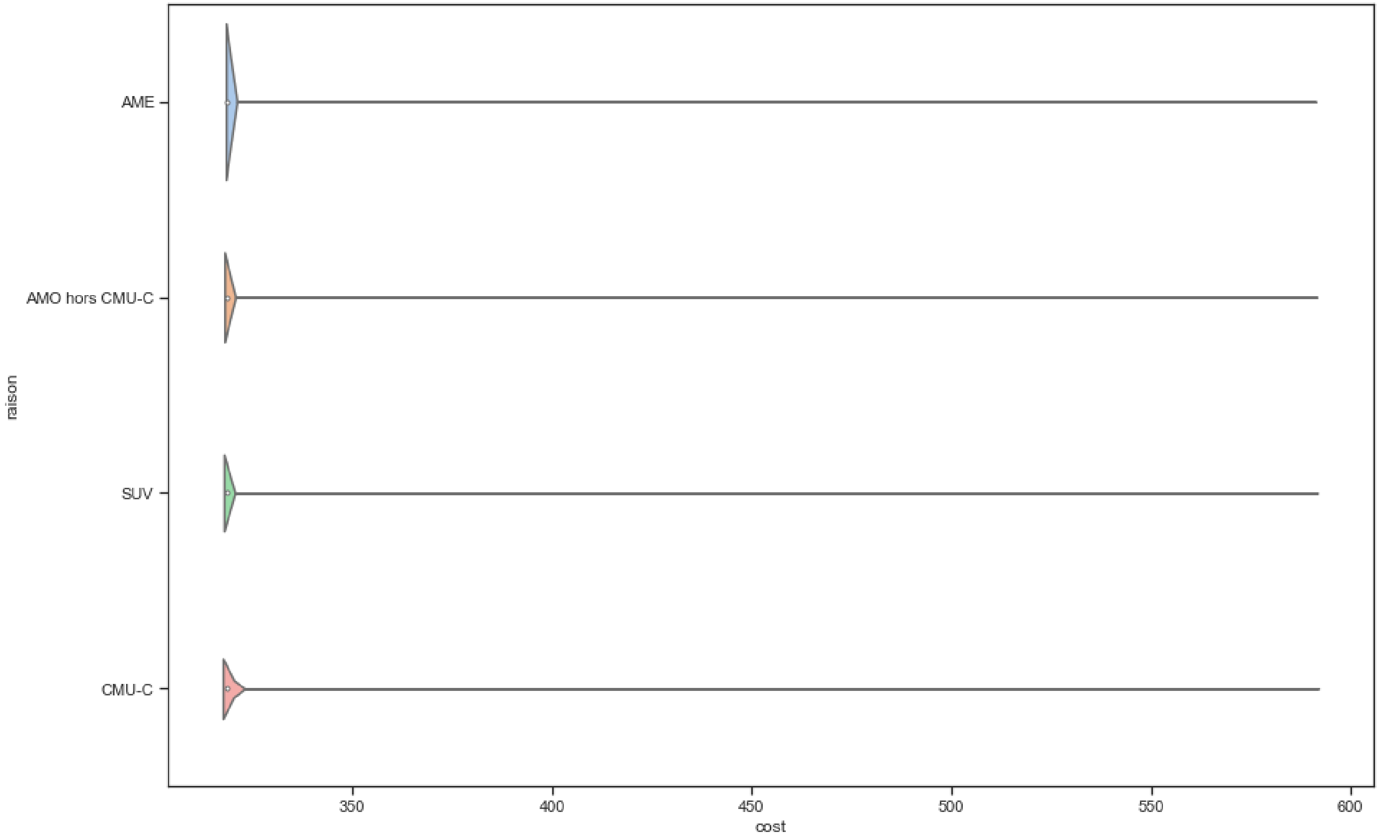


In [44]:

```
plt.figure(figsize=(15,10))
sns.violinplot(data=df_c_trim,x='cost',y='raison')
```

Out[44]:

```
<AxesSubplot:xlabel='cost', ylabel='raison'>
```



In [45]:

```
df_c[df_c['raison']=='SUV']['cost'].describe()
```

Out[45]:

```
count    19204.000000
mean       362.549827
std        548.837252
min        318.771289
```

```
25%        318.771289
50%        318.771289
75%        318.771289
max      17726.171746
Name: cost, dtype: float64
```

In [46]:
```
df_c[df_c['raison']=='AMO hors CMU-C']['cost'].describe()
```

Out[46]:
```
count    106888.000000
mean        431.880184
std         954.173946
min         318.771289
25%         318.771289
50%         318.771289
75%         318.771289
max       24797.928181
Name: cost, dtype: float64
```

In [47]:
```
df_c[df_c['raison']=='CMU-C']['cost'].describe()
```

Out[47]:
```
count    32501.000000
mean       472.771830
std       1037.388130
min        318.771289
25%        318.771289
50%        318.771289
75%        318.771289
max      14190.293528
Name: cost, dtype: float64
```

In [48]:
```
df_c[df_c['raison']=='AME']['cost'].describe()
```

Out[48]:
```
count    168608.000000
mean        384.902528
std         710.784340
min         318.771289
25%         318.771289
50%         318.771289
75%         318.771289
max       27245.843871
Name: cost, dtype: float64
```

In [49]:
```
def age_class(age):
    if age in list(range(0,18)):
        return '0 à 17 ans'
    elif age in list(range(18,36)):
        return '18 à 35 ans'
    elif age in list(range(36,54)):
        return '36 à 53 ans'
    elif age in list(range(54,72)):
        return '54 à 71 ans'
    elif age in list(range(72,90)):
        return '72 à 89 ans'
    else:
        return '90 ans et plus'
```

In [50]:
```
df_c['age_class'] = df_c['age'].apply(age_class)
```

```
<ipython-input-50-64996861a71c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
```

```
guide/indexing.html#returning-a-view-versus-a-copy
  df_c['age_class'] = df_c['age'].apply(age_class)
```

In [ ]:

In [51]:

```
df_c.sort_values(by='age_class',inplace=True)
```

```
<ipython-input-51-7d14fcbfc09c>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df_c.sort_values(by='age_class',inplace=True)
```

In [66]:

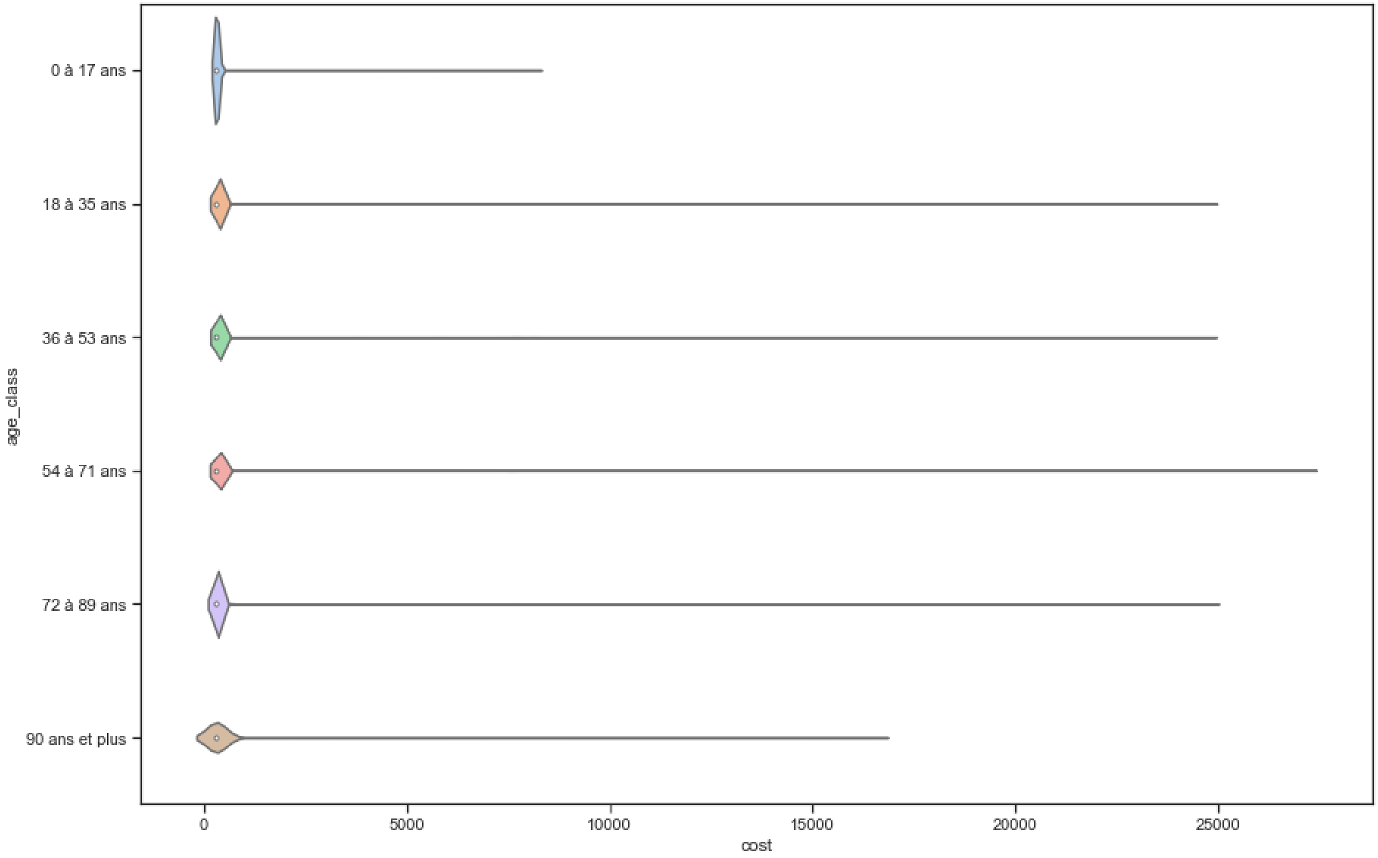
```
df_c_trim = df_c[(df_c['cost']<(((1000))))&(df_c['cost']>200)]
```

In [64]:

```
plt.figure(figsize=(15,10))
sns.violinplot(data=df_c,x='cost',y='age_class')
```

Out[64]:

```
<AxesSubplot:xlabel='cost', ylabel='age_class'>
```

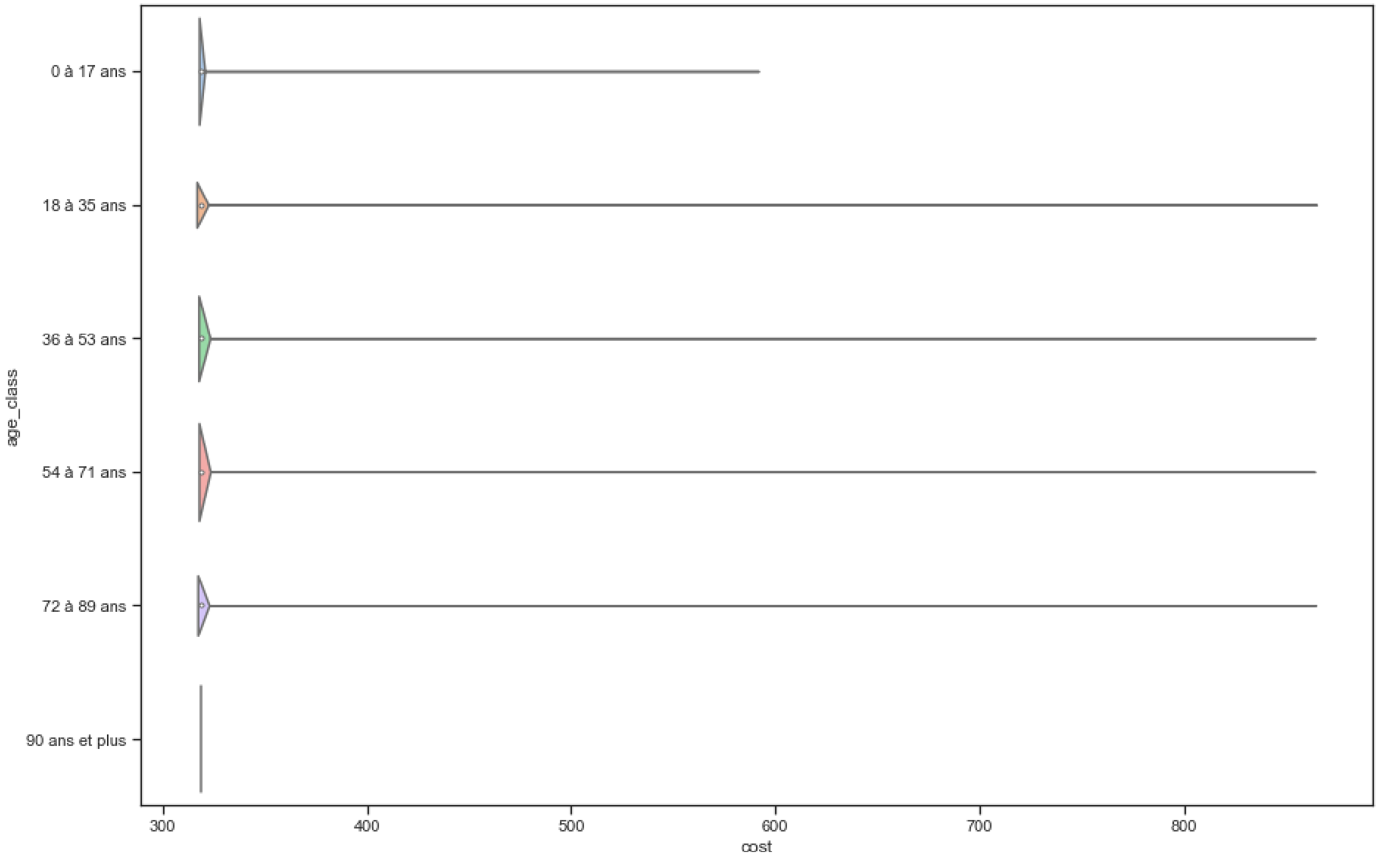


In [67]:

```
plt.figure(figsize=(15,10))
sns.violinplot(data=df_c_trim,x='cost',y='age_class')
```

Out[67]:

```
<AxesSubplot:xlabel='cost', ylabel='age_class'>
```



```
In [68]: df_c[df_c['age_class']=='0 à 17 ans']['cost'].describe()
```

```
Out[68]: count    10231.000000
         mean       337.939038
         std        375.483623
         min        318.771289
         25%        318.771289
         50%        318.771289
         75%        318.771289
         max       8206.499621
         Name: cost, dtype: float64
```

```
In [69]: df_c[df_c['age_class']=='18 à 35 ans']['cost'].describe()
```

```
Out[69]: count    45350.000000
         mean       383.023453
         std        683.588442
         min        318.771289
         25%        318.771289
         50%        318.771289
         75%        318.771289
         max      24797.928181
         Name: cost, dtype: float64
```

```
In [70]: df_c[df_c['age_class']=='36 à 53 ans']['cost'].describe()
```

```
Out[70]: count    86482.000000
         mean       395.966660
         std        749.824789
         min        318.771289
         25%        318.771289
         50%        318.771289
         75%        318.771289
         max      24797.928181
         Name: cost, dtype: float64
```

```
In [71]: df_c[df_c['age_class']=='54 à 71 ans']['cost'].describe()
```

```
Out[71]: count    109712.000000
         mean        407.844056
         std         837.837540
         min         318.771289
         25%         318.771289
         50%         318.771289
         75%         318.771289
         max       27245.843871
         Name: cost, dtype: float64
```

```
In [72]: df_c[df_c['age_class']=='72 à 89 ans']['cost'].describe()
```

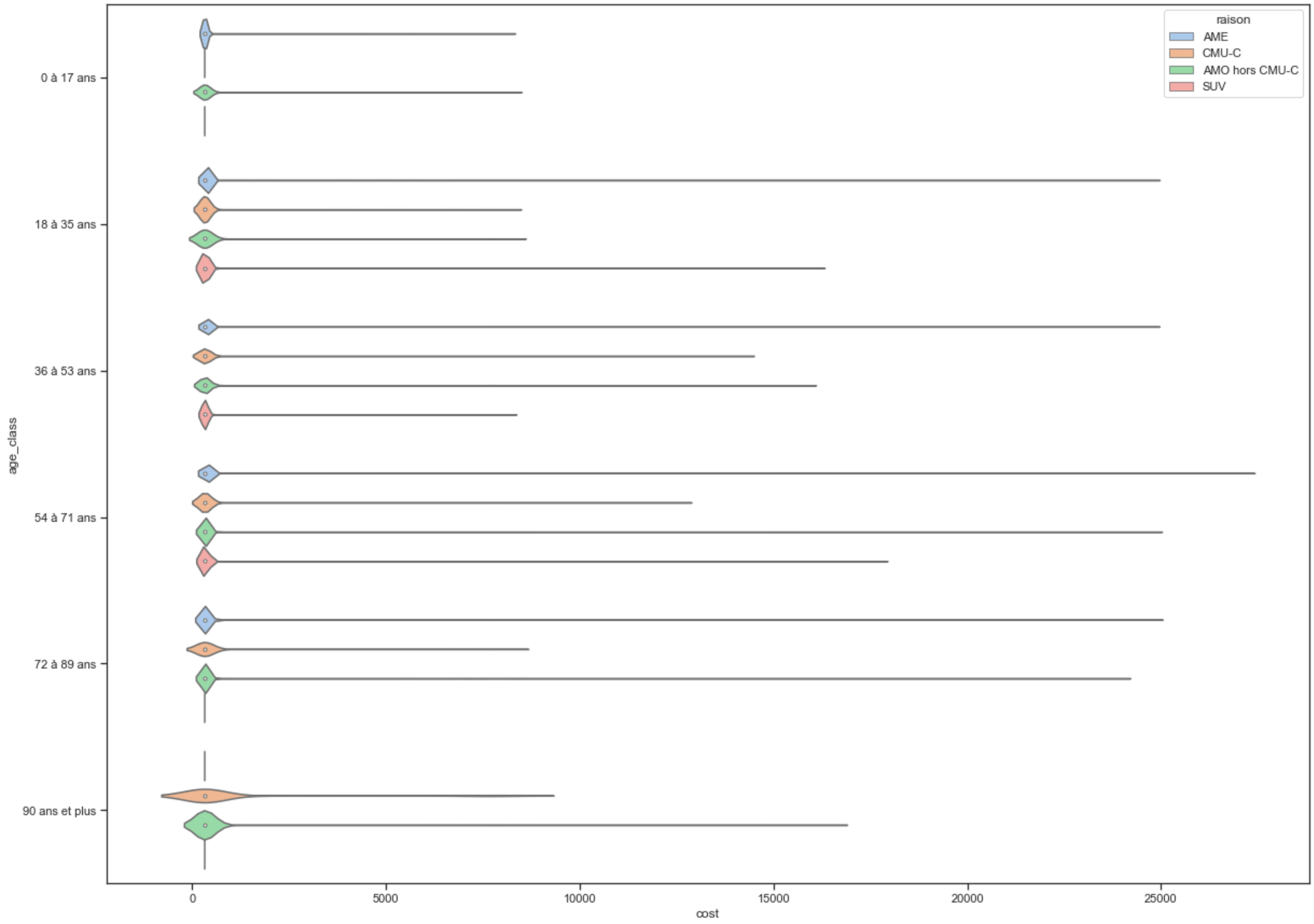
```
Out[72]: count    71516.000000
         mean       440.299180
         std        980.473443
         min        318.771289
         25%        318.771289
         50%        318.771289
         75%        318.771289
         max      24797.928181
         Name: cost, dtype: float64
```

```
In [73]: df_c[df_c['age_class']=='90 ans et plus']['cost'].describe()
```

```
Out[73]: count     3910.000000
         mean       532.746528
         std       1273.374199
         min        318.771289
         25%        318.771289
         50%        318.771289
         75%        318.771289
         max      16366.218585
         Name: cost, dtype: float64
```

```
In [74]: plt.figure(figsize=(20,15))
         sns.violinplot(data=df_c,x='cost',y='age_class',hue='raison')
```

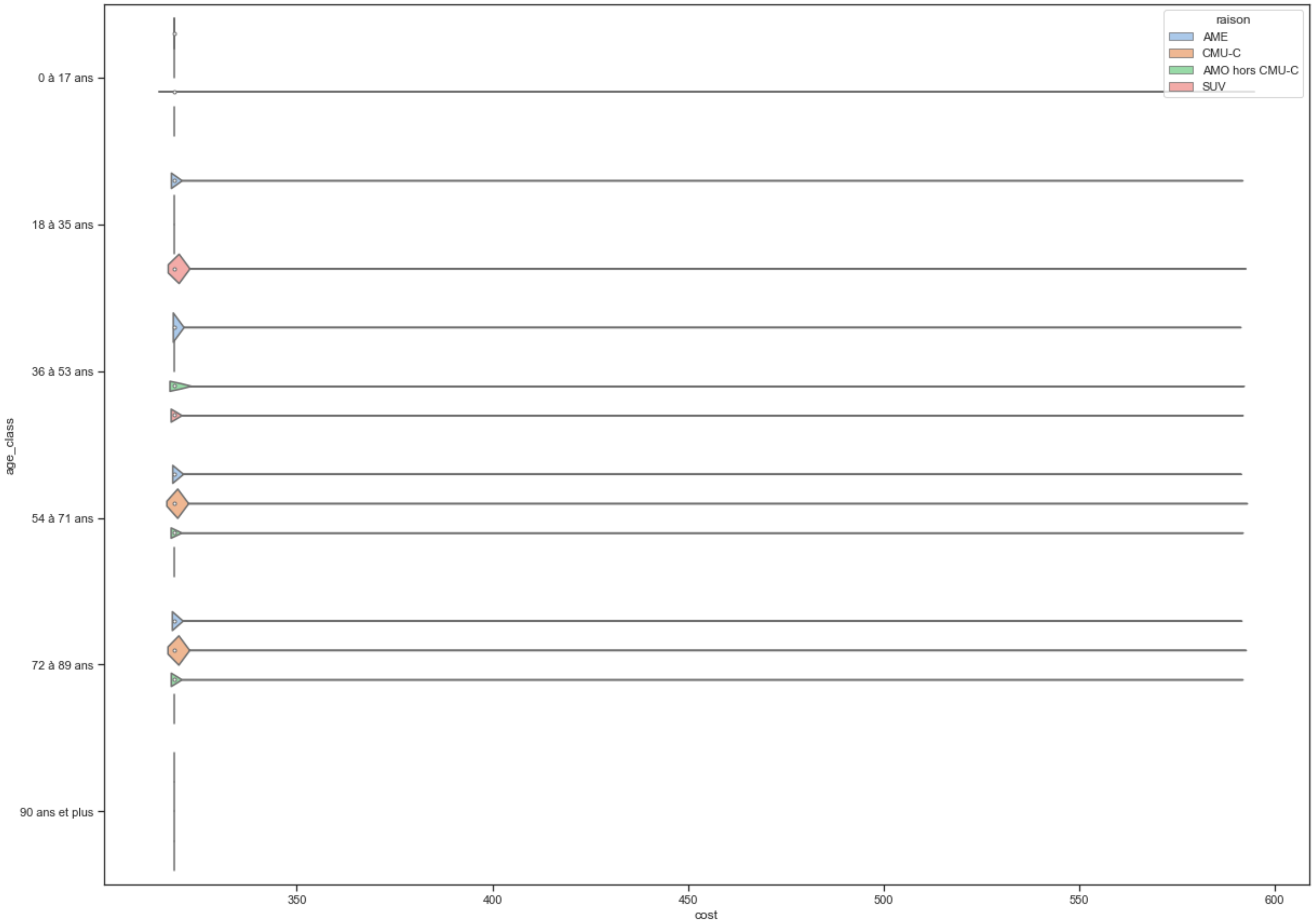
```
Out[74]: <AxesSubplot:xlabel='cost', ylabel='age_class'>
```



```
In [85]: df_c_trim = df_c[(df_c['cost']<(((600))))&(df_c['cost']>200)]
```

```
In [86]: plt.figure(figsize=(20,15))
         sns.violinplot(data=df_c_trim,x='cost',y='age_class',hue='raison')
```

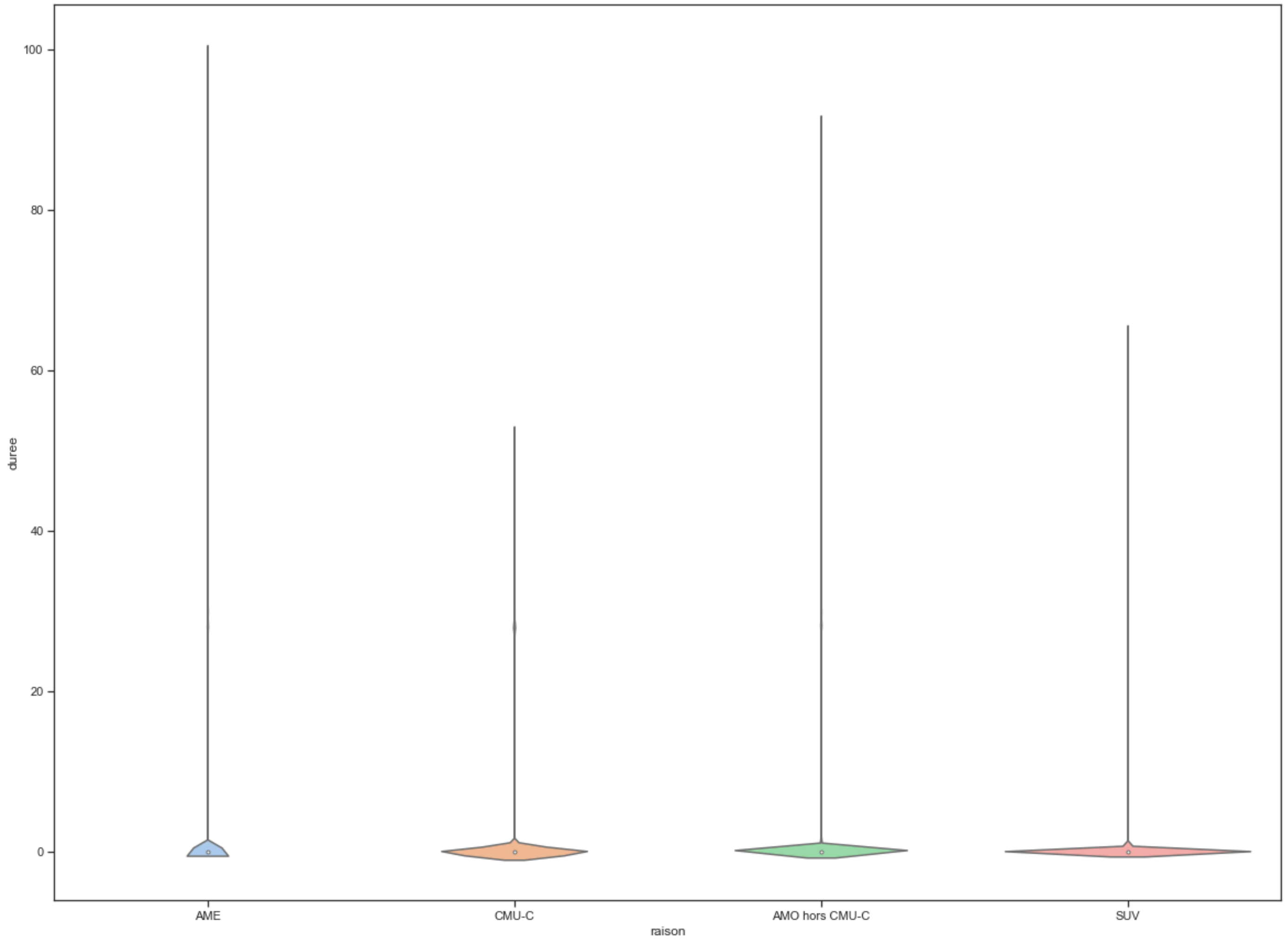
```
Out[86]: <AxesSubplot:xlabel='cost', ylabel='age_class'>
```



Durée

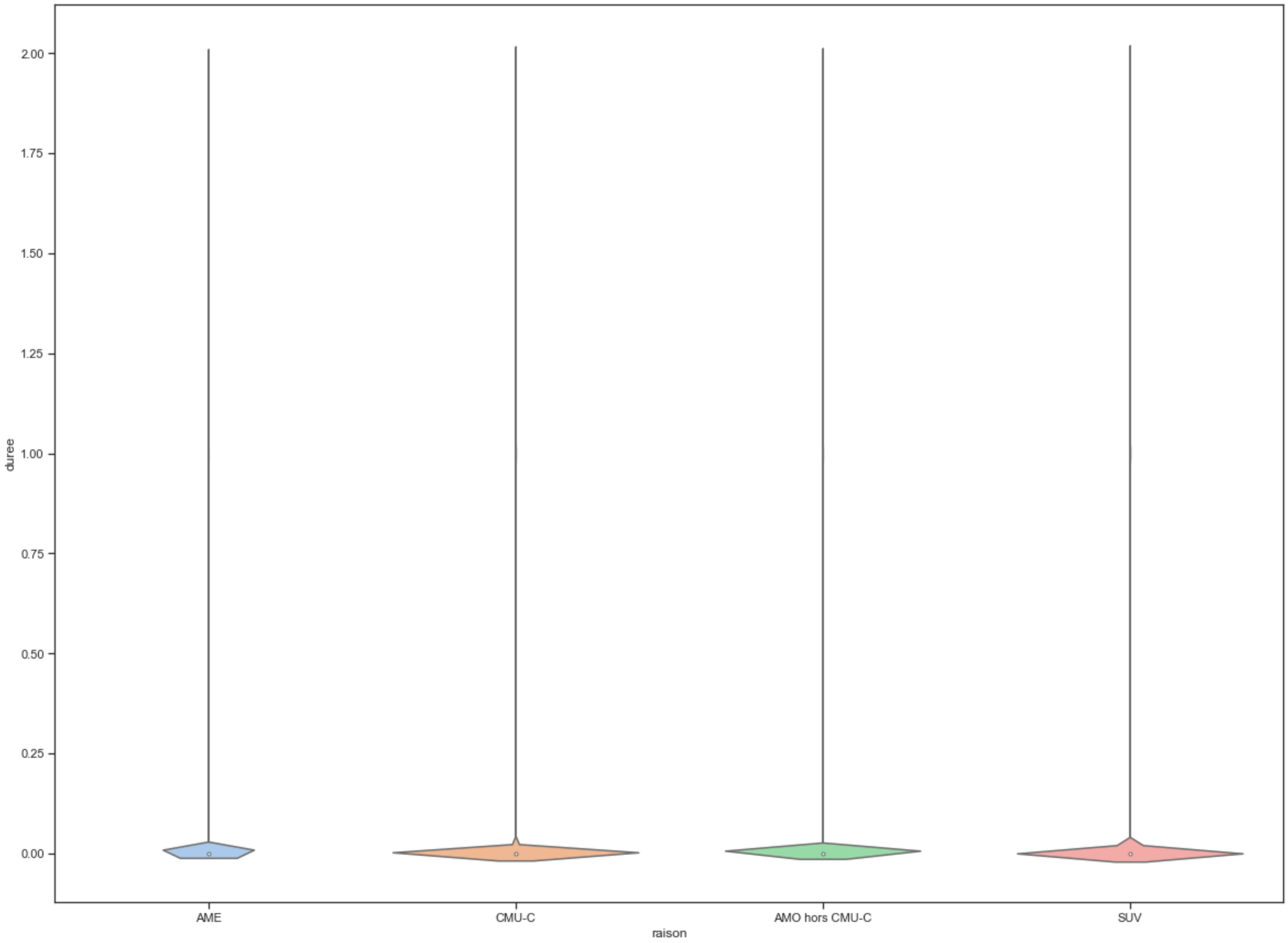
In [87]:
```
plt.figure(figsize=(20,15))
sns.violinplot(data=df_c,y='duree',x='raison')
```

Out[87]:
```
<AxesSubplot:xlabel='raison', ylabel='duree'>
```



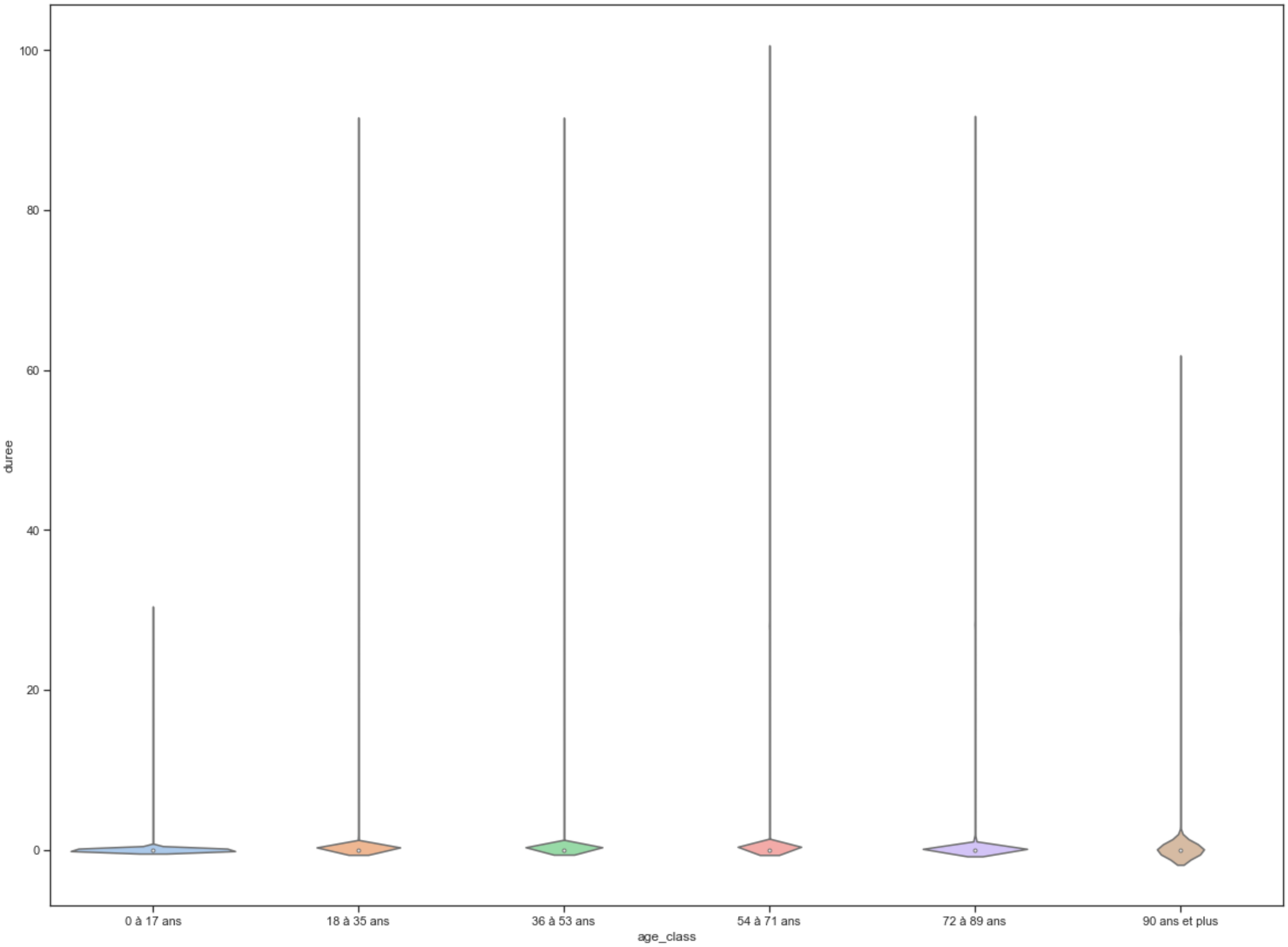
In [88]:
```
plt.figure(figsize=(20,15))
sns.violinplot(data=df_c_trim,y='duree',x='raison')
```

Out[88]:
```
<AxesSubplot:xlabel='raison', ylabel='duree'>
```



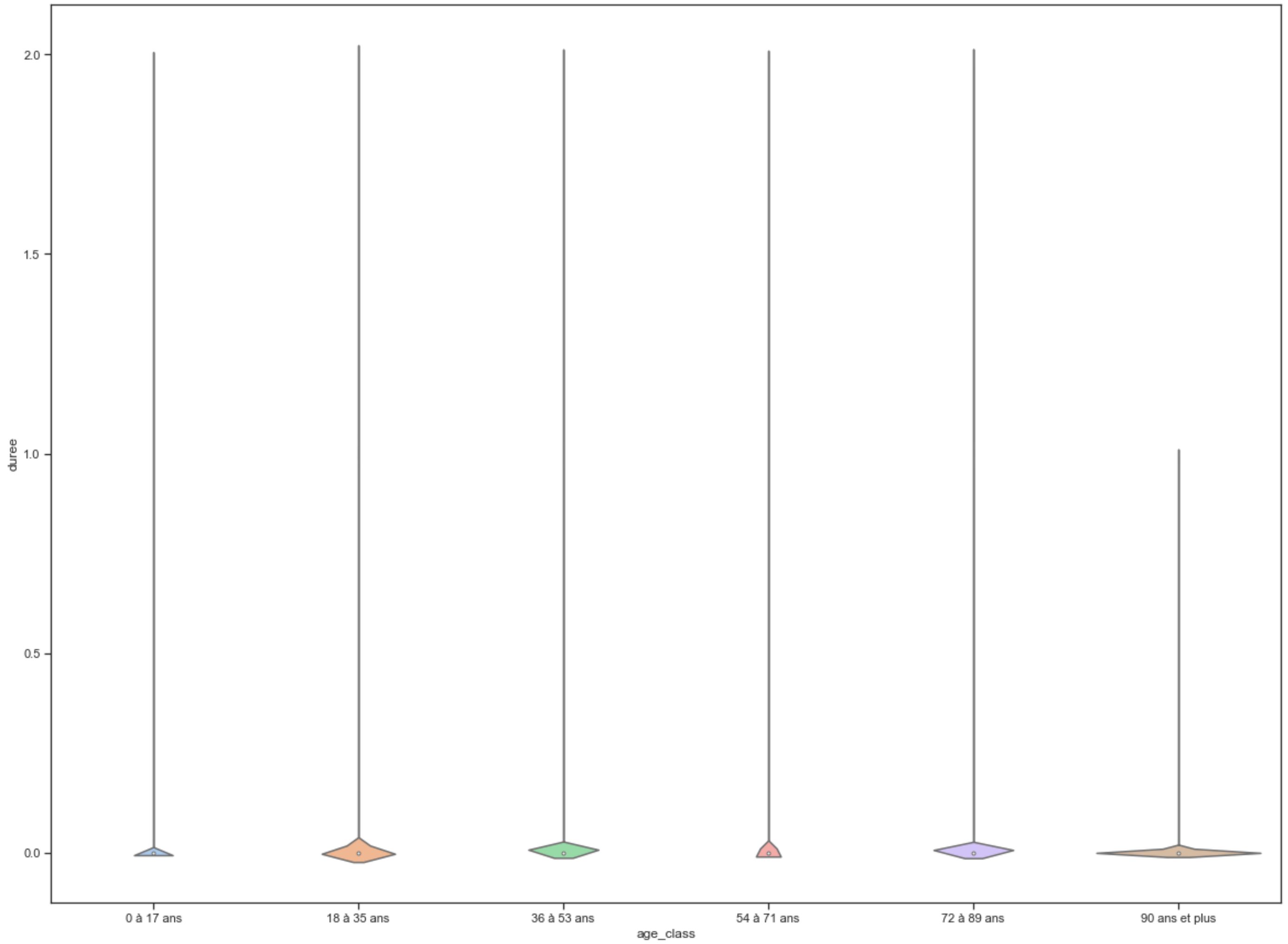
```
In [89]: plt.figure(figsize=(20,15))
         sns.violinplot(data=df_c,y='duree',x='age_class')
```

```
Out[89]: <AxesSubplot:xlabel='age_class', ylabel='duree'>
```



In [90]:
```
plt.figure(figsize=(20,15))
sns.violinplot(data=df_c_trim,y='duree',x='age_class')
```

Out[90]:
```
<AxesSubplot:xlabel='age_class', ylabel='duree'>
```

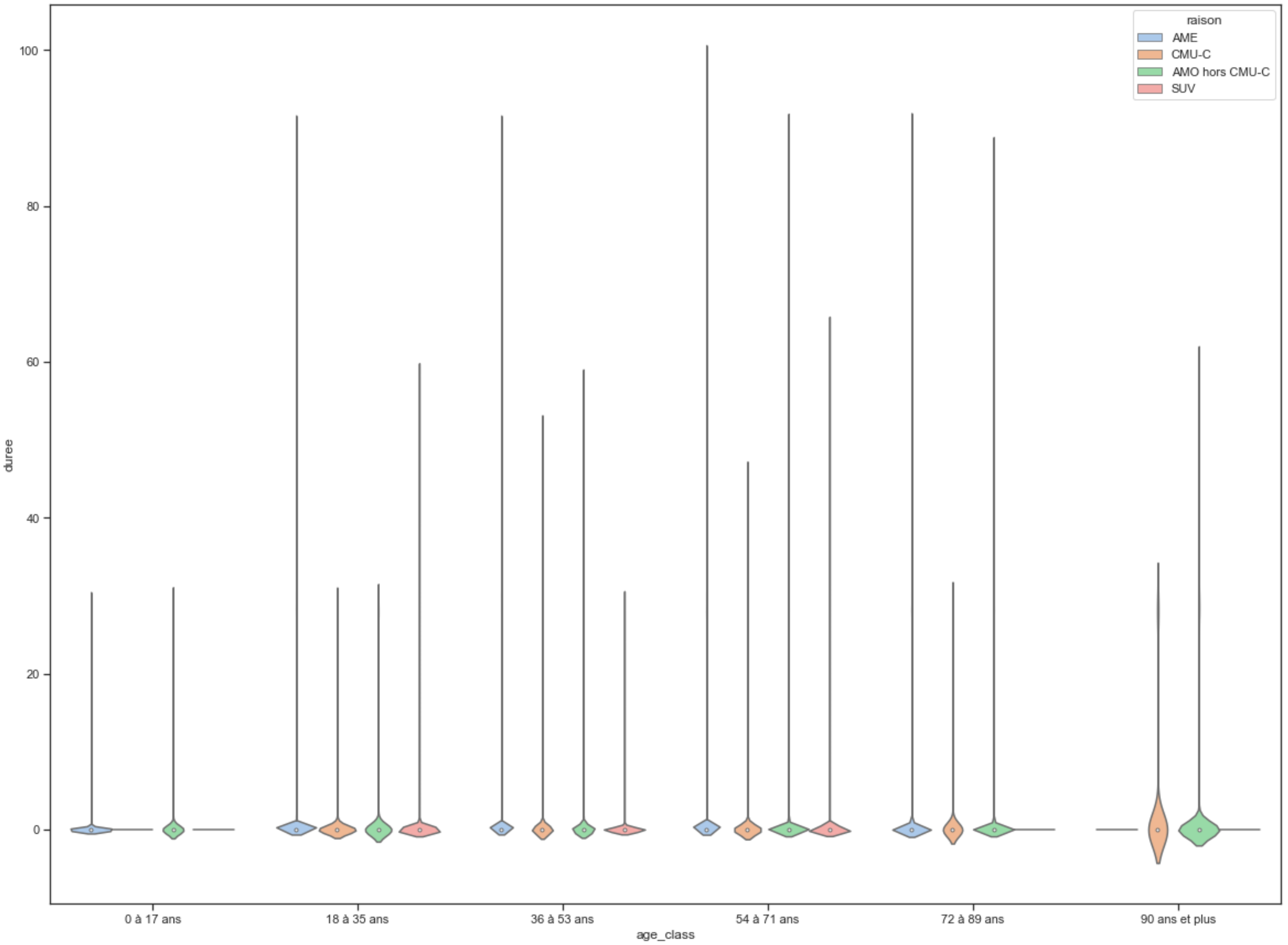


In [91]:

```
plt.figure(figsize=(20,15))
sns.violinplot(data=df_c,y='duree',x='age_class',hue='raison')
```

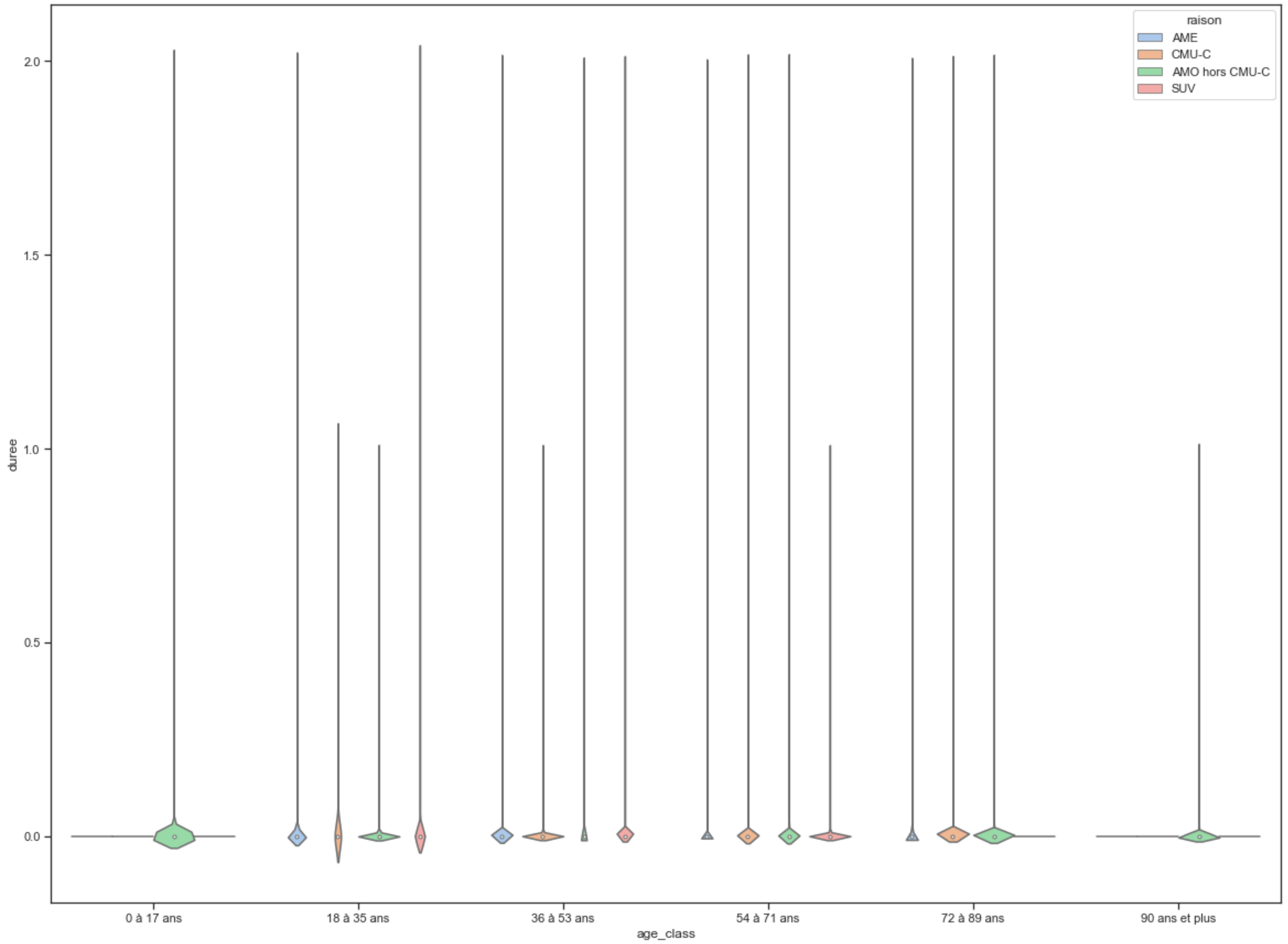
Out[91]:

```
<AxesSubplot:xlabel='age_class', ylabel='duree'>
```



```
In [92]: plt.figure(figsize=(20,15))
         sns.violinplot(data=df_c_trim,y='duree',x='age_class',hue='raison')

Out[92]: <AxesSubplot:xlabel='age_class', ylabel='duree'>
```



In [ ]:

In [93]:

```
df_c[df_c['raison']=='SUV']['duree'].describe()
```

Out[93]:

```
count    19204.000000
mean         0.173453
std          2.098090
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         65.000000
Name: duree, dtype: float64
```

In [94]:

```
df_c[df_c['raison']=='AMO hors CMU-C']['duree'].describe()
```

Out[94]:

```
count    106888.000000
mean          0.435465
std           3.625120
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max          91.000000
Name: duree, dtype: float64
```

In [95]:

```
df_c[df_c['raison']=='CMU-C']['duree'].describe()
```

Out[95]:

```
count    32501.000000
mean         0.592936
std          3.959194
min          0.000000
25%          0.000000
```

```
50%           0.000000
75%           0.000000
max          52.000000
Name: duree, dtype: float64
```

In [96]:
```
df_c[df_c['raison']=='AME']['duree'].describe()
```

Out[96]:
```
count    168608.000000
mean          0.255818
std           2.705883
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max         100.000000
Name: duree, dtype: float64
```

In [97]:
```
df_c[df_c['age_class']=='0 à 17 ans']['duree'].describe()
```

Out[97]:
```
count    10231.000000
mean         0.073209
std          1.431463
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         30.000000
Name: duree, dtype: float64
```

In [98]:
```
df_c[df_c['age_class']=='18 à 35 ans']['duree'].describe()
```

Out[98]:
```
count    45350.000000
mean         0.255502
std          2.607458
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         91.000000
Name: duree, dtype: float64
```

In [99]:
```
df_c[df_c['age_class']=='36 à 53 ans']['duree'].describe()
```

Out[99]:
```
count    86482.000000
mean         0.298779
std          2.858995
min          0.000000
25%          0.000000
50%          0.000000
75%          0.000000
max         91.000000
Name: duree, dtype: float64
```

In [100…
```
df_c[df_c['age_class']=='54 à 71 ans']['duree'].describe()
```

Out[100]:
```
count    109712.000000
mean          0.341795
std           3.185674
min           0.000000
25%           0.000000
50%           0.000000
75%           0.000000
max         100.000000
Name: duree, dtype: float64
```

In [101…
```
df_c[df_c['age_class']=='72 à 89 ans']['duree'].describe()
```

```
Out[101]: count    71516.000000
          mean         0.467224
          std          3.726931
          min          0.000000
          25%          0.000000
          50%          0.000000
          75%          0.000000
          max         91.000000
          Name: duree, dtype: float64
```

```
In [102…  df_c[df_c['age_class']=='90 ans et plus']['duree'].describe()
```

```
Out[102]: count    3910.000000
          mean        0.816624
          std         4.844988
          min         0.000000
          25%         0.000000
          50%         0.000000
          75%         0.000000
          max        60.000000
          Name: duree, dtype: float64
```

```
In [ ]:
```

```
In [ ]:
```