Mini Project

Members: Ellie Cheng, Stephen Dong, Selena Arias, Shashvat Gupta

1. What data do you have?

- We have data on the following questions from the survey:
  1. How many units are you taking currently?
  2. What is your current living situation?
  3. How do you commute to school?
  4. Do you have a UCR dining plan?
  5. How many times a week do you purchase food or drinks from somewhere on campus?
  6. What items do you purchase?
  7. What is the biggest reason you do not purchase more food and drinks on campus?

Each column consists of one of these questions, while each row consists of an individual's response to each question. Essentially, the data we have relates to the student's amount of units they are taking, their living situation and if they commute, whether or not they purchase food on campus, and if they don't what is the reason why.

2. What would you like to know?

We would like to discover if the amount of enrolled units and living situations of students affects how they obtain food on campus. We are attempting to determine if a student's total enrolled units affect if they buy food more often. Particularly, we are curious if a student who has at least 12 enrolled units buys food on campus more than a student enrolled in less than 12 units. We are also trying to determine if their living situation affects whether or not students purchase food on campus.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from scipy.stats import chi2_contingency
```

In [2]:
```
raw_data = pd.read_csv("CS105Fall2022.csv")
data = pd.DataFrame(raw_data[['18. How many units are you taking currently?',
            '54. What is your current living situation?',
            '55. How do you commute to school?',
            '56. Do you have a UCR dining plan?',
            '57. How many times a week do you purchase food or drinks from somewhere on campus?',
            '58. What items do you purchase?',
            '59. What is the biggest reason you do not purchase more food and drinks on campus?']])
data.rename(columns={'18. How many units are you taking currently?' : 'units',
            '54. What is your current living situation?': 'live situation',
            '55. How do you commute to school?' : 'commute',
            '56. Do you have a UCR dining plan?' : 'dining plan',
            '57. How many times a week do you purchase food or drinks from somewhere on campus?' : 'avg purc
            '58. What items do you purchase?' : 'items',
            '59. What is the biggest reason you do not purchase more food and drinks on campus?' : 'why nobu
          inplace=True)
max_num = np.shape(data)[0]
data
```

Out[2]:

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
0	16.0	Off-Campus (3-mile radius)	Walk	No	1	Entrees, Water	Financial issues
1	12.0	Off-Campus (3-mile radius)	Public Transportation	No	0	NaN	Expensive compared to other options
2	16.0	Off-Campus (4-30 mile radius)	Drive Yourself	No	0.5	Entrees	Not hungry
3	16.0	Off-Campus (3-mile radius)	Public Transportation	No	5	Entrees, Snacks, Desserts, Juice/Tea/Coffee	Not hungry
4	17.0	Off-Campus (4-30 mile radius)	Drive Yourself	No	3	Entrees, Snacks, Juice/Tea/Coffee	Not hungry
...	...	...	...	...	...	...	...
106	12.0	Off-Campus (3-mile radius)	Bicycle / Scooter / Skateboard	No	1	Snacks, Desserts	Prefer cooking/packing
107	13.0	On Campus	Walk	Yes	14	Entrees, Snacks, Desserts, Fruits/Salads, Soda...	NaN
108	12.0	On Campus	Walk	No	0	None	Lack of options
109	13.0	On Campus	Walk	Yes	4	Entrees, Snacks, Juice/Tea/Coffee	Financial issues
110	3.0	Off-Campus (3-mile radius)	Walk	No	3	Entrees, Snacks, Desserts, Juice/Tea/Coffee	Prefer cooking/packing

111 rows × 7 columns

Cleaning

In [3]:
```
# Manually drop and fill nans in avg purchases
na_data = data.loc[data['avg purchases'].isna()]
display(na_data)    # For displaying/testing purposes

data.drop(index=na_data.index.values[1:], inplace=True)
data['avg purchases'] = data['avg purchases'].fillna('0')

data.loc[data['avg purchases'].isna()]    # For displaying/testing purposes
```

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
31	12.0	Off-Campus (3-mile radius)	Walk	No	NaN	Snacks	No Time
36	NaN	NaN	NaN	NaN	NaN	NaN	NaN
38	12.0	NaN	NaN	NaN	NaN	NaN	NaN
88	NaN	NaN	NaN	NaN	NaN	NaN	NaN
91	15.0	Off-Campus (3-mile radius)	NaN	NaN	NaN	NaN	NaN

Out[3]:

units	live situation	commute	dining plan	avg purchases	items	why nobuy

In [4]:
```
# Manually fill nans and clean items
data['items'] = data['items'].fillna('')
bad_data = data.loc[(~data['items'].str.contains('Desserts|Entrees|Fruits/Salads|Juice/Tea/Coffee|Snacks|Soda|W
                    (data['items'] != '')]
display(bad_data)

data.loc[108, 'items'] = ''   # None -> '' (na)
# data.drop(index=bad_data.index.values[1:], inplace=True)

data.loc[(~data['items'].str.contains('Desserts|Entrees|Fruits/Salads|Juice/Tea/Coffee|Snacks|Soda|Water')) &
         (data['items'] != '')]    # For displaying/testing purposes
```

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
108	12.0	On Campus	Walk	No	0	None	Lack of options

Out[4]:

units	live situation	commute	dining plan	avg purchases	items	why nobuy

In [5]:
```
# Manually Fixing Non-numeric Values
non_num_data = data.loc[data['avg purchases'].apply(str).str.isnumeric() == False]    # For displaying/testing p
display(non_num_data)

data.loc[2, 'avg purchases'] = 0.5   # str to float
data.drop(index=10, inplace=True)   # invalid data
data.loc[43, 'avg purchases'] = (2 * 7)   # 2x per day per week
data.loc[78, 'avg purchases'] = 1   # ~1 => 1

data.loc[data['avg purchases'].apply(str).str.isnumeric() == False]    # For displaying/testing purposes
```

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
2	16.0	Off-Campus (4-30 mile radius)	Drive Yourself	No	0.5	Entrees	Not hungry
10	16.0	Off-Campus (4-30 mile radius)	Drive Yourself	No	every day	Snacks, Soda, Water	Lack of options
43	12.5	Off-Campus (4-30 mile radius)	Public Transportation	Yes	twice a day	Entrees, Juice/Tea/Coffee	Lack of options
78	16.5	Off-Campus (4-30 mile radius)	Drive Yourself	No	~1	Entrees, Snacks, Fruits/Salads, Water	low quality fast food

Out[5]:

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
2	16.0	Off-Campus (4-30 mile radius)	Drive Yourself	No	0.5	Entrees	Not hungry

In [6]:
```
# Convert avg purchases to numerical
data['avg purchases'] = data['avg purchases'].astype('float64')
data.dtypes
```

Out[6]:
```
units              float64
live situation      object
commute             object
dining plan         object
avg purchases      float64
items               object
why nobuy           object
dtype: object
```

In [7]:
```
# Remove non-valid (outlier) unit counts
non_valid_unit = data.loc[data['units'] > 28.0]
display(non_valid_unit)    # For displaying/testing purposes

data.drop(index=non_valid_unit.index.values, inplace=True)
data.loc[data['units'] > 28.0]    # For displaying/testing purposes
```

	units	live situation	commute	dining plan	avg purchases	items	why nobuy
69	128.5	Off-Campus (4-30 mile radius)	Drive Yourself	No	1.0	Snacks	No Time

Out[7]:

units	live situation	commute	dining plan	avg purchases	items	why nobuy

In [8]:
```
# Filter and drop invalid responses: if avg purchases != 0.0, but items = 0.0
invalid_purchases = data.loc[(data['items'] == '') & (data['avg purchases'] != 0.0)]
display(invalid_purchases)    # For displaying/testing purposes

data.drop(index=invalid_purchases.index.values, inplace=True)
data.loc[(data['items'] == '') & (data['avg purchases'] != 0.0)]    # For displaying/testing purposes
```

units	live situation	commute	dining plan	avg purchases	items	why nobuy

Out[8]:

units	live situation	commute	dining plan	avg purchases	items	why nobuy

Data Transform

In [9]:
```
units = pd.DataFrame(data.units)

# splitting item purchased of each into a list
indv_items = pd.DataFrame(data['items'].apply(lambda x: x.split(', ')))
indv_items['indx'] = pd.DataFrame(range(0, max_num), columns=['indx'])

# convert list into appropriate columns and data values to True/False
indv_items = pd.pivot_table(indv_items.explode('items'),
                            values='items', index='indx',
                            columns='items', aggfunc=lambda x: True)
indv_items = indv_items.fillna(False)
indv_items = indv_items.rename(columns={'' : 'None'})
units_types = units.join(indv_items)
units_types
```

Out[9]:

	units	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
0	16.0	False	False	True	False	False	False	False	True
1	12.0	True	False	False	False	False	False	False	False
2	16.0	False	False	True	False	False	False	False	False
3	16.0	False	True	True	False	True	True	False	False
4	17.0	False	False	True	False	True	True	False	False
...	...	...	...	...	...	...	...	...	...
106	12.0	False	True	False	False	False	True	False	False
107	13.0	False	True	True	True	True	True	True	True
108	12.0	True	False	False	False	False	False	False	False
109	13.0	False	False	True	False	True	True	False	False
110	3.0	False	True	True	False	True	True	False	False

105 rows × 9 columns

3. Explore the data. (Generate statistics, perform visualizations)

Explain what you are computing (mean, SD, ...), and then compute using Python.

Create some visualizations (at least 7, and at least 5 must be of different types); use Python.

Visualization 1:

Number of Units Taking and Number of Items Purchased

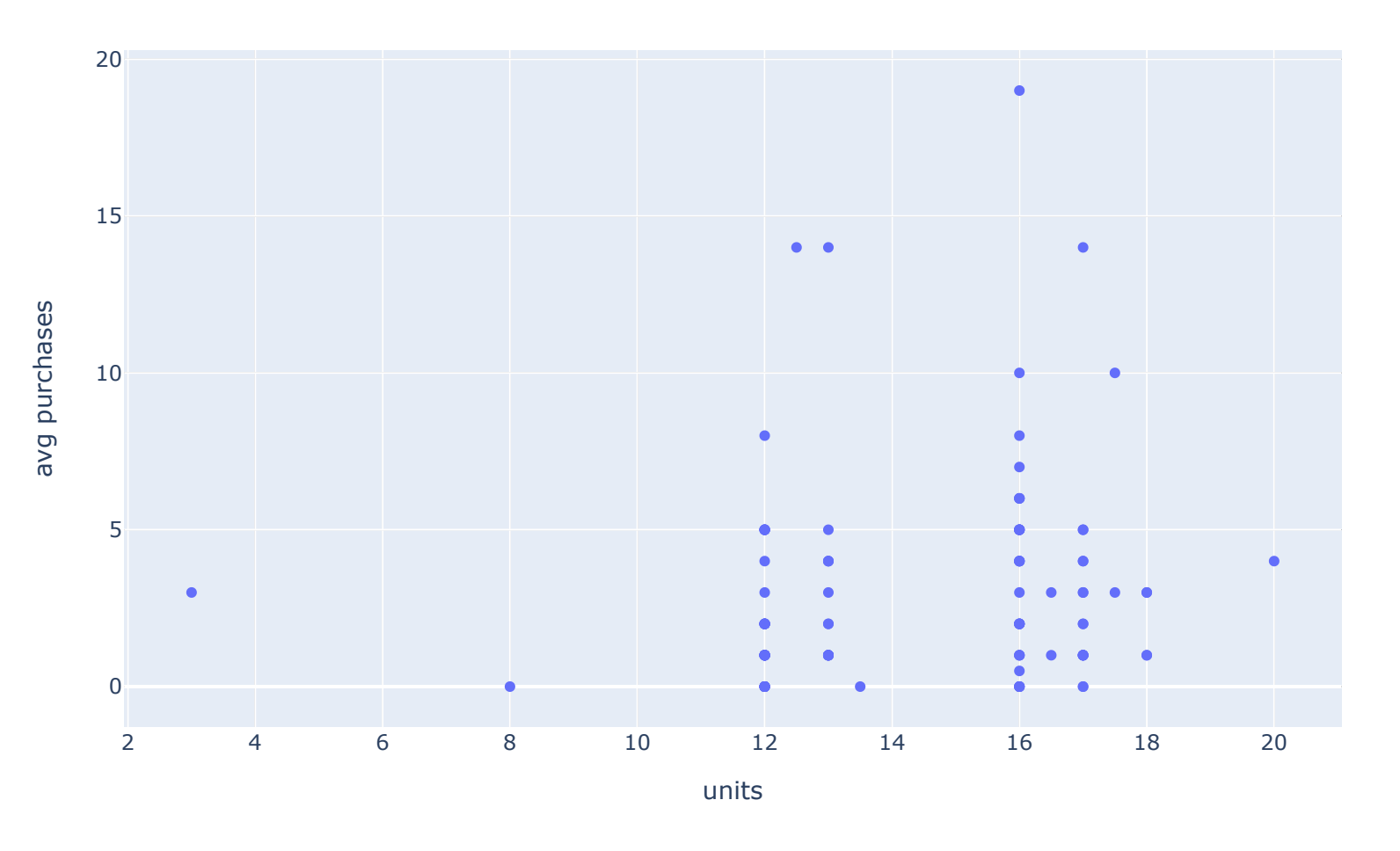
In [10]:
```
units_num_items = pd.DataFrame(data.loc[:, ['units', 'avg purchases']])
units_num_items
```

Out[10]:

	units	avg purchases
0	16.0	1.0
1	12.0	0.0
2	16.0	0.5
3	16.0	5.0
4	17.0	3.0
...	...	...
106	12.0	1.0
107	13.0	14.0
108	12.0	0.0
109	13.0	4.0
110	3.0	3.0

105 rows × 2 columns

In [11]:
```
units_num_items_fig = px.scatter(data_frame=units_num_items, x='units', y='avg purchases')
units_num_items_fig.show()
```



Explanation: We directly plotted the units on the x-axis and the average purchases on the y axis to see any patterns. In the figure above, we can see there's a slight positive trend for the first 10 units, this trend becomes lost after the first 10 when the values begins spreading out more. Although, we can see that there may be a slight correlation such that we can determine what the possible average number of purchases may be. For example, we can say that a student with more units is more likely to be around or below 5 purchases per week; however, given that the correlation is weak, we can't say for sure.

Visualization 2:

Number of Units Taking and Types of Items Purchased

In [12]:
```
# Create joint distribution table
units_types_ct = pd.pivot_table(units_types, index='units', aggfunc='sum')
display(units_types_ct)

# Sum of total # total different items
sums = sum(units_types_ct.sum())
sums
```

units	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	None	Snacks	Soda	Water
3.0	1	1	0	1	0	1	0	0
8.0	0	0	0	0	1	0	0	0
12.0	1	19	2	9	4	9	3	6
12.5	0	1	0	1	0	0	0	0
13.0	1	12	3	8	0	9	1	2
13.5	0	0	0	0	1	0	0	0
16.0	4	18	3	12	3	18	0	4
16.5	0	1	1	0	0	2	0	1
17.0	0	9	1	7	2	7	4	1
17.5	0	2	0	1	0	2	1	0
18.0	1	4	2	1	0	2	0	1
20.0	0	1	0	0	0	0	0	0

Out[12]: 213

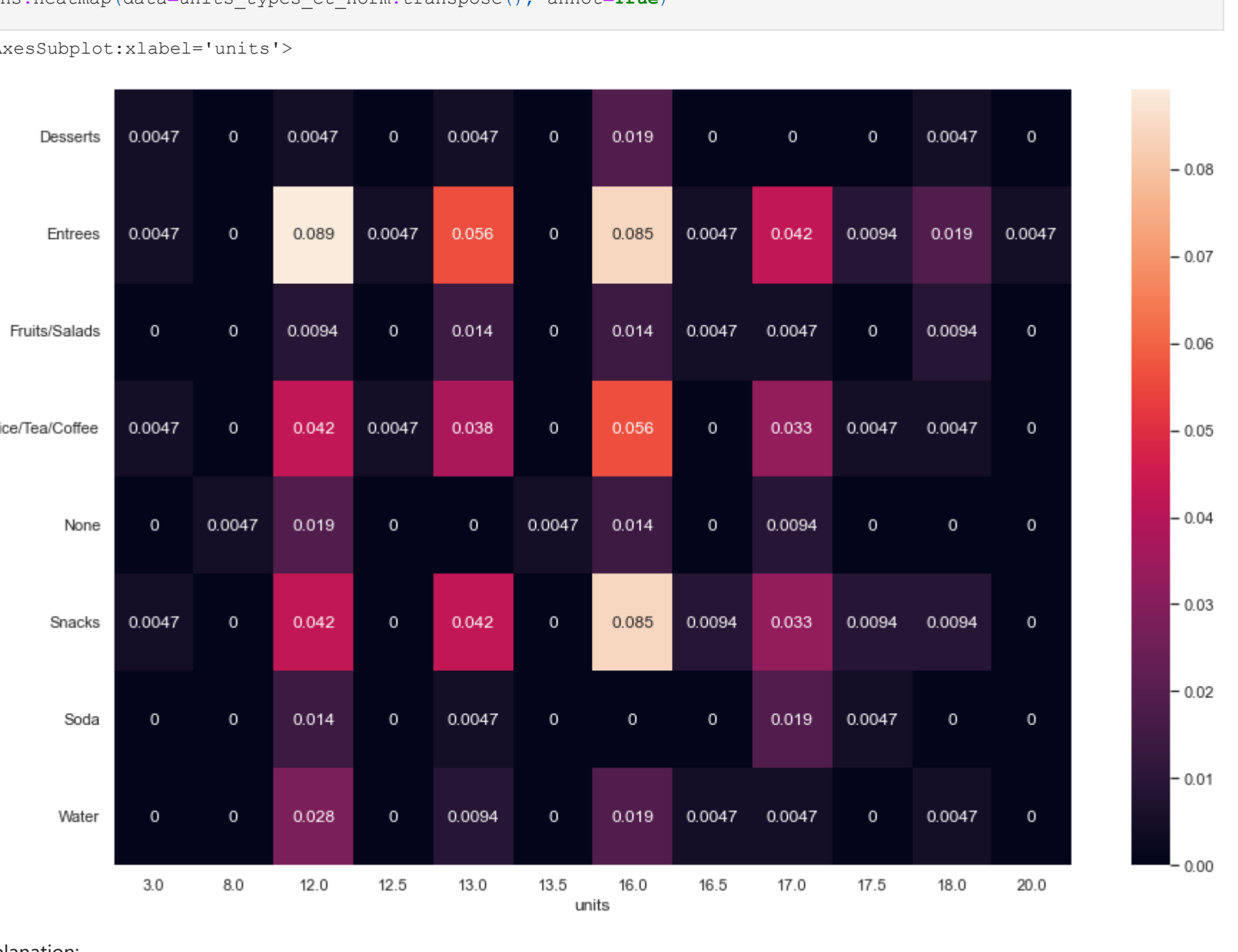
In [13]:
```
# Normalize Join Table
units_types_ct_norm = units_types_ct / sums
units_types_ct_norm
```

Out[13]:

units	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	None	Snacks	Soda	Water
3.0	0.004695	0.004695	0.000000	0.004695	0.000000	0.004695	0.000000	0.000000
8.0	0.000000	0.000000	0.000000	0.000000	0.004695	0.000000	0.000000	0.000000
12.0	0.004695	0.089202	0.009390	0.042254	0.018779	0.042254	0.014085	0.028169
12.5	0.000000	0.004695	0.000000	0.004695	0.000000	0.000000	0.000000	0.000000
13.0	0.004695	0.056338	0.014085	0.037559	0.000000	0.042254	0.004695	0.009390
13.5	0.000000	0.000000	0.000000	0.000000	0.004695	0.000000	0.000000	0.000000
16.0	0.018779	0.084507	0.014085	0.056338	0.014085	0.084507	0.000000	0.018779
16.5	0.000000	0.004695	0.004695	0.000000	0.000000	0.009390	0.000000	0.004695
17.0	0.000000	0.042254	0.004695	0.032864	0.009390	0.032864	0.018779	0.004695
17.5	0.000000	0.009390	0.000000	0.004695	0.000000	0.009390	0.004695	0.000000
18.0	0.004695	0.018779	0.009390	0.004695	0.000000	0.009390	0.000000	0.004695
20.0	0.000000	0.004695	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

In [14]:
```
sns.set(rc={"figure.figsize":(15, 10)})
sns.heatmap(data=units_types_ct_norm.transpose(), annot=True)
```

Out[14]: <AxesSubplot:xlabel='units'>



Explanation:
This visualization was done as we are comparing 2 dimensions, but are interested in the values the two variables give. Since the values give a count, we used a heatmap to display when our value is significant versus when it is not. Units is labeled on the x-axis, with item types labeled on the y-axis. For this visualization, the values were normalized as it gave a slightly better view of how much the units affect their choices. We can determine from this visualization that units don't really have an effect on what they tend to buy as it is evenly distributed amongst 3 main types across the board. However, we can determine that most students tend to purchases entrees, and students with higher course loads given the number of units will have a higher chance of purchasing an item than others; although, they are all relatively likely to.

Visualization 3

How a student's method of commuting affect how many items they purchase

In [15]:
```
commute_table = pd.pivot_table(data=data, values = 'avg purchases', index = 'commute', aggfunc = 'count')
commute_table
```

Out[15]:

commute	avg purchases
Bicycle / Scooter / Skateboard	16
Carpool	1
Drive Yourself	34
Driven (Dropped-off)	4
Public Transportation	5
Walk	45

Group 1

In [16]:
```
#Table of group1: non-car consists of  Walk, Public transportation, and Bicycle / Scooter / Skateboard
group1 = pd.pivot_table(data=data, values = 'avg purchases', index = 'commute', aggfunc = 'count')
group1.loc[[ 'Walk' ,'Public Transportation', 'Bicycle / Scooter / Skateboard']]
```

Out[16]:

commute	avg purchases
Walk	45
Public Transportation	5
Bicycle / Scooter / Skateboard	16

In [17]:
```
#Take group1 mean in terms of how many items they purchase
group1 = pd.pivot_table(data=data, values = 'avg purchases', index = 'commute', aggfunc = 'count')
group1_combined = group1.loc[[ 'Walk' ,'Public Transportation', 'Bicycle / Scooter / Skateboard']].mean()[0]
group1.loc[[ 'Walk' ,'Public Transportation', 'Bicycle / Scooter / Skateboard']].mean()
```

Out[17]:
```
avg purchases    22.0
dtype: float64
```

Group 2

In [18]:
```
#Table of group2 consists of car which are 'Carpool' ,'Driven (Dropped-off)', 'Drive Yourself'
group2 = pd.pivot_table(data=data, values = 'avg purchases', index = 'commute', aggfunc = 'count')
group2.loc[[ 'Carpool' ,'Driven (Dropped-off)', 'Drive Yourself']]
```

Out[18]:

commute	avg purchases
Carpool	1
Driven (Dropped-off)	4
Drive Yourself	34

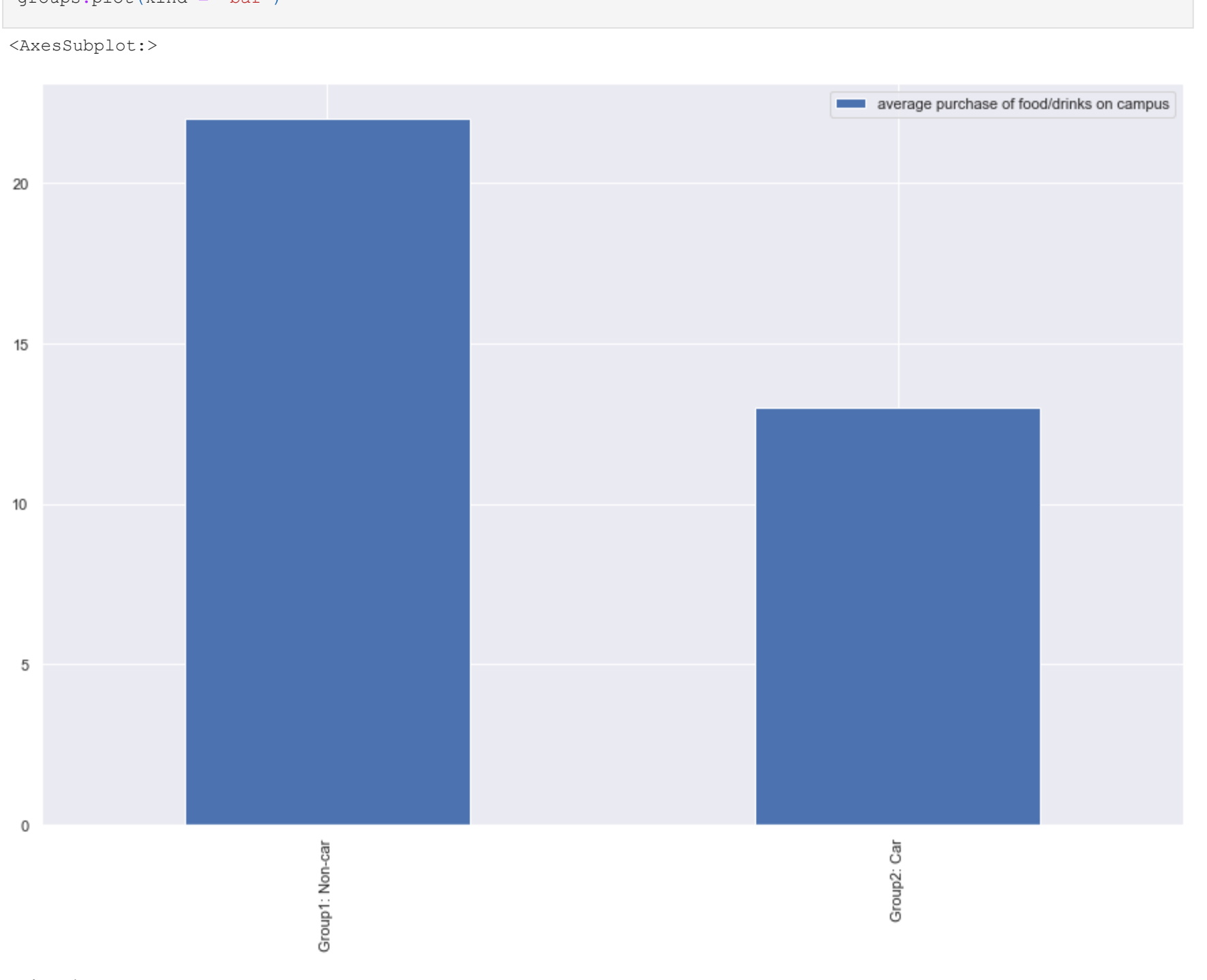
In [19]:
```
#Take group2 mean in terms of how many items they purchase
group2 = pd.pivot_table(data=data, values = 'avg purchases', index = 'commute', aggfunc = 'count')
group2_combined = group2.loc[[ 'Carpool' ,'Driven (Dropped-off)', 'Drive Yourself']].mean()[0]
group2.loc[[ 'Carpool' ,'Driven (Dropped-off)', 'Drive Yourself']].mean()
```

Out[19]:
```
avg purchases    13.0
dtype: float64
```

In [20]:
```
#plot the new sub groups into 2 groups noncar vs car
group_data = [group1_combined, group2_combined]
groups = pd.DataFrame(group_data)
mapping = {groups.columns[0]:'average purchase of food/drinks on campus'}
groups = groups.rename(columns=mapping)

mappings = {groups.index[0]:'Group1: Non-car', groups.index[1]:'Group2: Car'}
groups = groups.rename(index=mappings)
groups.plot(kind = 'bar')
```

Out[20]: <AxesSubplot:>



Explanation:
Here we defined two subgroups which consists of Group1:Non-Car and Group2:Car. Group1:Non-car consists of groups 'Walk, Public Transportation, and Bicycle / Scooter / Skateboard)' while Group2:Car consists of 'Carpool , Driven (Dropped-off), Drive Yourself'. We first figure out the count for each individual group then split them into Group1:Non-car and Group2:Car. By doing so, we take the mean for Group1:Non-car which is 22.0, and the mean for Group2:Car which is 13.0. We can conclude that those who do not commute by car (aka Group1:Non-car) are the ones who end up purchasing more food/drinks on campus than those who commute by car (aka Group2:Car) as seen on the graph.

Visualization 4

Given a living condition, on average what items do they purchase?

In [21]:
```
live_sit = pd.DataFrame(data['live situation'])
live_sit['live code'] = live_sit["live situation"].astype("category").cat.codes
live_sit['live code'] = live_sit['live code'].apply(lambda x: x if x == 3 else 0)
live_sit = live_sit.join(indv_items)
live_sit.head()    # live_sit = ls
```

Out[21]:

	live situation	live code	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
0	Off-Campus (3-mile radius)	0	False	False	True	False	False	False	False	True
1	Off-Campus (3-mile radius)	0	True	False	False	False	False	False	False	False
2	Off-Campus (4-30 mile radius)	0	False	False	True	False	False	False	False	False
3	Off-Campus (3-mile radius)	0	False	True	True	False	True	True	False	False
4	Off-Campus (4-30 mile radius)	0	False	False	True	False	True	True	False	False

In [22]:
```
live_sit_on = (live_sit.loc[live_sit['live code'] == 3])
ls_on_d = live_sit_on.drop(columns=["live situation", "live code"])
live_sit_off = live_sit.loc[live_sit['live code'] == 0]
ls_off_d = live_sit_off.drop(columns=["live situation", "live code"])

print(f"Total: {np.shape(live_sit)}")
print(f"On Campus: {np.shape(ls_on_d)}")
print(f"Off Campus: {np.shape(ls_off_d)}")

display(ls_on_d.head())
display(ls_off_d.head())
```

```
Total: (105, 10)
On Campus: (37, 8)
Off Campus: (68, 8)
```

	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
5	False	False	False	False	True	True	False	False
7	False	False	True	False	True	False	False	False
8	False	False	True	False	True	True	False	False
12	False	False	True	False	False	True	True	False
15	False	True	True	True	False	True	False	False

	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
0	False	False	True	False	False	False	False	True
1	True	False	False	False	False	False	False	False
2	False	False	True	False	False	False	False	False
3	False	True	True	False	True	True	False	False
4	False	False	True	False	True	True	False	False

In [23]:
```
# Rebalancing data by appending random sample from same dataset
if np.shape(ls_on_d)[0] <= np.shape(ls_off_d)[0]:
    dif = np.shape(ls_off_d)[0] - np.shape(ls_on_d)[0]
    rebalance_df = ls_on_d.sample(dif)
    ls_on_d = ls_on_d.append(rebalance_df)
else:
    dif = np.shape(ls_on_d)[0] - np.shape(ls_off_d)[0]
    ls_on_d = rebalance_df = ls_off_d.sample(dif)

display(ls_on_d)
display(ls_off_d)
```

	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
5	False	False	False	False	True	True	False	False
7	False	False	True	False	True	False	False	False
8	False	False	True	False	True	True	False	False
12	False	False	True	False	False	True	True	False
15	False	True	True	True	False	True	False	False
...	...	...	...	...	...	...	...	...
93	False	False	False	False	True	False	False	False
67	False	False	True	False	True	False	False	False
15	False	True	True	True	False	True	False	False
94	False	False	True	False	True	True	False	False
72	False	False	False	False	False	False	False	True

68 rows × 8 columns

	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water
0	False	False	True	False	False	False	False	True
1	True	False	False	False	False	False	False	False
2	False	False	True	False	False	False	False	False
3	False	True	True	False	True	True	False	False
4	False	False	True	False	True	True	False	False
...	...	...	...	...	...	...	...	...
103	False	False	True	False	False	False	False	False
104	False	False	True	False	False	False	False	False
105	True	False	False	False	False	False	False	False
106	False	True	False	False	False	True	False	False
110	False	True	True	False	True	True	False	False

68 rows × 8 columns

In [24]:
```
ls_on_s = list(ls_on_d.sum())
ls_off_s = list(ls_off_d.sum())
print(f"On campus: {ls_on_s}")
print(f"Off campus: {ls_off_s}")
```

```
On campus: [2, 7, 45, 10, 32, 41, 9, 6]
Off campus: [10, 4, 43, 7, 22, 27, 4, 12]
```

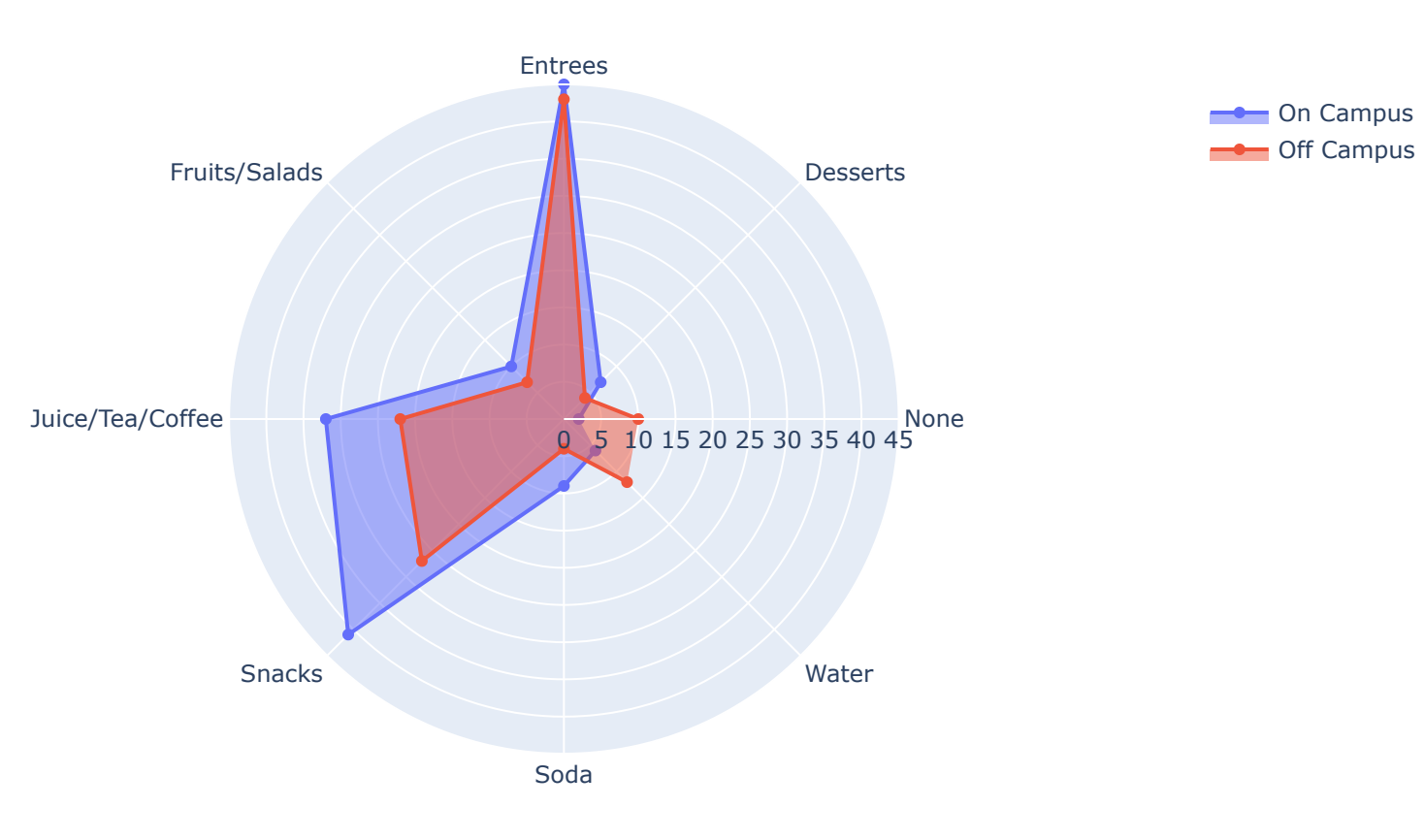
In [25]:
```
spider_categories = live_sit.columns.values[2:]
spider_categories
```

Out[25]:
```
array(['None', 'Desserts', 'Entrees', 'Fruits/Salads', 'Juice/Tea/Coffee',
       'Snacks', 'Soda', 'Water'], dtype=object)
```

In [26]:
```
spider_fig = go.Figure()

spider_max = max(max(ls_on_s), max(ls_off_s))
spider_fig.update_traces(fill='toself')
spider_fig.add_trace(go.Scatterpolar(r=ls_on_s,
                                     theta=spider_categories,
                                     fill='toself',
                                     name="On Campus"))
spider_fig.add_trace(go.Scatterpolar(r=ls_off_s,
                                     theta=spider_categories,
                                     fill='toself',
                                     name="Off Campus"))
spider_fig.update_layout(polar=dict(radialaxis=dict(visible=True, range=[0, spider_max])),
                         showlegend=True)

spider_fig.show()
```



Explanation:
We used a spider/radar chart to plot and compare the choices amongst students of different campuses and see exactly how they differ in decisions. We set the categories θ to be the individual possible options the students can choose from and set the groups to be On-campus and Off-campus. From this visualization, we split the dataset into the two groups and summed the individual items per group. Additionally, since the data is inbalanced between the groups, we picked the minority group and randomly sampled values within its group dataset to fill in the missing data in order to rebalanced the data to give a slightly more accurate representation. By analyzing this chart, we can see very clearly that Off-campus students are likely to purchase less of the smaller-sized items (i.e. fruits/salds/snacks/sodas), but are just as equally likely to purchases entrees. Likewise, they're also more likely to simply not purchase anything at all. This can confirms that on-campus students, as they spend more time on-campus besides from classes, they're more likely to purchase things. Meanwhile, off-campus students, they're less likely to purchase items, but since they still need a full meal, they are still willing to purchase an entree for lunch, etc.

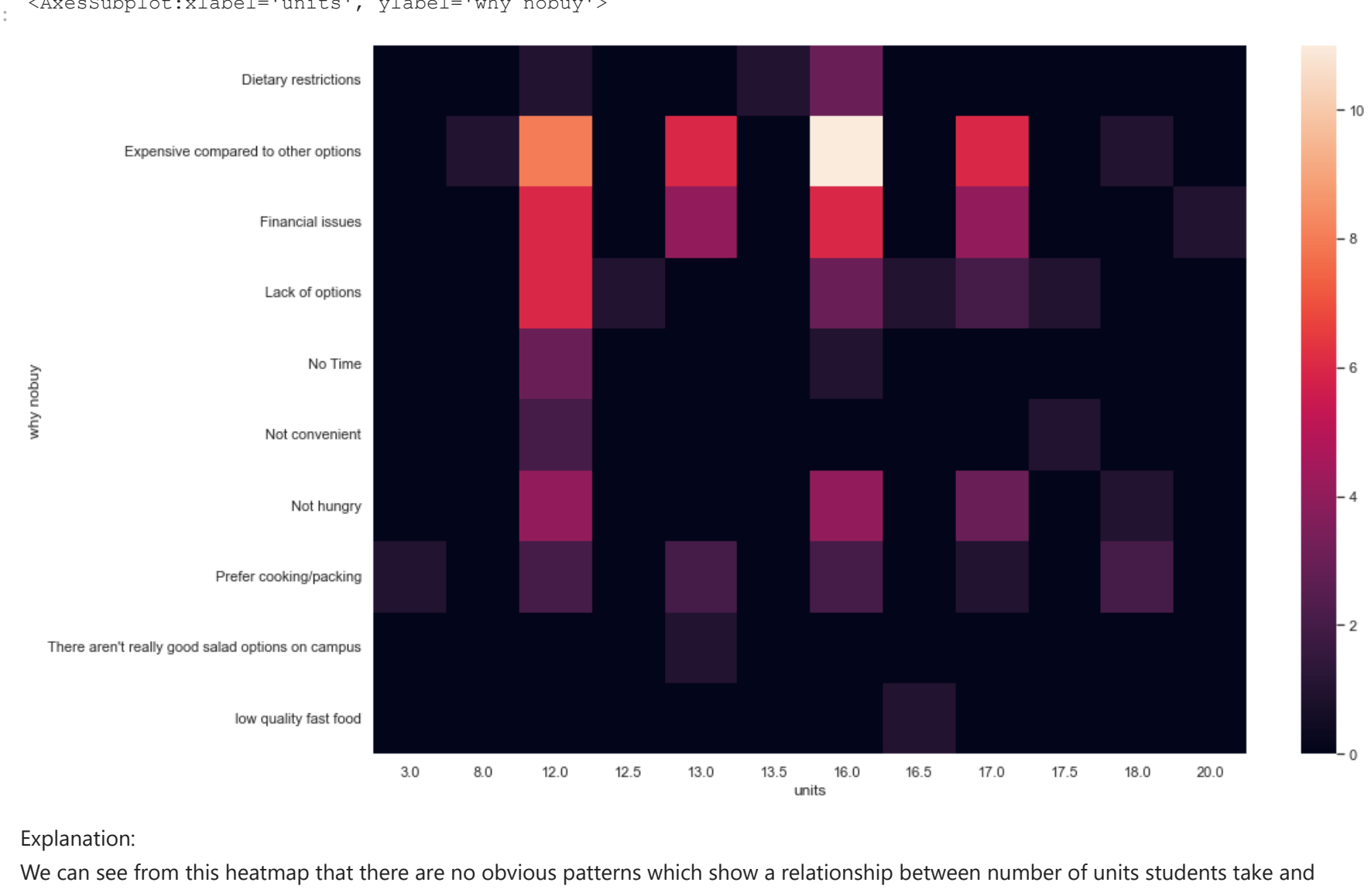
Visualization 5

Number of units affects greatest reason they don't buy more on campus

In [27]:
```
units_why_ct = pd.crosstab(index=data.units, columns=data['why nobuy'], margins=False)
```

In [28]:
```
sns.heatmap(data=units_why_ct.transpose())
```

Out[28]: <AxesSubplot:xlabel='units', ylabel='why nobuy'>



Explanation:
We can see from this heatmap that there are no obvious patterns which show a relationship between number of units students take and the biggest reason why they don't purchase more food on campus. However, even though number of units does not appear to make a big difference in students' reasons, we can see that the most prominent reasons regardless of number of units are financial, either because students believe options on campus are more expensive compared to other options or they have financial issues.

Visualization 6

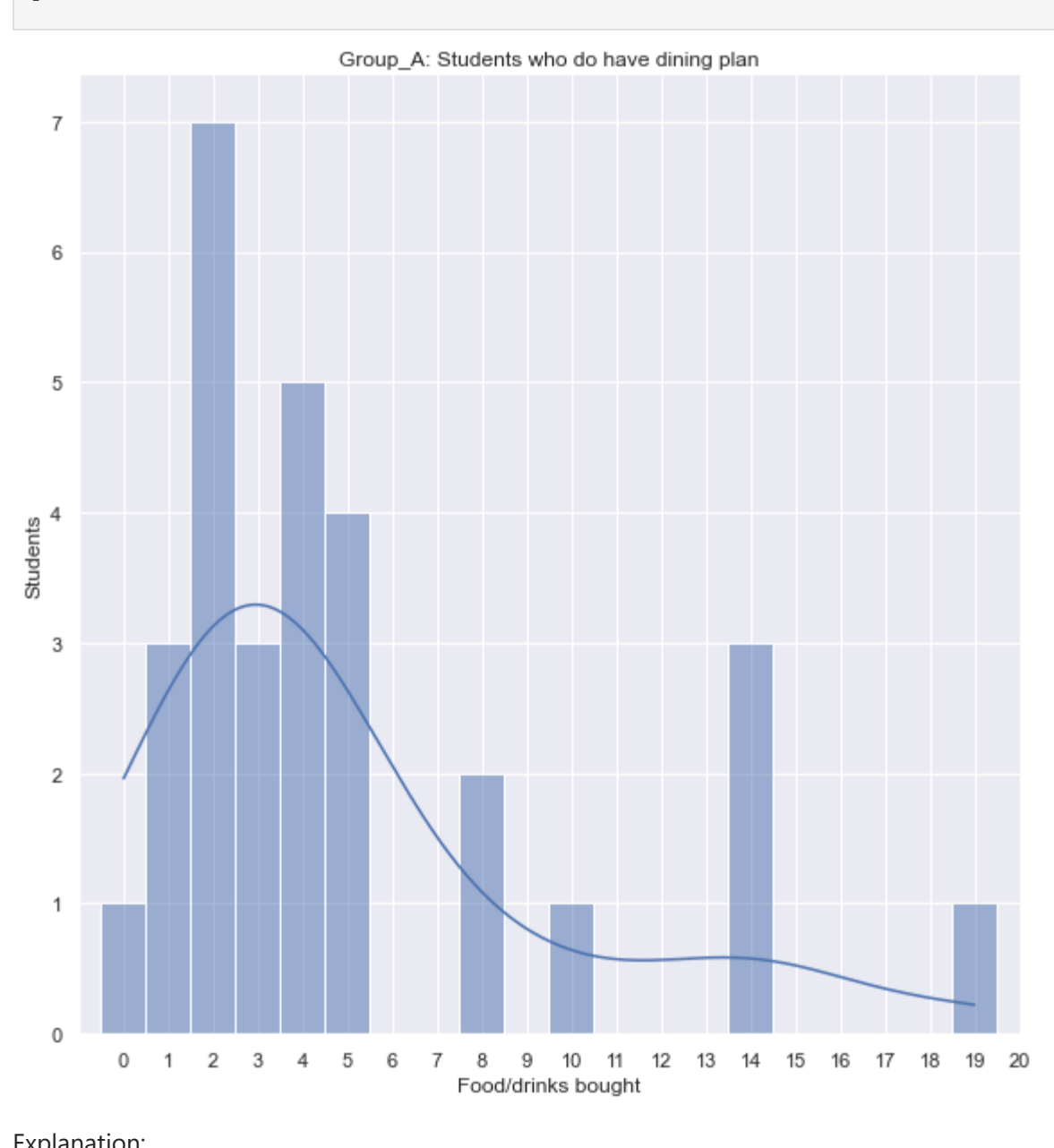
In [29]:
```
%matplotlib inline
#Group_a: students who have dining plan
group56_57 = data[['dining plan', 'avg purchases']]
group56_57_yes = group56_57.loc[group56_57['dining plan'] == "Yes"]
group56_57_yes = group56_57_yes.drop('dining plan', axis=1)
group56_57_yes = group56_57_yes.reset_index(drop=True)

students = group56_57_yes.value_counts().sum()
total_items = group56_57_yes['avg purchases'].sum()

total_items / students
#The mean for group_a: students who do have a dining plan
#if we were to exclude the outlier is would instead be 80 (total items) / 22(students) = average of 3.6 items
```

Out[29]: 5.1

```
In [30]:
# Plot of students who do have a UCR dining plan,
# and how many times a week they purchase food/drink on campus
bins = np.arange(26) - 0.5
sns.displot(data=group56_57_yes, bins=bins, kde=True, height=8, legend=False)
plt.xticks(range(32))
plt.xlim([-1,20])
plt.title('Group_A: Students who do have dining plan')
plt.xlabel('Food/drinks bought')
plt.ylabel('Students')
plt.show()
```



Explanation:
For Group_A:Students who do have dining plan, there are a total of 30 students with a total of 153 food/drinks. By taking the mean of the total amount of food/drinks bought and # of students who do have dining plan we got the average of 5.1 items per week.

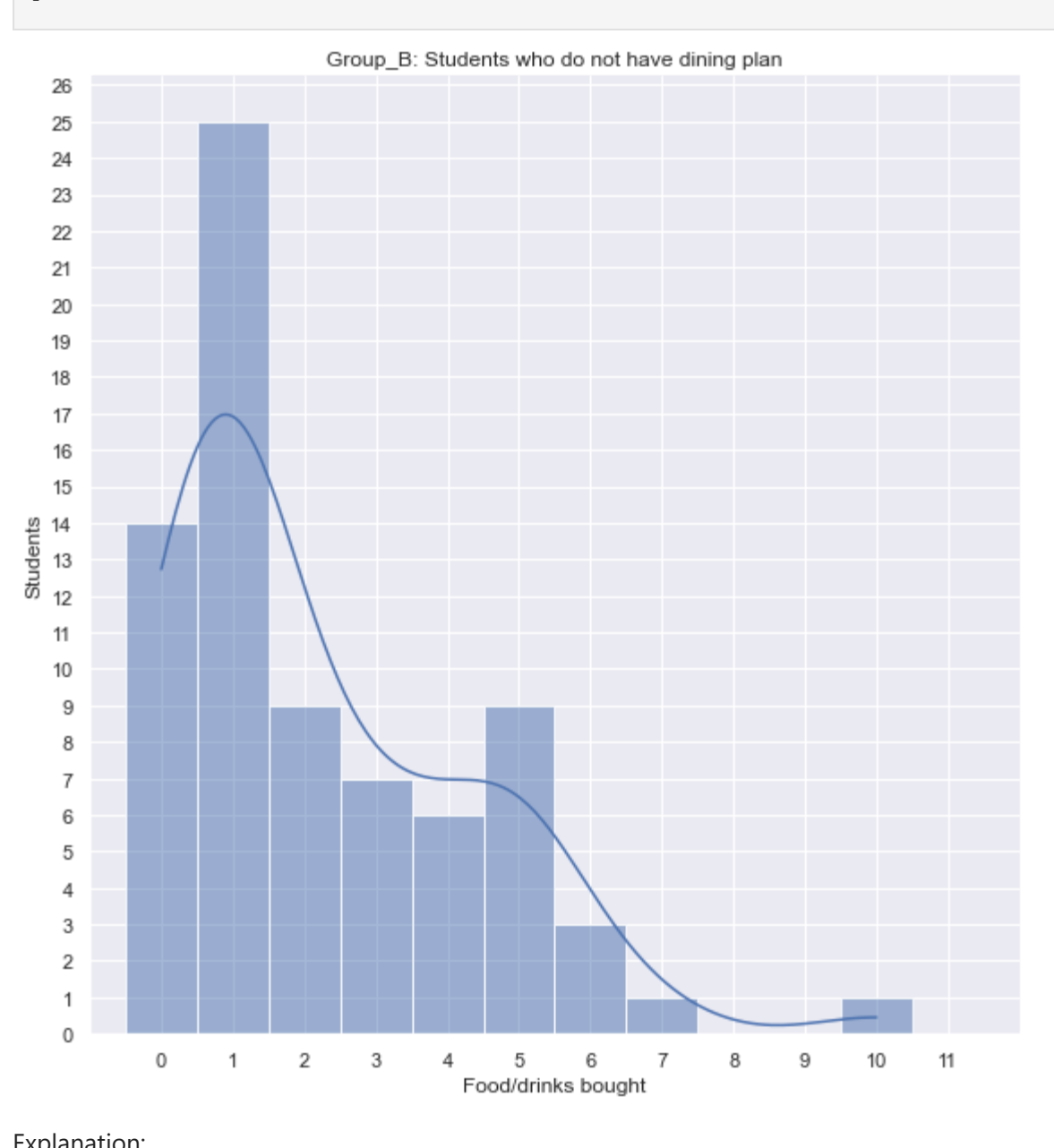
```
In [31]:
#group_b: students who do not have a dining plan
group56_57 = data[['dining plan', 'avg purchases']]
group56_57_no = group56_57.loc[group56_57['dining plan'] == "No"]
group56_57_no = group56_57_no.drop('dining plan', axis=1)
group56_57_no = group56_57_no.reset_index(drop=True)
```

```
In [32]:
#The mean for group_b: students who do not have a dining plan
students = group56_57_no.value_counts().sum()
total_items = group56_57_no['avg purchases'].sum()

total_items / students
#avereage amount of items bought for group_b
```

```
Out[32]: 2.2333333333333334
```

```
In [33]:
# Plot of students who do NOT have a UCR dining plan,
# and how many times a week they purchase food/drink on campus
bins = np.arange(57) - 0.5
sns.displot(data=group56_57_no, bins=bins, kde=True, height=8, legend=False)
plt.xticks(range(12))
plt.yticks(range(27))
plt.xlim([-1,12])
plt.title('Group_B: Students who do not have dining plan')
plt.xlabel('Food/drinks bought')
plt.ylabel('Students')
plt.show()
```



Explanation:
For Group_B: Students who do not have dining plan, there are a total of 75 students with a total of 168 food/drinks.By taking the mean of the total amount of food/drinks bought and # of students who do not have dining plan we got the average of 2.2333333333333334 items.

Summary:
Here we defined two groups which consists of "Group_A:Students who have dining plan" and "Graph_B: Students who do not have dining plan". We can see that Group_a has a bigger average of items than Group_B which is surprising due to the fact that we believed that it would be the other way around. We can conclude that those who have a dining plan end up buying more food/drinks than those who do not have a dining plan.

```
In [34]:
# Visualization 7
```

4. Can you state any hypotheses or make some predictions? Which tests can you apply to verify your hypothesis?

State clearly each of your hypotheses (at least 3).

Hypothesis 1: The number of units a student is taking affects average number of items they will purchase per week.

- We can use Pearson correlation analysis to verify hypothesis 1.

Hypothesis 2: There is no correlation between the number of units a student is taking and the types of items they purchase.

- We can use chi-square test to verify hypothesis 2.

Hypothesis 3: A student's living condition (where they live + dining plan) does not affect the type of items they purchase.

- We can use chi-square test to verify hypothesis 3.

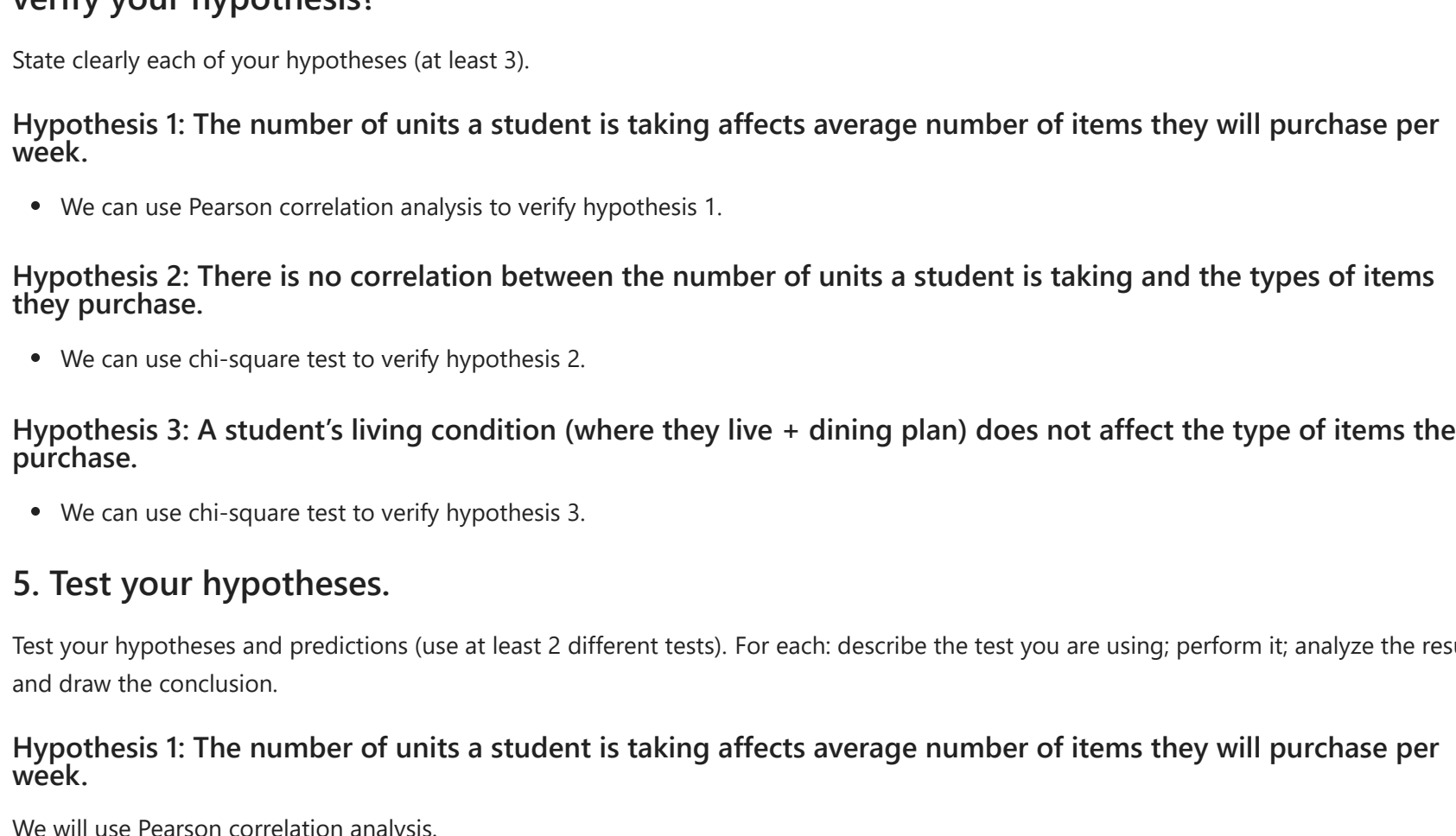
5. Test your hypotheses.

Test your hypotheses and predictions (use at least 2 different tests). For each: describe the test you are using; perform it; analyze the results and draw the conclusion.

Hypothesis 1: The number of units a student is taking affects average number of items they will purchase per week.

We will use Pearson correlation analysis.

```
In [35]:
units_num_items_fig = px.scatter(data_frame=units_num_items, x='units', y='avg purchases')
units_num_items_fig.show()
ua_ccoef = np.corrcoef(x=units_num_items['units'], y=units_num_items['avg purchases'])[1,0]
print(f"Pearson Correlation Coefficient: {ua_ccoef}")
```




```
Pearson Correlation Coefficient: 0.14532892862498967
```

We perform Pearson correlation analysis which takes in the units as `x` and average purchases as `y` and in order to test whether there is a correlation. The Pearson correlation analysis shows that `0.18245918236827244`. We can conclude that there is almost no correlation between the two.

Hypothesis 2: There is no correlation between the number of units a student is taking and the types of items they purchase.

We will use chi-square test.

```
In [36]:
units_items_ct = data[["units"]].join(indv_items.astype(int)).fillna(0).groupby("units").sum()
chi2, p, df, expected = chi2_contingency(units_items_ct)
chi2, df
```

```
Out[36]: (82.19406709104699, 77)
```

Null Hypothesis: There is no correlation between the number of units a student is taking and the types of items they purchase.

Results: The chi-square value was `82.19406709104699` and with degrees of freedom of `77` the table shows that in order to reject null hypothesis, we need a chi-square value of `sig level = 0.05`, which is `98.484`. Therefore, our chi-square is less than `98.484` so our chi-square test shows that we should **fail to reject** the null hypothesis.

Hypothesis 3: A student's living condition (where they live + dining plan) does not affect the type of items they purchase.

We will use chi-square test.

```
In [37]:
live_items_ct = data[["live situation"]].join(indv_items.astype(int)).fillna(0).groupby("live situation").sum()
chi2, p, df, expected = chi2_contingency(live_items_ct)
chi2, df
```

```
Out[37]: (29.451571862592957, 21)
```

Null Hypothesis: A student's living condition (where they live + dining plan) does not affect the type of items they purchase.

The chi-square value was `29.451571862592957` and with degrees of freedom of `21` the table shows that in order to reject null hypothesis, we need a chi-square value of `sig level = 0.05`, which is `32.671`. Therefore, our chi-square is less than 32.671 so our chi-square test shows that we should **fail to reject** the null hypothesis.

Pearson Correlation Coef:

x = units y = avg purchases

```
units_num_items.head(3)
```

	units	avg purchases
0	16.0	1.0
1	12.0	0.0
2	16.0	0.5

$$r = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum(x_i-\bar{x})^2\sum(y_i-\bar{y})^2}}$$

$\bar{x} = 14.667$

$\bar{y} = .5$

$16 - 14.667 = 1.333$

$$\frac{\left((16-14.667)(1-.5)\right)+\left((12-14.667)(0-.5)\right)+\left((16-14.667)(.5-.5)\right)}{\sqrt{\left((16-14.667)^2+(12-14.667)^2+(16-14.667)^2\right)\cdot\left((1-.5)^2+(0-.5)^2+(.5-.5)^2\right)}}$$

$$\frac{\left((1.333)(.5)\right)+\left((-2.667)(-.5)\right)+\cancel{\left((1.333)(0)\right)}}{\sqrt{\left((1.333)^2+(-2.667)^2+(1.333)^2\right)\cdot\left((.5)^2+(-.5)^2+\cancel{(0)^2}\right)}}$$

$$\frac{.6665 + 1.3335}{\sqrt{(10.667)(.5)}}$$

$$\frac{2}{2.309} = \boxed{.866025}$$

```
In [30]: ua_ccoef = np.corrcoef(x=units_num_items['units'].head(3), y=units_num_items['avg purchases'].head(3))[1,0]
         print(f"Pearson Correlation Coefficient: {ua_ccoef}")

         Pearson Correlation Coefficient: 0.8660254037844385
```

units	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water	Total
3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
8.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
12.0	2.0	0.0	19.0	2.0	9.0	8.0	3.0	6.0	49
Total:	3	0	19	2	9	8	3	6	

	none	Dessert	Entrees	Fruit	Juice	Snacks	Soda	Water
3.0	$\frac{0-0}{0}$	0	0	0	0	0	0	0
8.0	$\frac{(1-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$	$\frac{(0-.125)^2}{.125}$
12.0	$\frac{(2-6.125)^2}{6.125}$	$\frac{(6-6.125)^2}{6.125}$	$\frac{(19-6.125)^2}{6.125}$	$\frac{(2-6.125)^2}{6.125}$	$\frac{(9-6.125)^2}{6.125}$	$\frac{(8-6.125)^2}{6.125}$	$\frac{(3-6.125)^2}{6.125}$	$\frac{(6-6.125)^2}{6.125}$

$$6.125 + 8(.125) + 2.778 + .625 + 27.063 + 6.125 + 1.349 + .5739 + 1.594 + .002$$

$$\boxed{\chi^2 = 47.24}$$

live situation	None	Desserts	Entrees	Fruits/Salads	Juice/Tea/Coffee	Snacks	Soda	Water	Total
Off-Campus (3-mile radius)	4	3	23	6	11	13	3	8	71
Off-Campus (31+ -mile radius)	2	0	3	0	0	0	0	0	5
Total:	6	3	26	6	11	13	3	8	

	none	Dessert	Entrees	Fruit	Juice	Snacks	Soda	Water
3mile	$\frac{(4-8.875)^2}{8.875}$	$\frac{(3-8.875)^2}{8.875}$	$\frac{(23-8.875)^2}{8.875}$	$\frac{(6-8.875)^2}{8.875}$	$\frac{(11-8.875)^2}{8.875}$	$\frac{(13-8.875)^2}{8.875}$	$\frac{(3-8.875)^2}{8.875}$	$\frac{(8-8.875)^2}{8.875}$
31+ mile	$\frac{(2-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$	$\frac{(3-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$	$\frac{(0-.625)^2}{.625}$

$6(.625) + .3025 + 2.677 + 3.889 + 22.48 + .931 +$
$.508 + 1.917 + 3.889 + .086$

$\chi^2 = 40.4297$