Methodological guidelines.

Methods of data grouping

⚠️ **SUMMARY:**

The goal of the lesson is to formulate the students' understanding of data grouping as a complex filtering tool.

During the lesson, the students review tasks and filtering methods: they calculate basic statistics for individual columns of the original DataFrame or DataFrame filtered by a given condition, and they come to the conclusion that they need to learn more to work on more complex tasks. The children are introduced to functions for calculating the number of unique occurrences, value_counts(), and grouping, groupby() and pivot_table(), and they compare and evaluate how to use them to solve tasks. They are introduced to a new business term – the "target audience."

💾 **LINKS AND ACCESSORIES:**

- presentation for the lesson;
- tasks in VSC and on the platform: part 1, part 2.

🎯 EDUCATIONAL OUTCOMES OF THE LESSON

After the lesson, the students will:	*The results is achieved when the students:*
know: • that grouping can be used to find generalized values of characteristics for certain data categories; • the syntax for the value_counts(), groupby(), and pivot_table() methods; • what the term "target audience" means and its significance in terms of market research; understand: • what types of tasks can be solved using grouping; • that tasks performed using groupby() can be accomplished more elegantly using pivot_table(); know how to: • choose a data grouping method to find the answer to a question; • configure the parameters for grouping methods depending on the task; • get the values of specific indicators from DataFrame and Series obtained using grouping.	• answer the review questions correctly; • participate in the discussion of the new topic and ask clarifying questions; • participate in the discussion of the questions on the dataset; • access the DataFrame elements correctly; • complete the tasks on the platform; • try to correct their own mistakes.

RECOMMENDED LESSON STRUCTURE

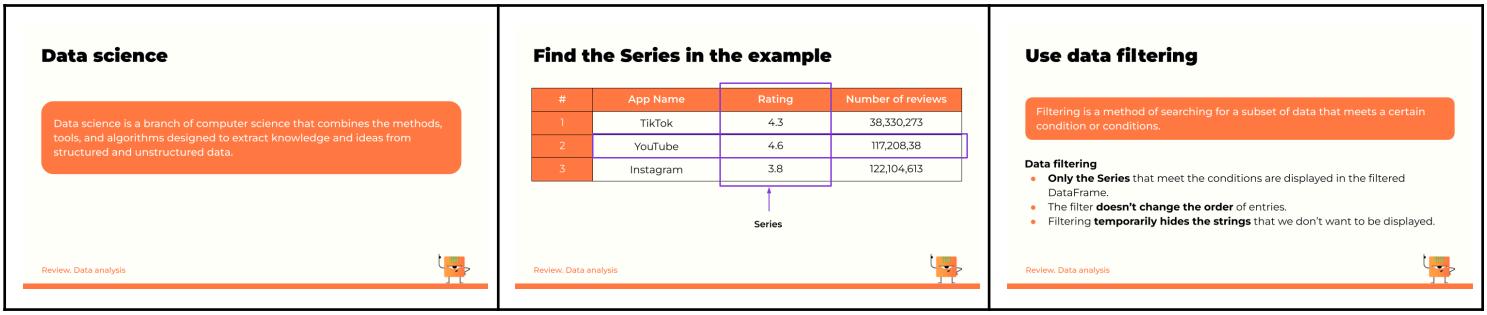
Time	Stage	Stage aims
5 min	**Review**	❏ Do a review of the material from the previous lesson: ❏ Pandas data structures: Series, DataFrame; ❏ filtering data by one or multiple conditions.
5 min	**Discussion: “Data grouping”**	❏ Formulate the problem: the filtering methods we've studied won't give us the answers to lots of important questions for our analysis. Arrive at the idea that we need to learn new data grouping tools.
20 min	**New topic: “Data grouping methods: simple grouping and calculating unique values”**	❏ Learn the syntax of the value_counts() method. ❏ Learn the syntax of the groupby() method. ❏ Learn how to apply multiple methods to grouped data. ❏ Learn how to access an element of the Series object obtained as a result of grouping.
15 min	**VSC + Platform. Data grouping**	❏ Have them complete the task “Grouping. Part 1” on the platform.
5 min	**Break**	❏ Help the students regain their concentration.
5 min	**Discussion: “Grouping data by multiple characteristics”**	❏ Go over the questions that require data from multiple columns. ❏ Arrive at the idea that they need to learn additional grouping methods.
15 min	**New topic: “Grouping by multiple characteristics. Pivot tables”**	❏ Go over the syntax of the groupby() method using lists as parameter values. ❏ Go over the syntax of the pivot_table() method.
15 min	**VSC + Platform. Data grouping**	❏ Have them complete the task “Grouping. Part 2” on the platform.
5 min	**Wrapping up the lesson. Reflection**	❏ Discuss the outcome of the lesson with the students. ❏ Answer any questions they have.

Review

(5 min)

Do a review of the previous lesson: Pandas data structures, Series and DataFrame, and filtering data by one or multiple conditions.

Review the material from the previous lesson. Make sure that all the students can navigate the DataFrame structure and understand the essence and purpose of data filtering. Pay special attention to their understanding of the core business questions and their impact on the success of the future product.



Discussion: “Data grouping”

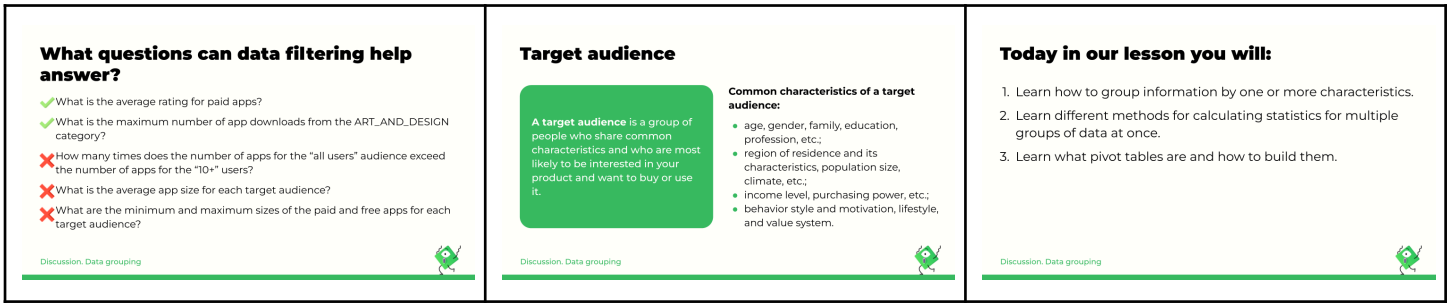
(5 min)

Formulate the problem: the filtering methods they've studied give us the answers to lots of important questions for our analysis. Arrive at the idea that they need to learn new data grouping tools.

Procedural comment. Some of the students may say that the last three tasks can be solved using filtering and basic algorithmic constructs. Explain to the students that this is true, but the code will have over a dozen strings to solve the last task. Moreover, there are standard methods in Pandas that will let us solve tasks of this type with a single string of code, and we will discuss those methods during the lesson.

Ask the students to think over the methods to solve the tasks on the slide. Discuss the context in which entrepreneurs need to know the answers to these questions. Explain what a target audience is and what its characteristics are. **Ask the students** why entrepreneurs study markets and target audiences. **Answer**: To make the highest possible profit. For that purpose, you need to understand who the product is for and who is most likely to use it.

Arrive at the conclusion that the first two tasks can easily be solved using filtering, but to solve the remaining three, we will need to learn some more powerful methods. Tell the children that the method that can help us get the data we need is called “grouping,” and that today’s lesson will be dedicated to different methods of grouping.

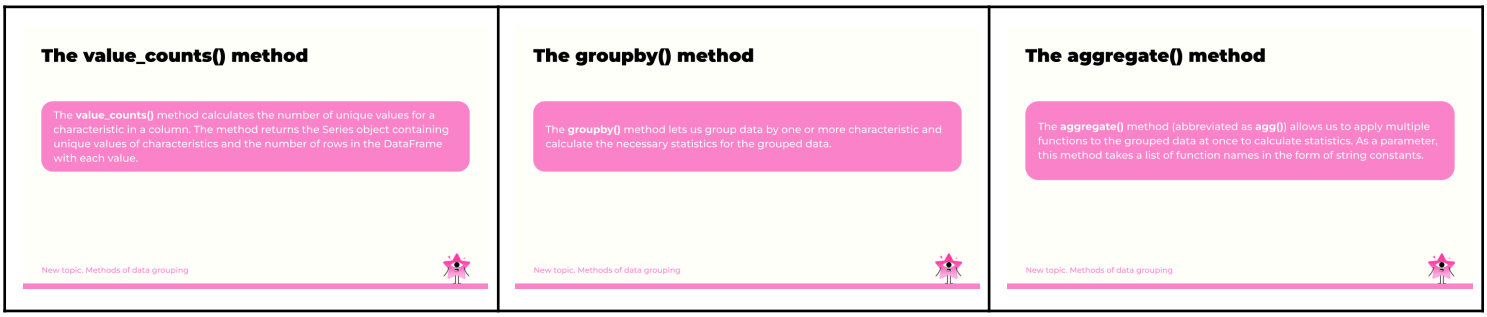


New topic: “Methods of grouping data: simple grouping and calculating unique values”

(20 min)

Give a definition of the concept of “grouping.” Discuss how grouping is different from filtering. Next, move on to discussing the tasks that were set during the previous stage of the lesson.

In the first half of the lesson, go over the solutions to the first two tasks. Go over the syntax of the value_counts() and groupby() methods. Pay special attention to the capability of applying multiple statistical functions to grouped data using the agg() method. Show the results of each method.



VSC + Platform. Data grouping

(15 min)

Have the students work with a dataset. Let the students know that they will need to write the code in VSC and then transfer the answers they get to the platform where the tasks with auto-check are.

If you see that the children are hesitant and having difficulty getting started, do a couple of tasks together with them. Encourage the students to be active in asking questions and comment on what you are doing.

You will find the answers to the tasks and the reference code to get them at the end of the methodological guidelines.

Break

(5 min)

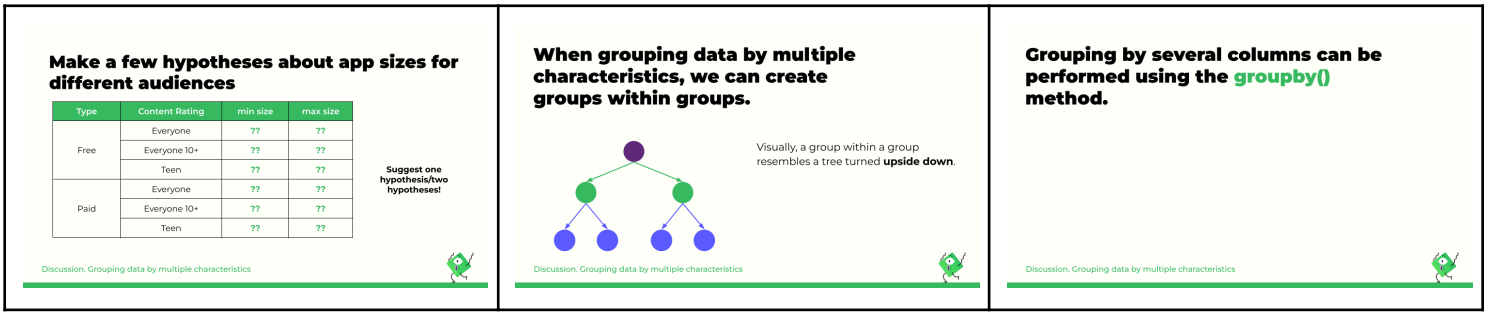
Get the students off their computers. The purpose of the break is to shift their attention and do an activity.

Discussion: “Grouping data by multiple characteristics”

(5 min)

Return to the discussion regarding the dataset. Before answering the last question, recall how an assumption differs from a fact. Ask the students to use the data they have to put forward their own hypotheses about the relationship between the target audience, the cost of the app, and how big it is. Remind the students that a refuted hypothesis also becomes a fact. **For example,** the assumption that all paid apps for each audience are smaller relative to free apps is not true, but the opposite is true.

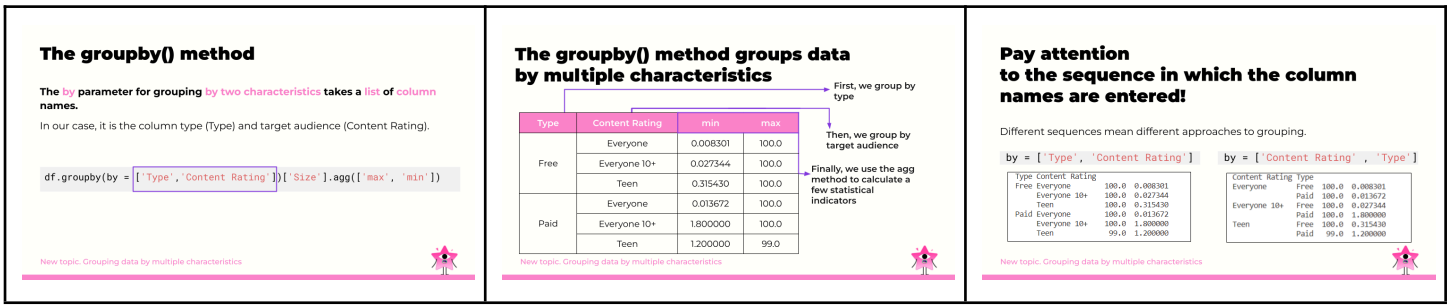
Discuss how to calculate the minimum and maximum size of the paid and free apps for each target audience. Use the table on the slide to make it more clear. Discuss whether grouping with the groupby() method can help with this task. Come to the conclusion that the groupby() method can help in solving this task if we apply grouping by multiple characteristics at the same time.



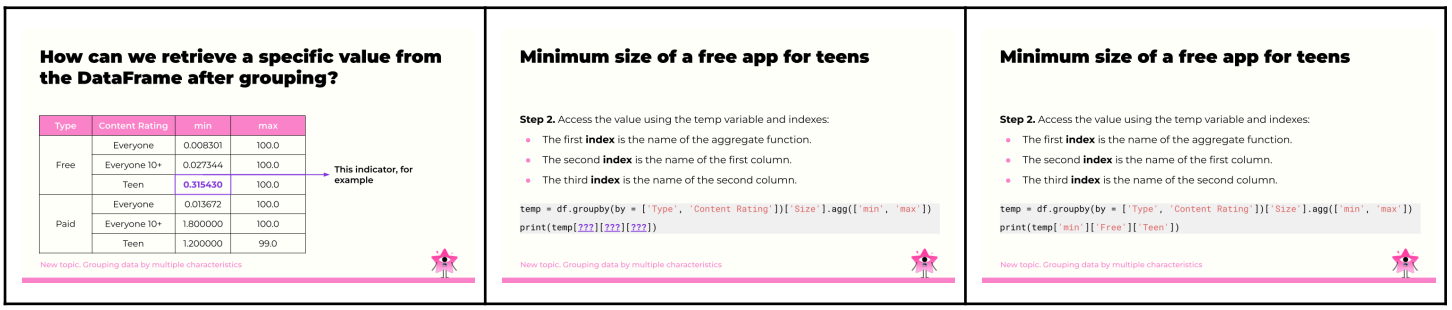
New topic: “Grouping by multiple characteristics. Pivot tables”

(15 min)

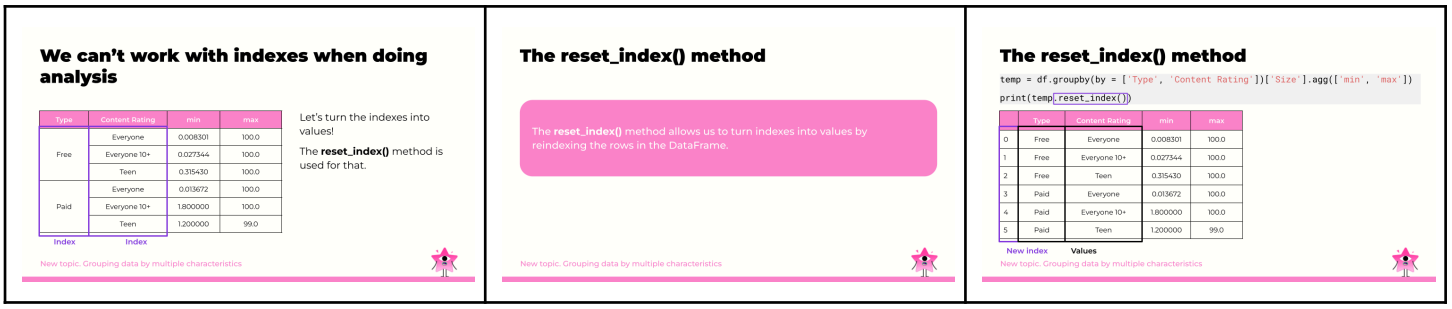
Solve the task from the previous stage using the groupby() method with two characteristics. Discuss that in this case the column names should be sent as a list, and the order in which the elements are sent within the list is important.



Discuss how to retrieve a specific value from DataFrame after grouping.

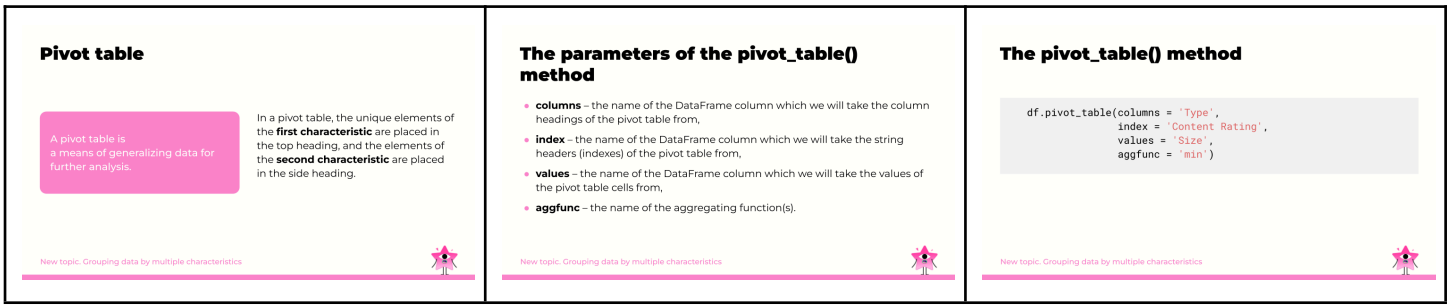


Discuss how a set of indexes can be converted to a value column using the reset_index() method. New indexes numbered starting from zero will appear in DataFrame. We will need this method in the future when building graphs.



Discuss how easy it is to navigate in the table after grouping by multiple columns. Draw the students' attention to the fact that data structures with hierarchical indexes are difficult to understand. In contrast to those structures, show the students a pivot table where the same information is placed more compactly and it is easier to navigate.

Go over the syntax of the pivot_table() method. Emphasize to the students that each parameter can take as a value either a string constant (or a number if the column headings are numeric) or a list of values. Use the third task from the set of tasks discussed at the beginning of the lesson as an example for going over the method.



VSC + Platform. Data grouping

(15 min)

Have the students work with a dataset. Let the students know that they will need to write the code in VSC and then transfer the answers they get to the platform where the tasks with auto-check are.

If you see that the children are hesitant and having difficulty getting started, do a couple of tasks together with them. Encourage the students to be active in asking questions and comment on what you are doing.

You will find the answers to the tasks and the reference code to get them at the end of the methodological guidelines.

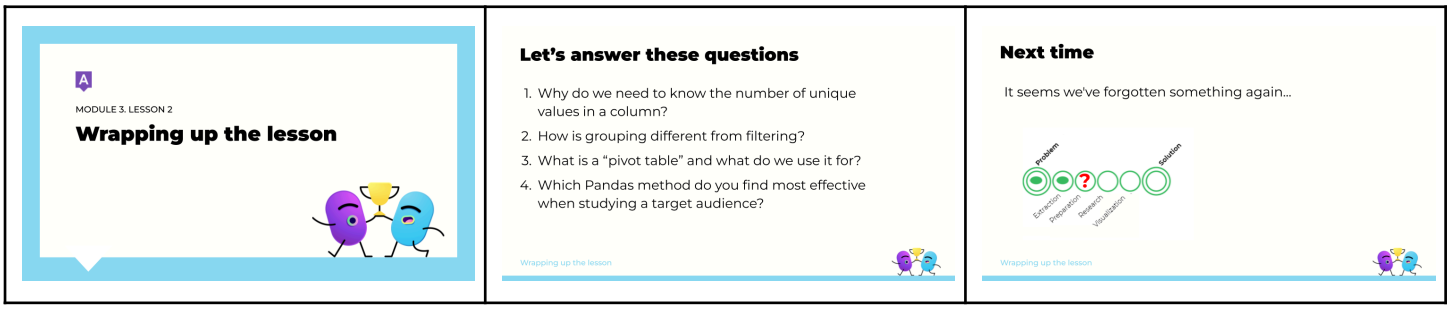
Wrapping up the lesson. Reflection

(5 min)

Ask the students to answer the questions to check their understanding of the new lesson material. Ask them to share their impression of the entire lesson and to answer the question about what was the most surprising/interesting for them.

Go back to the data analysis stages from the previous lesson with the students. Tell the children that we missed one of the important steps again: getting the data ready to work with.

Explain that this time it was done on purpose because getting the data ready (or cleaning it) requires that the analyst have a certain skill. That is why in their first lessons, they worked with a dataset that was prepared for them by specialists from the Algorithmics school. Tell the students that they are now knowledgeable enough to master the skill of data cleaning.



Answers to the tasks

Task 1.1

Reference code

```
import pandas as pd
df = pd.read_csv('GoogleApps.csv')

#1 How many apps are there in the 'BUSINESS' 'Category'?
print(df['Category'].value_counts())

#2 What is the ratio of apps for teenagers ('Teen') to those for children over 10 years old ('Everyone 10+')?
#Round the answer to the nearest hundredth.
temp = df['Content Rating'].value_counts()
print('Ratio:', round(temp['Teen'] / temp['Everyone 10+'], 2))

#3.1 What is the average 'Rating' of 'Paid' apps?
#Round the answer to the nearest hundredth.
temp = df.groupby(by = 'Type')['Rating'].mean()
print(temp['Paid'])

#3.2 How much lower is the average 'Rating' of 'Free' apps than the average rating of 'Paid' apps?
#Round the answer to the nearest hundredth.
print(round(temp['Paid'] - temp['Free'], 2))

#4 What are the minimum and maximum app 'Size' in the 'COMICS' 'Category'?
#Round the answer to the nearest hundredth.
```

```
print(df.groupby(by = 'Category')['Size'].agg(['min', 'max']))

#Bonus 1. How many apps have a 'Rating' of over 4.5 in the 'FINANCE' 'Category'?
temp = df[df['Rating'] > 4.5]['Category'].value_counts()
print(temp['FINANCE'])

#Bonus 2. What is the ratio of 'Free' to 'Paid' games with a 'Rating' greater than 4.9?
temp = df[(df['Category'] == 'GAME') & (df['Rating'] > 4.9)]['Type'].value_counts()
print(temp['Free'] / temp['Paid'])
```

Tasks 1.2–1.6

Part 1. My correct solution Attempt: 1/200

Fill in the blanks

Use the data that you were able to get from the dataset while working in VSC.

How many applications are there in the 'BUSINESS' category?

Enter your answer as an integer. 246

Part 1. My correct solution Attempt: 1/200

Fill in the blanks

Use the data that you were able to get from the dataset while working in VSC.

1. What is the average paid app rating? Round your answer to the nearest hundredth. 4.25

2. How much lower is the average rating of free apps than the average rating of paid ones? Round your answer to the nearest hundredth. 0.08

Part 1. My correct solution Attempt: 1/200

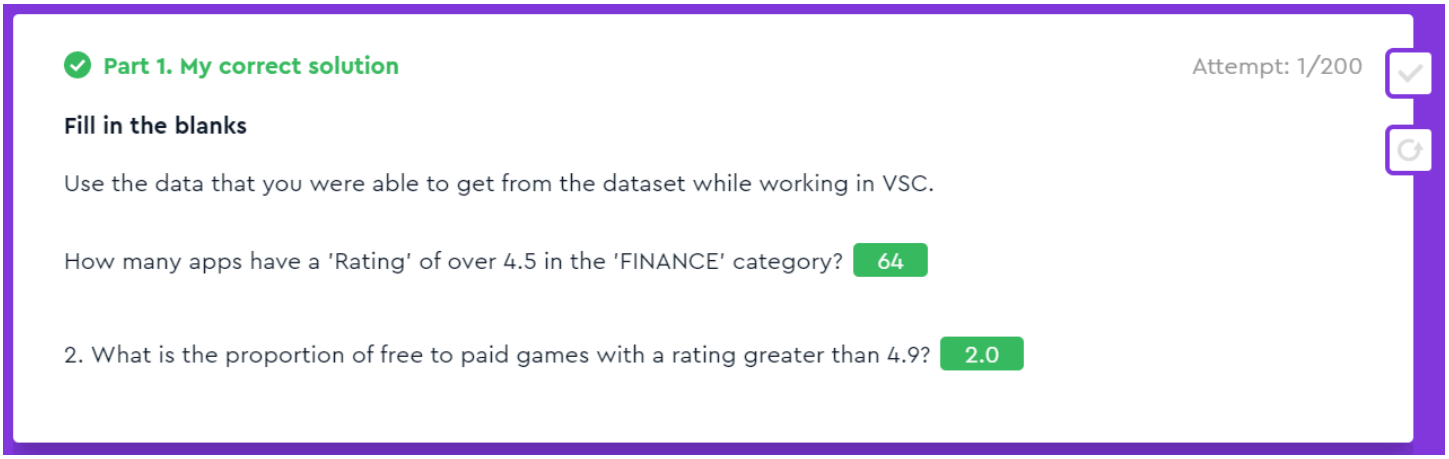
Fill in the blanks

Use the data that you were able to get from the dataset while working in VSC.

What are the minimum and maximum sizes of apps in the 'COMICS' category? Round your answers to the nearest hundredth.

min: 0.43

max: 40.0



Task 2.1

Reference code

```
import pandas as pd
df = pd.read_csv('GoogleApps.csv')

#1 Display the minimum, average, and maximum 'Rating' of paid and free apps ('Type') and
round to the nearest tenth.
print(round(df.groupby(by = 'Type')['Rating'].agg(['min', 'mean', 'max']), 1))

#2 Display the minimum, median, and maximum 'Price' of paid apps (Type == 'Paid') for #
different target audiences ('Content Rating')
print(df[df['Type'] == 'Paid'].groupby(by = 'Content Rating')['Price'].agg(['min', 'median',
'max']))

#3 Group the data by 'Category' and target audience ('Content Rating') however you prefer
#calculate the maximum number of 'Reviews' for each group.
#Compare the results for the 'EDUCATION', 'FAMILY', and 'GAME' categories:
#In which age group did the 'EDUCATION' category app receive the most reviews? 'FAMILY'?
'GAME'?

#Tip: You can select multiple columns from DataFrame at once using the following syntax:
# df[[<column 1>, <column 2>, <column 3>]]
temp = df.pivot_table(index = 'Content Rating', columns = 'Category', values = 'Reviews',
aggfunc = 'max')
print(temp[['EDUCATION', 'FAMILY', 'GAME']])

#4 Group the paid (Type == 'Paid') apps by 'Category' and target audience ('Content Rating')
#Calculate the average number of 'Reviews' for each group
```

```
#Please note that some cells in the resulting table have the value “NaN” – Not a Number – instead of a number
#That means that there are no apps in that group.
#Choose the names of the categories that have paid apps for all age groups and arrange them in alphabetical order.
print(df[df['Type'] == 'Paid'].pivot_table(columns = 'Content Rating', index = 'Category', values = 'Reviews', aggfunc = 'mean'))

#Bonus task. Find the categories of free (Type == 'Free') apps
#in which groups were the apps not designed for all age groups ('Content Rating')
print(df[df['Type'] == 'Free'].pivot_table(index = 'Category', columns = 'Content Rating', values = 'Reviews', aggfunc = 'mean'))
```

Tasks 2.2–2.6

Part 1. My correct solution Attempt: 1/200

Fill in the blanks in the text

What are the minimum, average, and maximum 'Rating' of paid and free apps ('Type') equal to? Round your answers to the nearest tenth.

Free_min 1.0 Free_mean 4.2 Free_max 5.0

Paid_min 1.0 Paid_mean 4.2 Paid_max 5.0

Part 1. My correct solution Attempt: 2/200

Select all the TRUE statements that can be established after grouping:

Grouping by minimum, median, and maximum price of paid applications (Type == 'Paid') for different target audiences ('Content Rating').

- [ ] The median price for each target audience is different
- [x] In the 'Teen' category, the median price is the highest among the entire target audience
- [ ] The maximum price of paid applications for all target audiences is the same
- [x] In the 'Everyone' category, the maximum price is the highest among the entire target audience
- [x] The minimum price of paid applications for all target audiences is the same

Part 1. My correct solution Attempt: 1/200

Fill in the blanks

Use the data that you were able to get from the dataset while working in VSC.

1. In which age group did the 'EDUCATION' category app receive the most reviews? Everyone

2. In which age group did the 'FAMILY' category app receive the most reviews? Everyone 10+

3. In which age group did the 'GAME' category app receive the most reviews? Everyone 10+

Part 1. My correct solution Attempt: 1/200

Which statement correctly lists the names of categories that have paid apps for all age groups.

Use the data that you were able to get from the dataset while working in VSC.

- ○ FAMILY
GAME
WEATHER
- ○ FAMILY
MEDICAL
- ○ BOOKS_AND_REFERENCE
GAME
MEDICAL
PERSONALIZATION
- ● FAMILY
GAME
MEDICAL

Part 1. My correct solution Attempt: 1/200

Fill in the blanks in the text

Use the data that you were able to get from the dataset while working in VSC.

How many free app categories (Type == 'Free') have apps that are not designed for all age groups ('Content Rating')?

Number of categories: 13

Filtering and tasks from the lesson that are solved using it

```
Task #1: How many times does the number of apps for the “all users” audience exceed the number of apps for the “10+” users?
Everyone = df[(df['Content Rating'] == 'Everyone')] ['Content Rating'].count()
Everyone10 = df[(df['Content Rating'] == 'Everyone 10+')] ['Content Rating'].count()
```

```
print('Proportion of apps for everyone to those for children:', Everyone/Everyone10)

Task #2: What is the average app size for each target audience?
print('Average app size for the Everyone audience:', df[df['Content Rating'] ==
'Everyone']['Size'].mean())
print('Average app size for the Everyone 10+ audience:', df[df['Content Rating'] == 'Everyone
10+']['Size'].mean())
print('Average app size for the Teen audience:', df[df['Content Rating'] ==
'Teen']['Size'].mean())

Task #3: What are the minimum and maximum size of the paid and free apps for each target
audience? # Legend: a) target audience E10 - Everyone 10+; E - Everyone; T -Teen b) app type
F - free; P - paid.

E10_min_F = df[(df['Content Rating'] == 'Everyone 10+') & (df['Type'] =='Free')]
['Size'].min()
E_min_F = df[(df['Content Rating'] == 'Everyone') & (df['Type'] =='Free')] ['Size'].min()
T_min_F = df[(df['Content Rating'] == 'Teen') & (df['Type'] =='Free')] ['Size'].min()
E10_min_F = df[(df['Content Rating'] == 'Everyone 10+') & (df['Type'] =='Free')]
['Size'].min()
E_min_F = df[(df['Content Rating'] == 'Everyone') & (df['Type'] =='Free')] ['Size'].min()
T_min_F = df[(df['Content Rating'] == 'Teen') & (df['Type'] =='Free')] ['Size'].min()
E10_max_F = df[(df['Content Rating'] == 'Everyone 10+') & (df['Type'] =='Free')]
['Size'].max()
E_max_F = df[(df['Content Rating'] == 'Everyone') & (df['Type'] =='Free')] ['Size'].max()
T_max_F = df[(df['Content Rating'] == 'Teen') & (df['Type'] =='Free')] ['Size'].max()
E10_min_P = df[(df['Content Rating'] == 'Everyone 10+') & (df['Type'] =='Paid')]
['Size'].min()
E_min_P = df[(df['Content Rating'] == 'Everyone') & (df['Type'] =='Paid')] ['Size'].min()
T_min_P = df[(df['Content Rating'] == 'Teen') & (df['Type'] =='Paid')] ['Size'].min()
E10_max_P = df[(df['Content Rating'] == 'Everyone 10+') & (df['Type'] =='Paid')]
['Size'].max()
E_max_P = df[(df['Content Rating'] == 'Everyone') & (df['Type'] =='Paid')] ['Size'].max()
T_max_P = df[(df['Content Rating'] == 'Teen') & (df['Type'] =='Paid')] ['Size'].max()
print('Minimum size of free apps:', 'E10 =', E10_min_F, ';',  'E =', E_min_F, ';',  'T =',
T_min_F)
print('Maximum size of free apps:', 'E10 =', E10_max_F, ';',  'E =', E_max_F, ';',  'T =',
T_max_F)
print('Minimum size of paid apps:', 'E10 =', E10_min_P, ';',  'E =', E_min_P, ';',  'T =',
T_min_P)
print('Maximum size of paid apps:', 'E10 =', E10_max_P, ';',  'E =', E_max_P, ';',  'T =',
T_max_P)
```