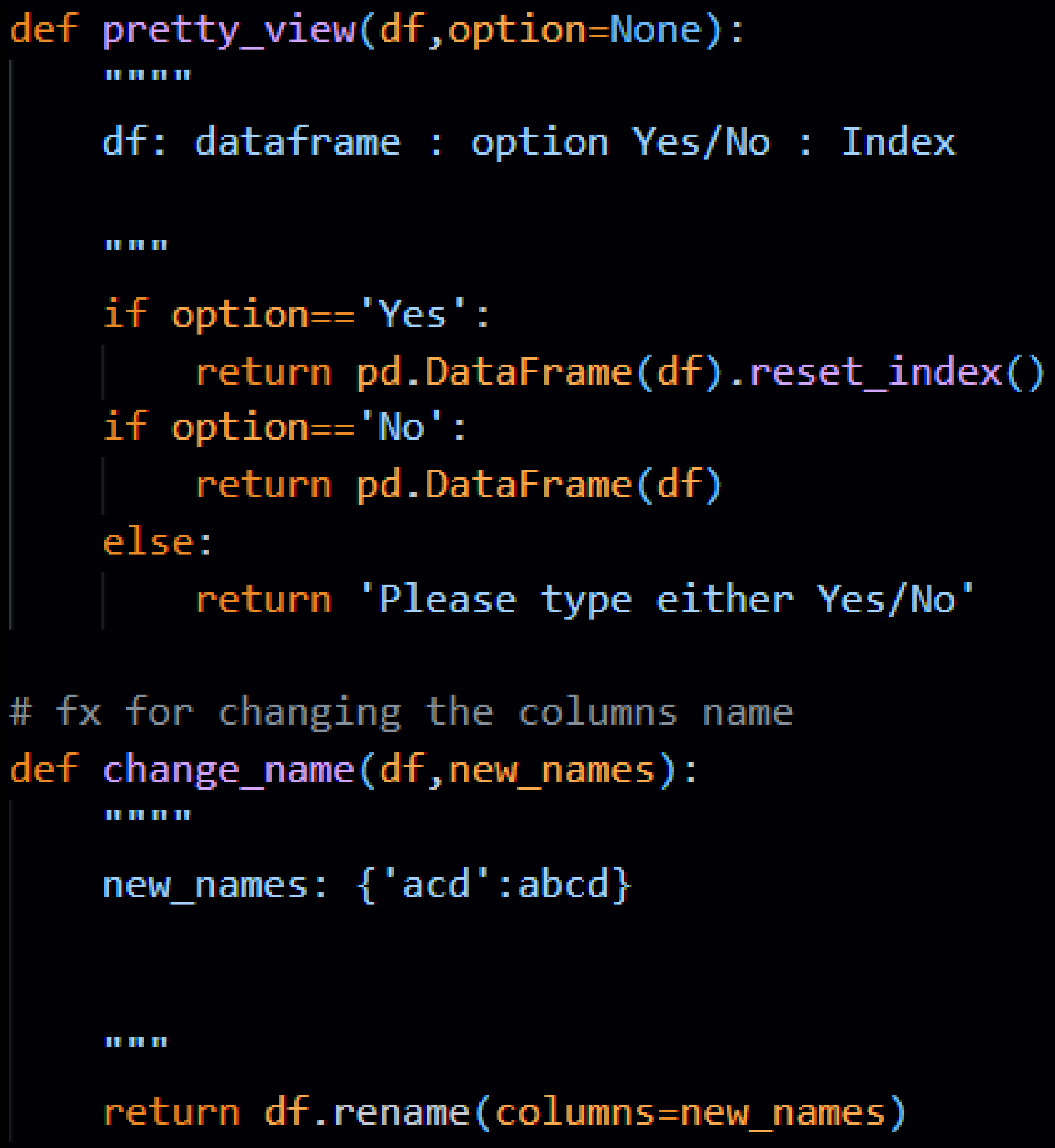
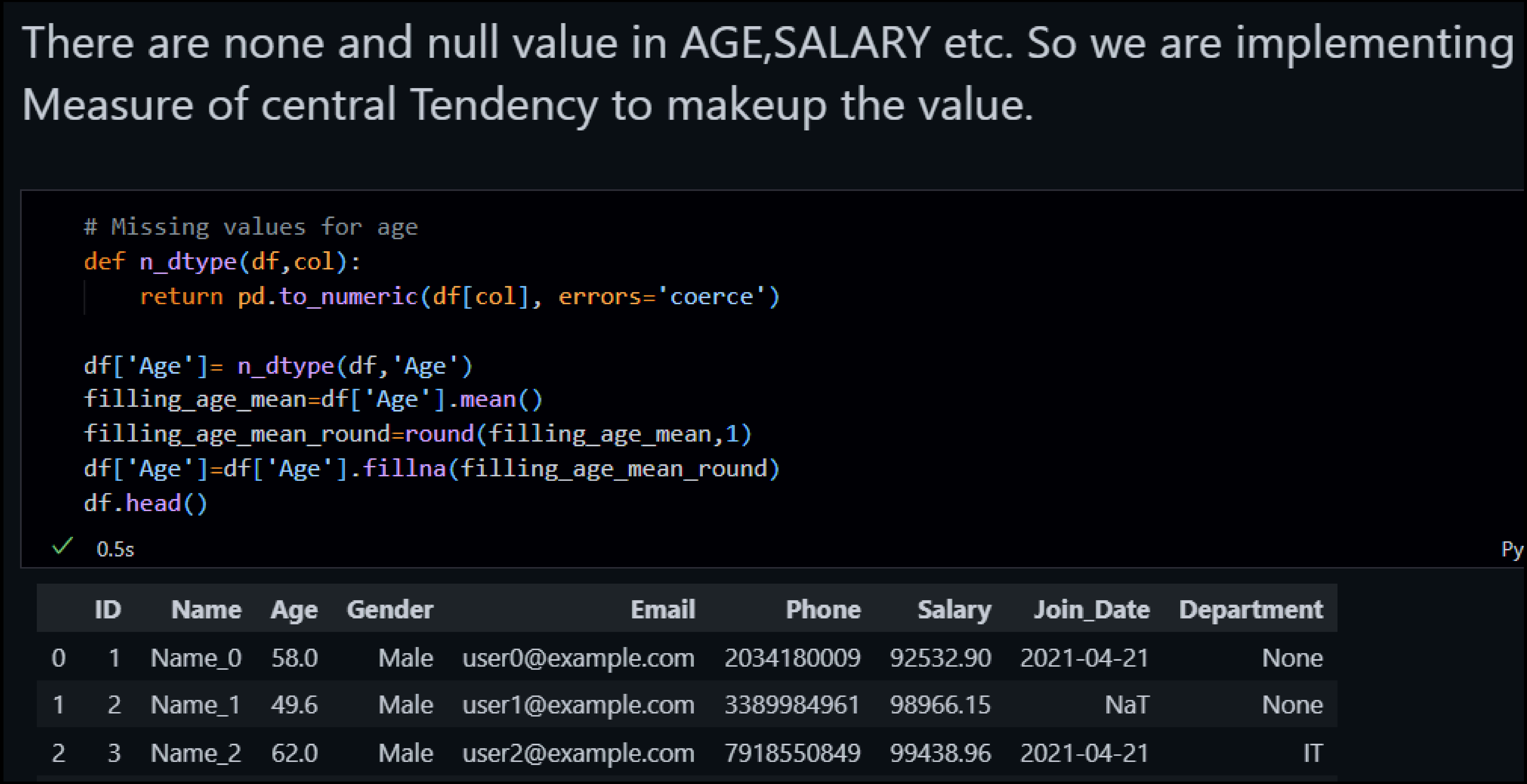
From Chaos to Clarity: A Data Cleaning Journey



```python
def pretty_view(df,option=None):
    """"
    df: dataframe : option Yes/No : Index

    """
    if option=='Yes':
        return pd.DataFrame(df).reset_index()
    if option=='No':
        return pd.DataFrame(df)
    else:
        return 'Please type either Yes/No'

# fx for changing the columns name
def change_name(df,new_names):
    """"
    new_names: {'acd':abcd}


    """
    return df.rename(columns=new_names)
```

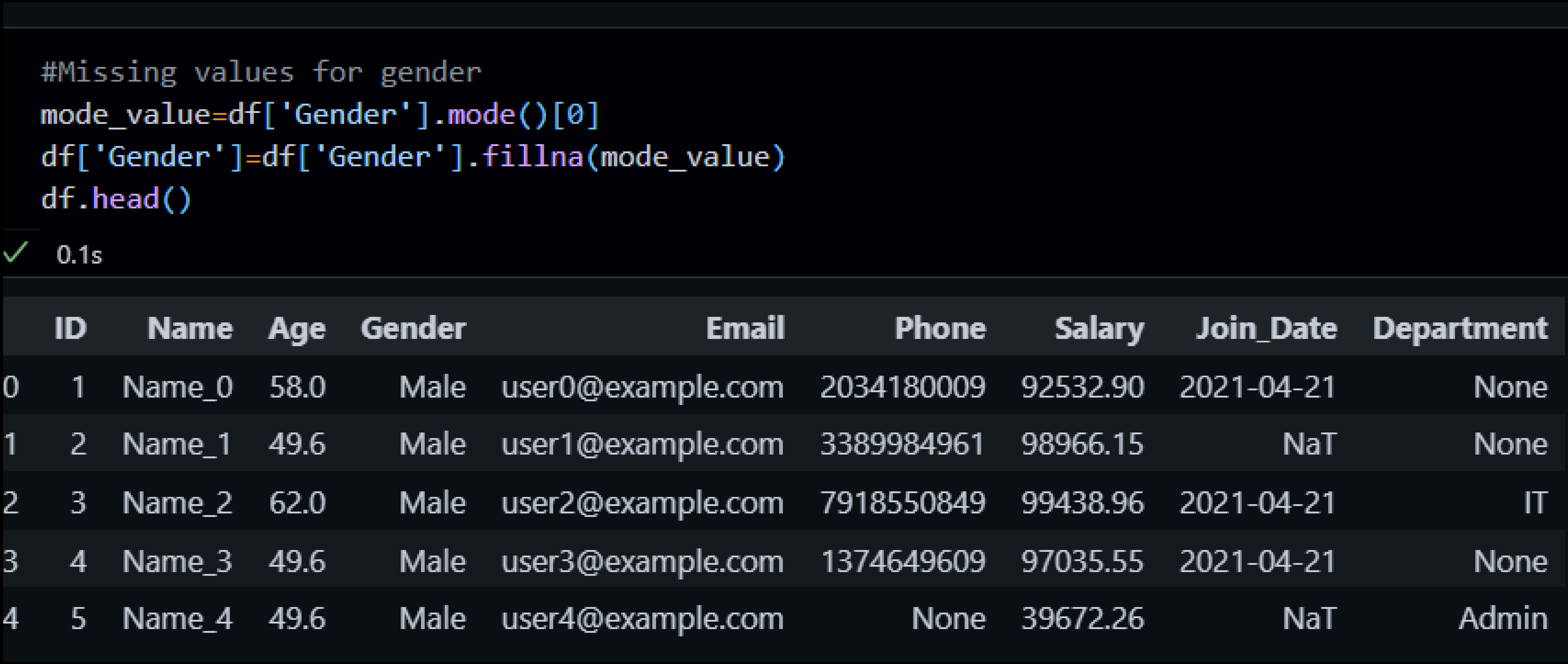


There are none and null value in AGE,SALARY etc. So we are implementing Measure of central Tendency to makeup the value.

```
# Missing values for age
def n_dtype(df,col):
    return pd.to_numeric(df[col], errors='coerce')

df['Age']= n_dtype(df,'Age')
filling_age_mean=df['Age'].mean()
filling_age_mean_round=round(filling_age_mean,1)
df['Age']=df['Age'].fillna(filling_age_mean_round)
df.head()
```

✓ 0.5s

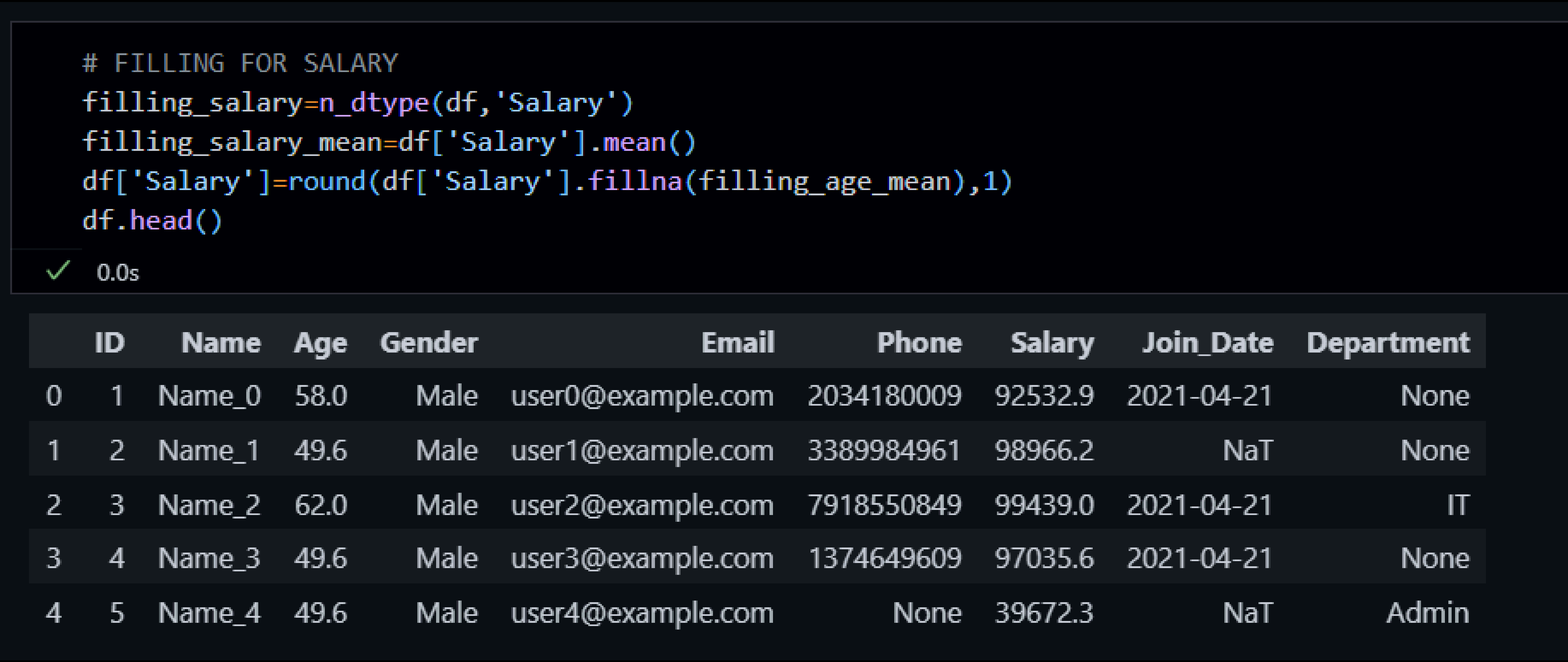
	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
0	1	Name_0	58.0	Male	user0@example.com	2034180009	92532.90	2021-04-21	None
1	2	Name_1	49.6	Male	user1@example.com	3389984961	98966.15	NaT	None
2	3	Name_2	62.0	Male	user2@example.com	7918550849	99438.96	2021-04-21	IT



```
#Missing values for gender
mode_value=df['Gender'].mode()[0]
df['Gender']=df['Gender'].fillna(mode_value)
df.head()
```

✓ 0.1s

	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
0	1	Name_0	58.0	Male	user0@example.com	2034180009	92532.90	2021-04-21	None
1	2	Name_1	49.6	Male	user1@example.com	3389984961	98966.15	NaT	None
2	3	Name_2	62.0	Male	user2@example.com	7918550849	99438.96	2021-04-21	IT
3	4	Name_3	49.6	Male	user3@example.com	1374649609	97035.55	2021-04-21	None
4	5	Name_4	49.6	Male	user4@example.com	None	39672.26	NaT	Admin

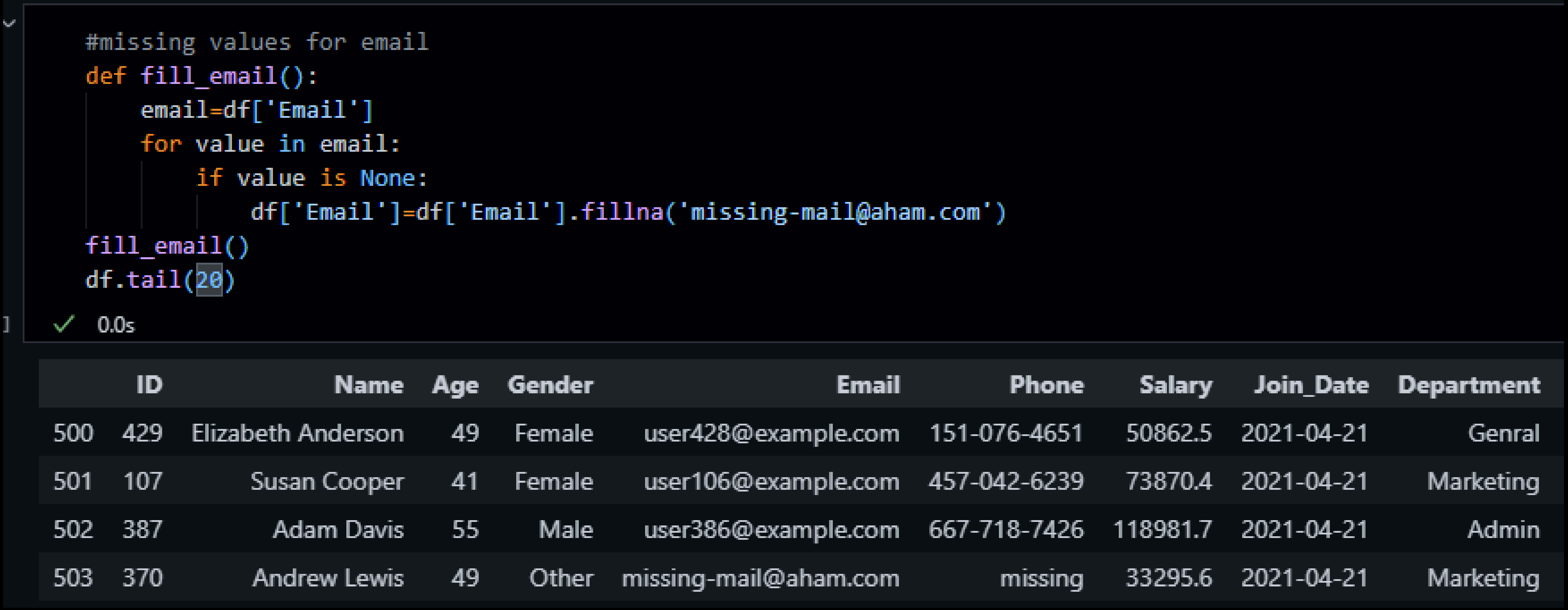


```python
# FILLING FOR SALARY
filling_salary=n_dtype(df,'Salary')
filling_salary_mean=df['Salary'].mean()
df['Salary']=round(df['Salary'].fillna(filling_age_mean),1)
df.head()
```

✓ 0.0s

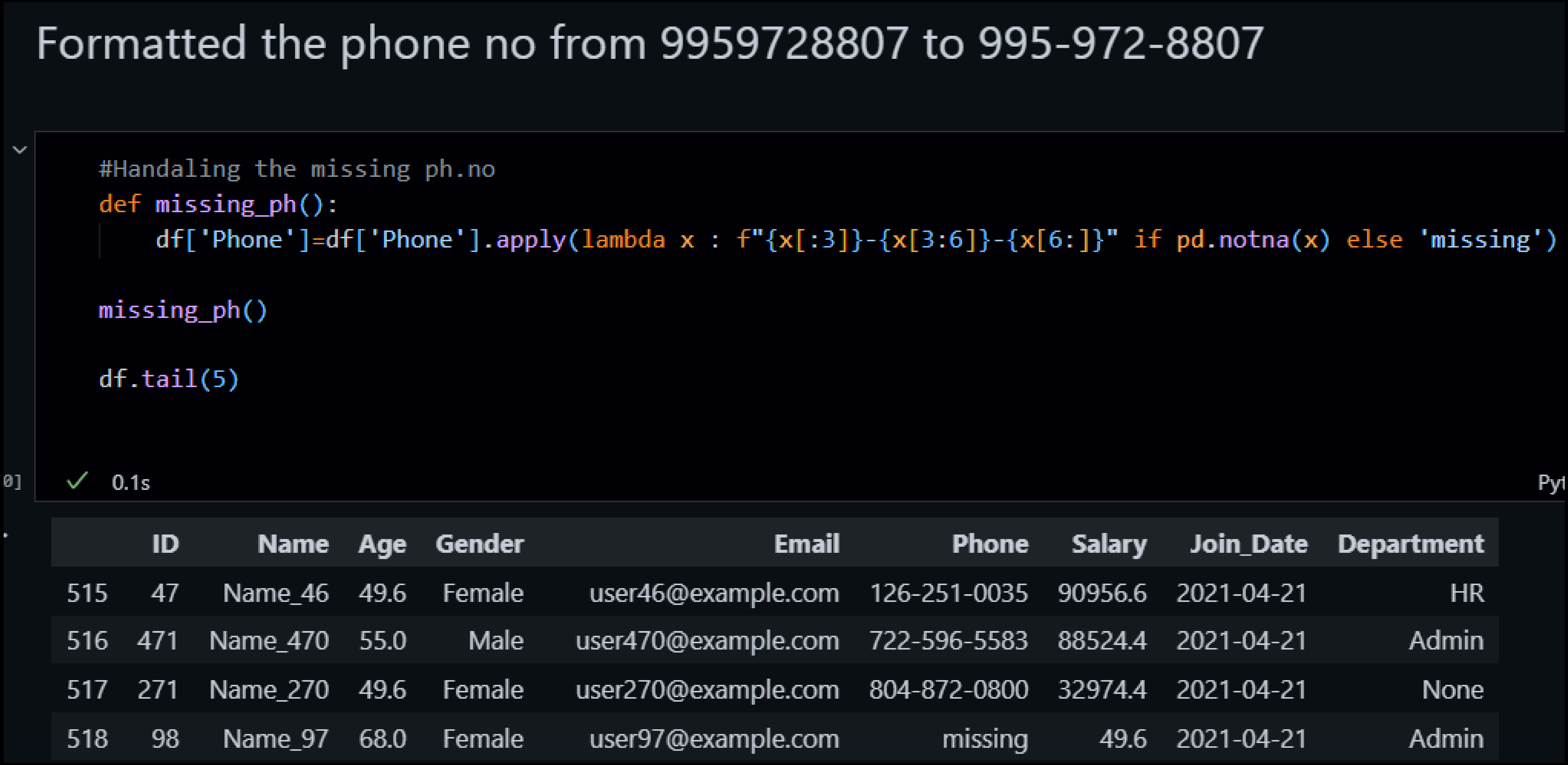
	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
0	1	Name_0	58.0	Male	user0@example.com	2034180009	92532.9	2021-04-21	None
1	2	Name_1	49.6	Male	user1@example.com	3389984961	98966.2	NaT	None
2	3	Name_2	62.0	Male	user2@example.com	7918550849	99439.0	2021-04-21	IT
3	4	Name_3	49.6	Male	user3@example.com	1374649609	97035.6	2021-04-21	None
4	5	Name_4	49.6	Male	user4@example.com	None	39672.3	NaT	Admin

Assigning 'missing-mail@aham.com' to the null entry



```python
#missing values for email
def fill_email():
    email=df['Email']
    for value in email:
        if value is None:
            df['Email']=df['Email'].fillna('missing-mail@aham.com')
fill_email()
df.tail(20)
```

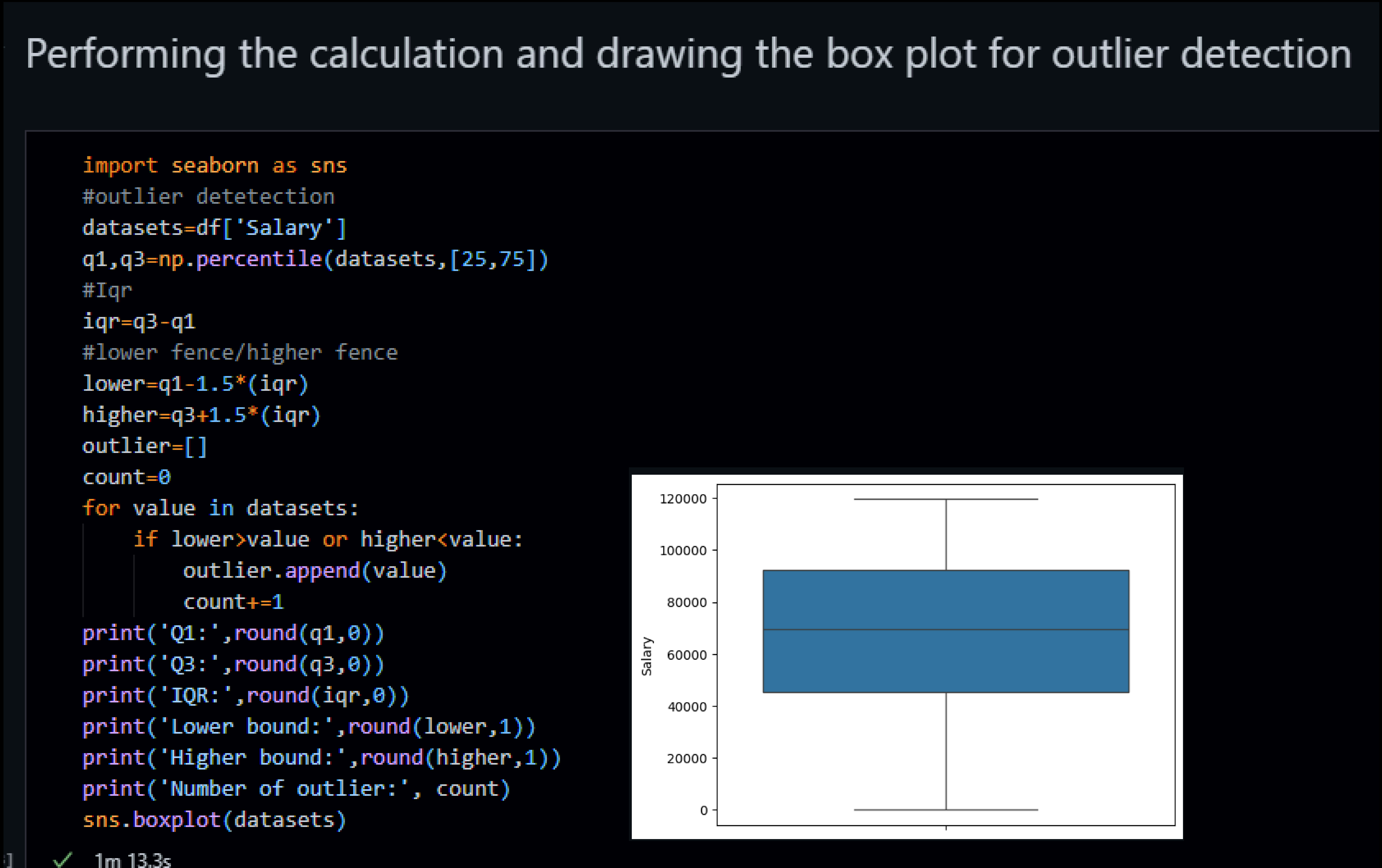
	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
500	429	Elizabeth Anderson	49	Female	user428@example.com	151-076-4651	50862.5	2021-04-21	Genral
501	107	Susan Cooper	41	Female	user106@example.com	457-042-6239	73870.4	2021-04-21	Marketing
502	387	Adam Davis	55	Male	user386@example.com	667-718-7426	118981.7	2021-04-21	Admin
503	370	Andrew Lewis	49	Other	missing-mail@aham.com	missing	33295.6	2021-04-21	Marketing



Formatted the phone no from 9959728807 to 995-972-8807

```python
#Handaling the missing ph.no
def missing_ph():
    df['Phone']=df['Phone'].apply(lambda x : f"{x[:3]}-{x[3:6]}-{x[6:]}" if pd.notna(x) else 'missing')

missing_ph()

df.tail(5)
```

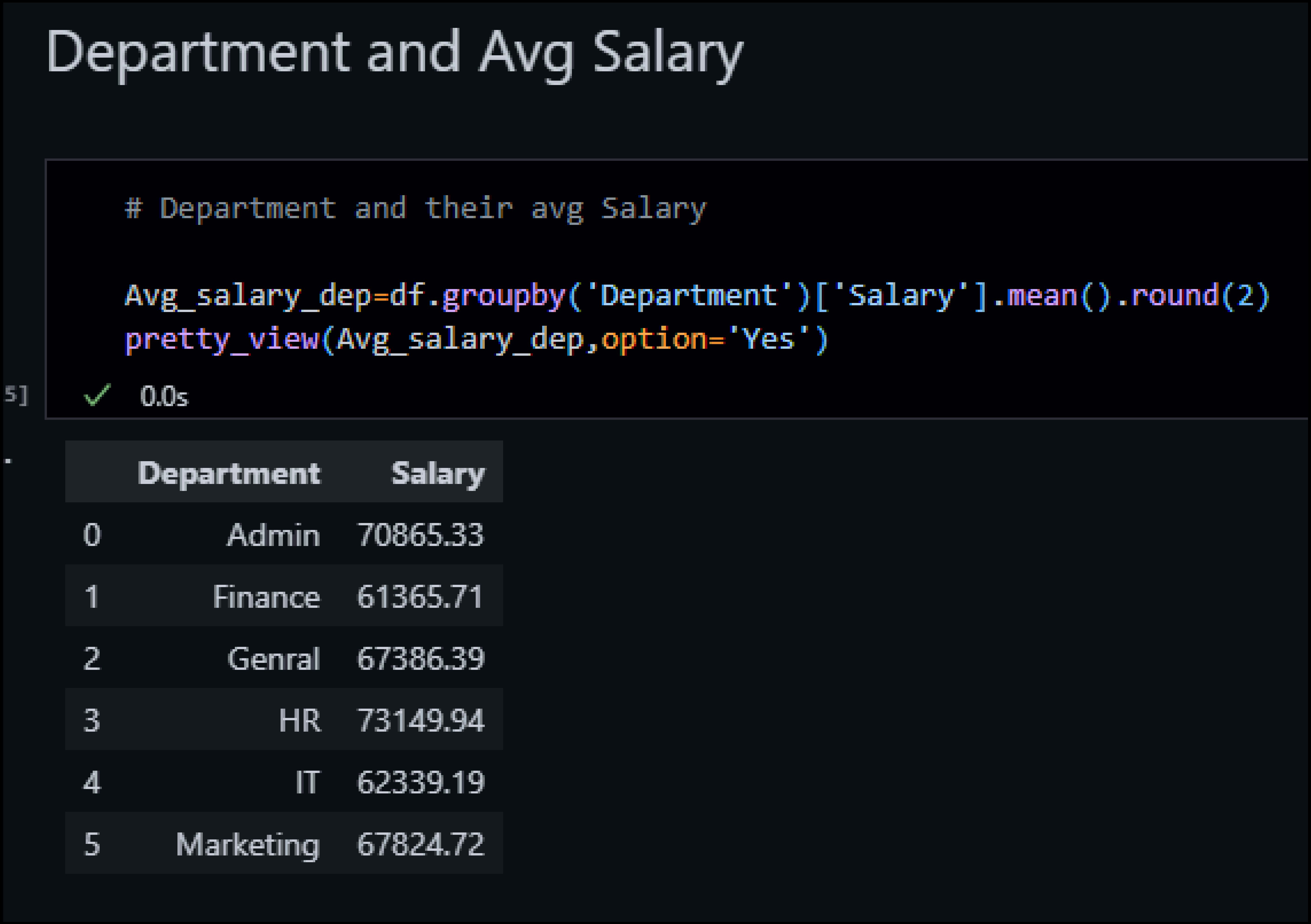
	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
515	47	Name_46	49.6	Female	user46@example.com	126-251-0035	90956.6	2021-04-21	HR
516	471	Name_470	55.0	Male	user470@example.com	722-596-5583	88524.4	2021-04-21	Admin
517	271	Name_270	49.6	Female	user270@example.com	804-872-0800	32974.4	2021-04-21	None
518	98	Name_97	68.0	Female	user97@example.com	missing	49.6	2021-04-21	Admin



Performing the calculation and drawing the box plot for outlier detection

```
import seaborn as sns
#outlier detetection
datasets=df['Salary']
q1,q3=np.percentile(datasets,[25,75])
#Iqr
iqr=q3-q1
#lower fence/higher fence
lower=q1-1.5*(iqr)
higher=q3+1.5*(iqr)
outlier=[]
count=0
for value in datasets:
    if lower>value or higher<value:
        outlier.append(value)
        count+=1
print('Q1:',round(q1,0))
print('Q3:',round(q3,0))
print('IQR:',round(iqr,0))
print('Lower bound:',round(lower,1))
print('Higher bound:',round(higher,1))
print('Number of outlier:', count)
sns.boxplot(datasets)
```

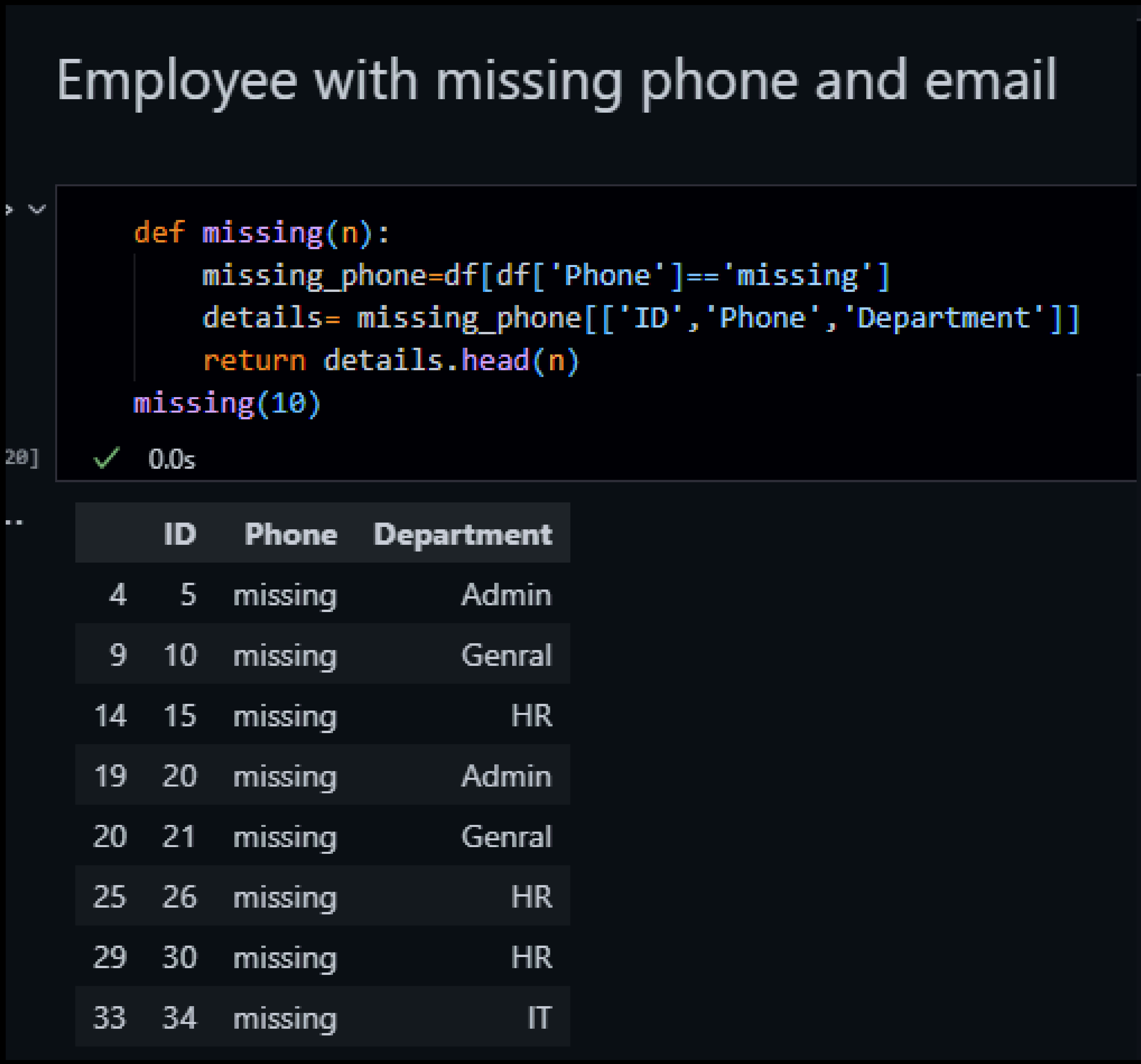
✓ 1m 13.3s



Department and Avg Salary

```python
# Department and their avg Salary

Avg_salary_dep=df.groupby('Department')['Salary'].mean().round(2)
pretty_view(Avg_salary_dep,option='Yes')
```

	Department	Salary
0	Admin	70865.33
1	Finance	61365.71
2	Genral	67386.39
3	HR	73149.94
4	IT	62339.19
5	Marketing	67824.72

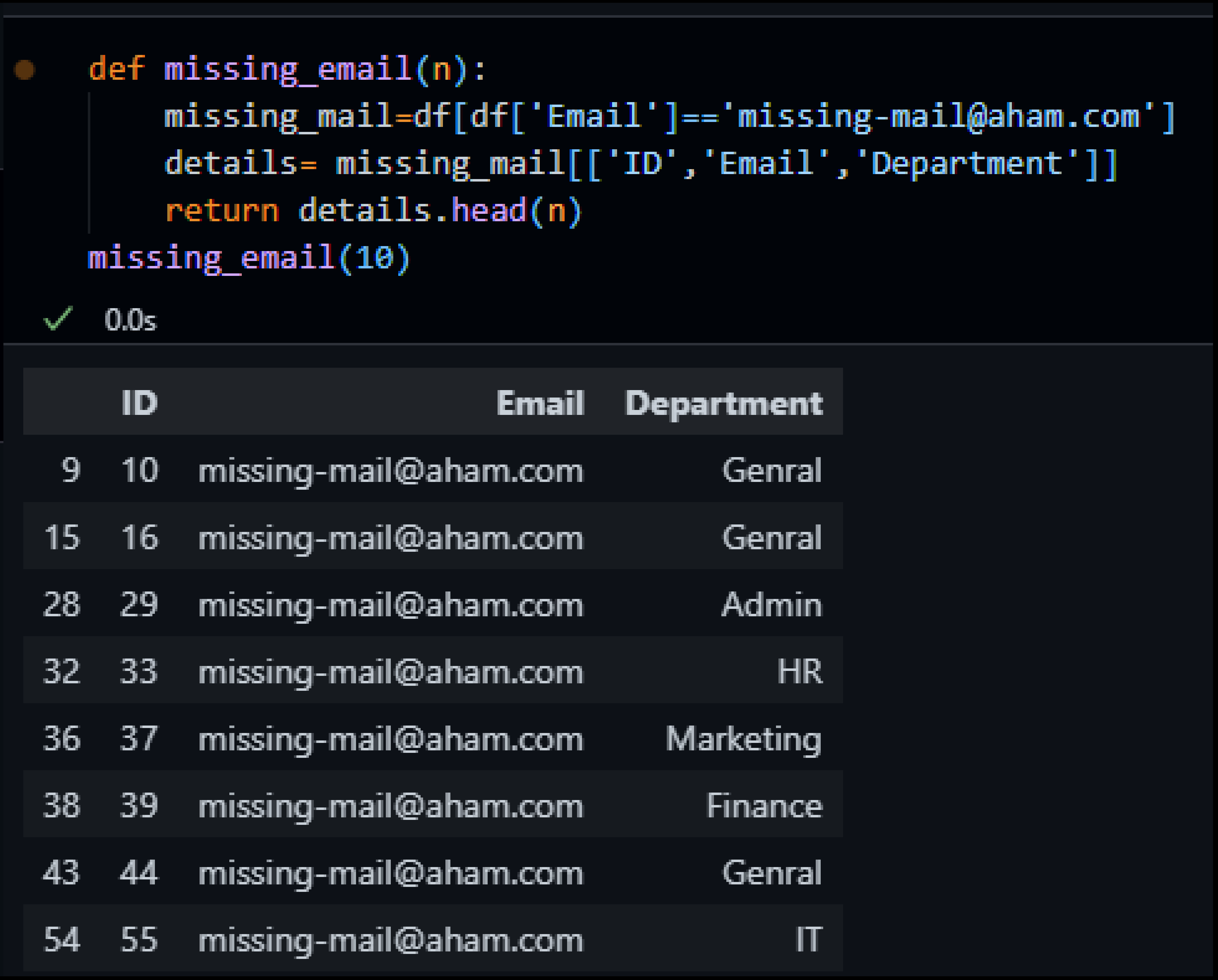


Employee with missing phone and email

```python
def missing(n):
    missing_phone=df[df['Phone']=='missing']
    details= missing_phone[['ID','Phone','Department']]
    return details.head(n)
missing(10)
```

✓ 0.0s

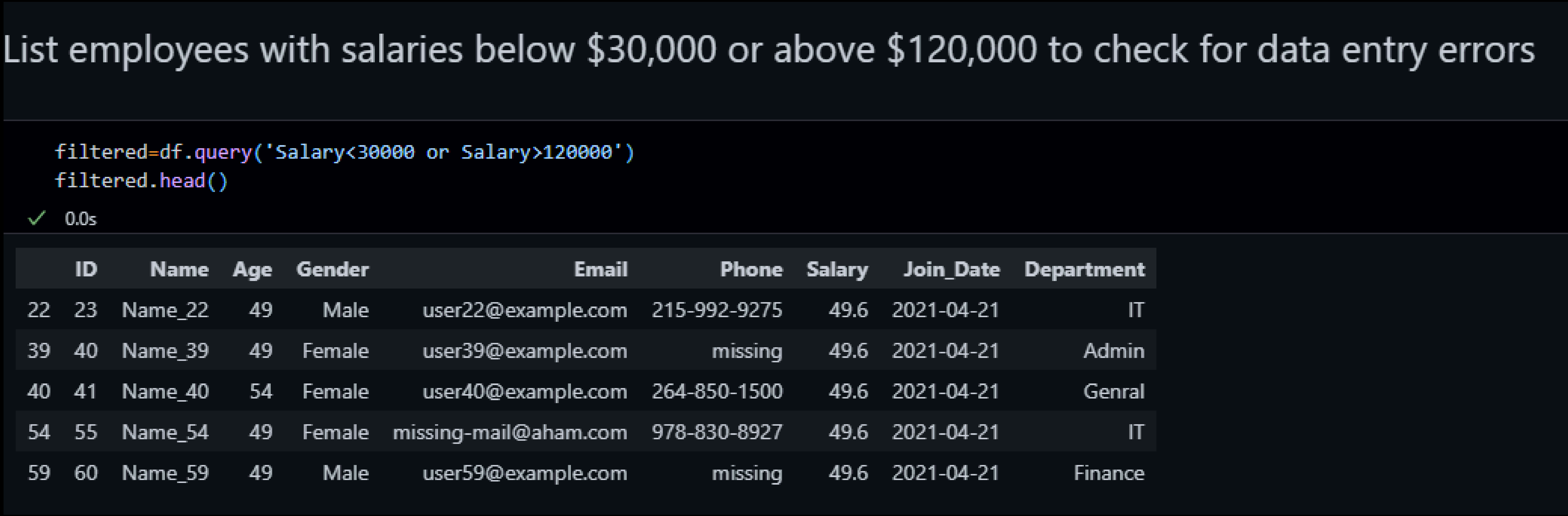
	ID	Phone	Department
4	5	missing	Admin
9	10	missing	Genral
14	15	missing	HR
19	20	missing	Admin
20	21	missing	Genral
25	26	missing	HR
29	30	missing	HR
33	34	missing	IT



```python
def missing_email(n):
    missing_mail=df[df['Email']=='missing-mail@aham.com']
    details= missing_mail[['ID','Email','Department']]
    return details.head(n)
missing_email(10)
```

✓ 0.0s

	ID	Email	Department
9	10	missing-mail@aham.com	Genral
15	16	missing-mail@aham.com	Genral
28	29	missing-mail@aham.com	Admin
32	33	missing-mail@aham.com	HR
36	37	missing-mail@aham.com	Marketing
38	39	missing-mail@aham.com	Finance
43	44	missing-mail@aham.com	Genral
54	55	missing-mail@aham.com	IT

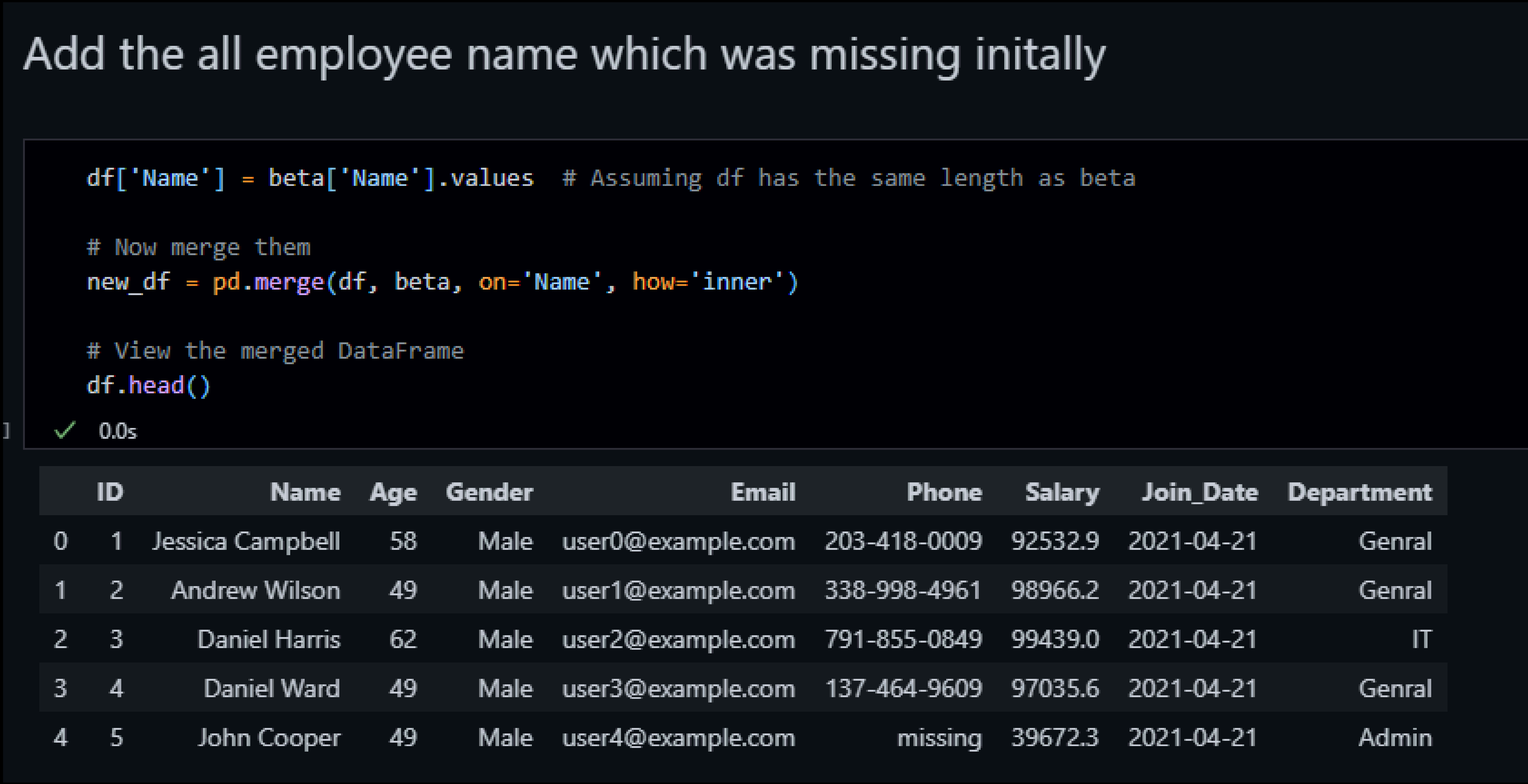


List employees with salaries below $30,000 or above $120,000 to check for data entry errors

```python
filtered=df.query('Salary<30000 or Salary>120000')
filtered.head()
```

✓ 0.0s

	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
22	23	Name_22	49	Male	user22@example.com	215-992-9275	49.6	2021-04-21	IT
39	40	Name_39	49	Female	user39@example.com	missing	49.6	2021-04-21	Admin
40	41	Name_40	54	Female	user40@example.com	264-850-1500	49.6	2021-04-21	Genral
54	55	Name_54	49	Female	missing-mail@aham.com	978-830-8927	49.6	2021-04-21	IT
59	60	Name_59	49	Male	user59@example.com	missing	49.6	2021-04-21	Finance

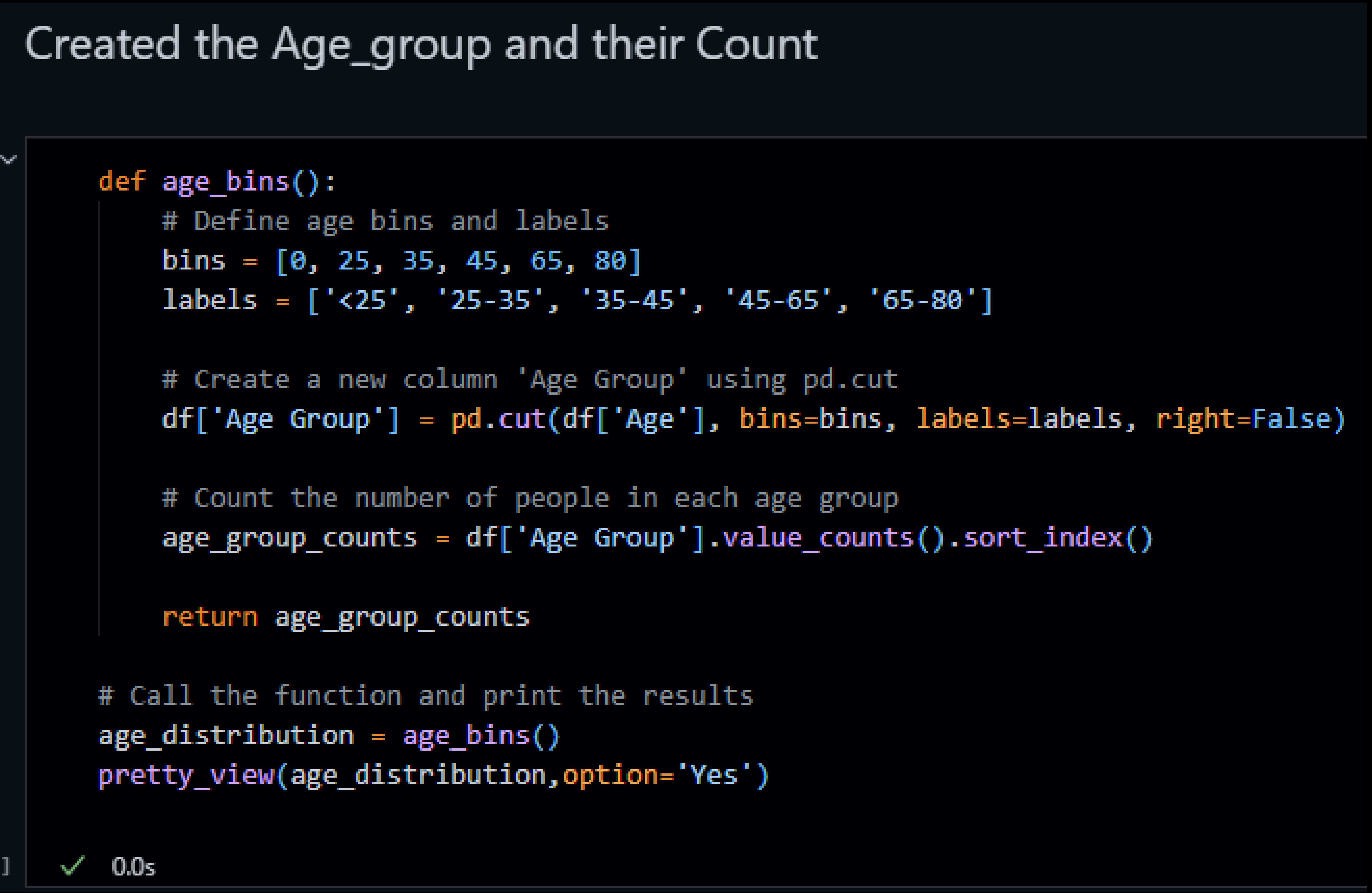


Add the all employee name which was missing initally

```
df['Name'] = beta['Name'].values  # Assuming df has the same length as beta

# Now merge them
new_df = pd.merge(df, beta, on='Name', how='inner')

# View the merged DataFrame
df.head()
```

✓ 0.0s

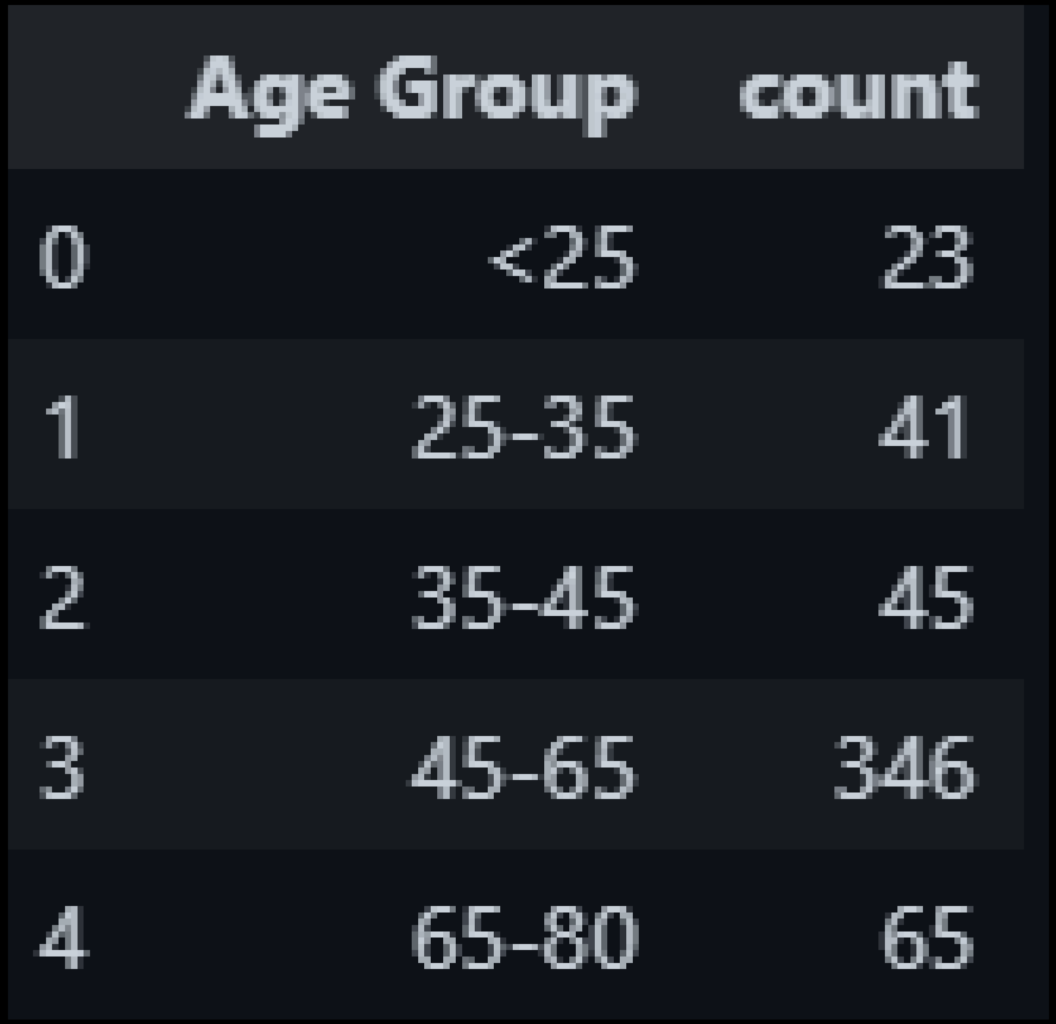
	ID	Name	Age	Gender	Email	Phone	Salary	Join_Date	Department
0	1	Jessica Campbell	58	Male	user0@example.com	203-418-0009	92532.9	2021-04-21	Genral
1	2	Andrew Wilson	49	Male	user1@example.com	338-998-4961	98966.2	2021-04-21	Genral
2	3	Daniel Harris	62	Male	user2@example.com	791-855-0849	99439.0	2021-04-21	IT
3	4	Daniel Ward	49	Male	user3@example.com	137-464-9609	97035.6	2021-04-21	Genral
4	5	John Cooper	49	Male	user4@example.com	missing	39672.3	2021-04-21	Admin



Created the Age_group and their Count

```
def age_bins():
    # Define age bins and labels
    bins = [0, 25, 35, 45, 65, 80]
    labels = ['<25', '25-35', '35-45', '45-65', '65-80']

    # Create a new column 'Age Group' using pd.cut
    df['Age Group'] = pd.cut(df['Age'], bins=bins, labels=labels, right=False)

    # Count the number of people in each age group
    age_group_counts = df['Age Group'].value_counts().sort_index()

    return age_group_counts

# Call the function and print the results
age_distribution = age_bins()
pretty_view(age_distribution,option='Yes')
```

✓ 0.0s



	Age Group	count
0	<25	23
1	25-35	41
2	35-45	45
3	45-65	346
4	65-80	65