Professor McDonald
FNCE 5352 – Financial Programming and Modeling
March 13, 2019
SQL Practice Assignment

Using the following link that allows you to execute SQL statements against an imaginary SQL database:

https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all

Answer the following questions:

1) Are there any customers that have never placed an order?

```
select c.*
from customers c left join orders o on c.customerid = o.customerid
where o.customerid is null;
```

Number of Records: 17

CustomerID	CustomerName	ContactName
1	Alfreds Futterkiste	Maria Anders
6	Blauer See Delikatessen	Hanna Moos
12	Cactus Comidas para llevar	Patricio Simpson
22	FISSA Fabrica Inter. Salchichas S.A.	Diego Roel
26	France restauration	Carine Schmitt
32	Great Lakes Food Market	Howard Snyder
40	La corne d'abondance	Daniel Tonini
42	Laughing Bacchus Wine Cellars	Yoshi Tannamuri

2) What does the distribution of revenues from orders look like?

```
select o.orderid, sum(p.price * od.quantity) revenue
from orders o
    join orderdetails od on o.orderid = od.orderid
     join products p on od.productid = p.productid
group by o.orderid
order by revenue desc
```
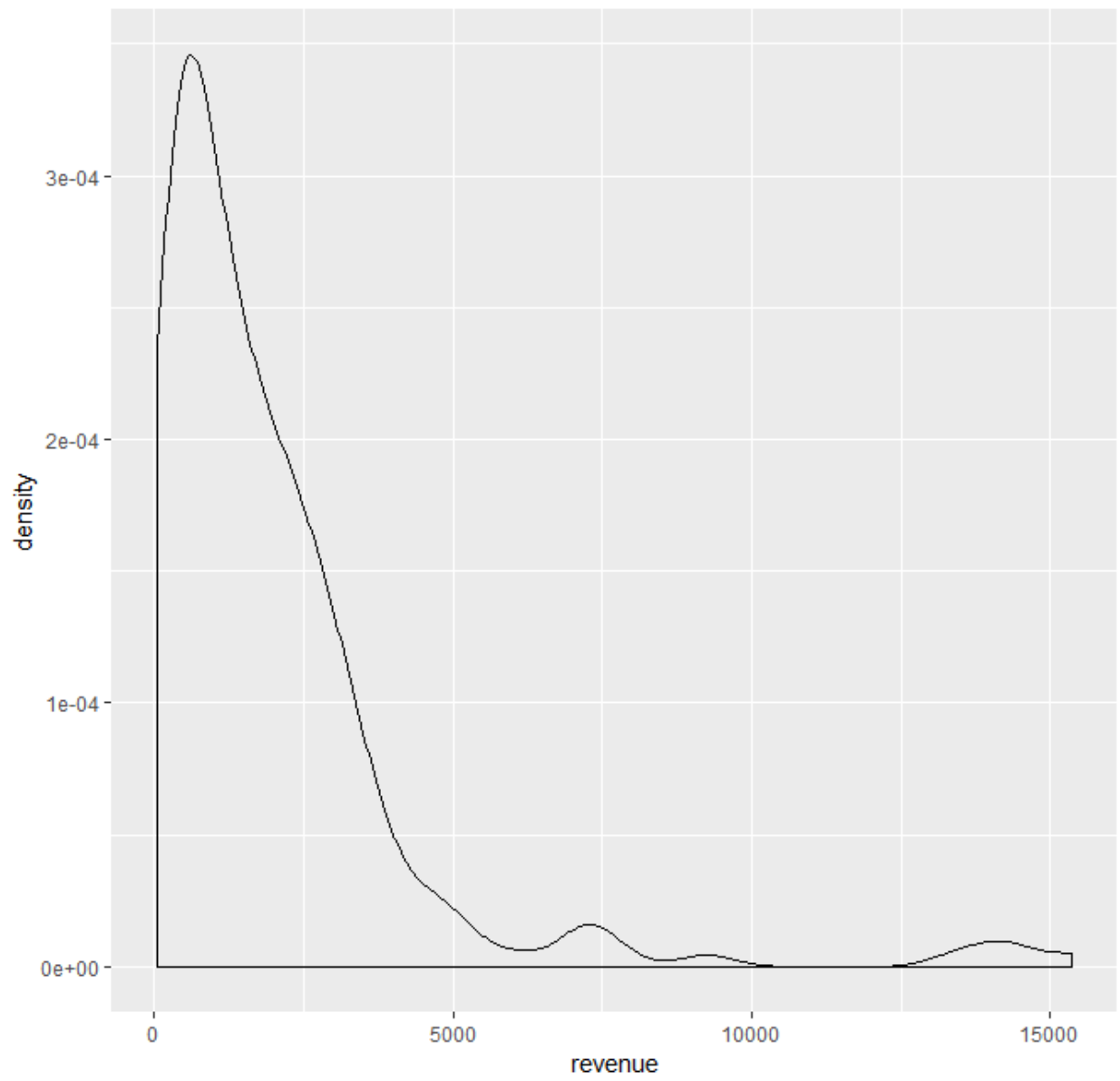
Number of Records: 196

OrderID	revenue
10372	15353.6
10424	14366.5
10417	14104
10353	13427
10360	9244.250000000002
10324	7698.45
10440	7246.01
10430	7245
10351	7103.599999999999

We can save this data out to a .csv file, import it into R and plot a density

```
> library(tidyverse)
> x <- read_csv('order_revenues.csv')
Parsed with column specification:
cols(
  OrderID = col_double(),
  revenue = col_double()
)
> ggplot(x, aes(revenue)) + geom_density() + ggtitle('distribution of revenues')
```

distribution of revenues

3) How many orders do we have?

```
select count(*) from orders
```

Result:

Number of Records: 1

count(*)
196

4) What are our top 10 customers (by revenue)?

```
select c.customerid, c.customername, sum(p.price * od.quantity) revenue
from customers c
    join orders o on c.customerid = o.customerid
    join orderdetails od on o.orderid = od.orderid
     join products p on od.productid = p.productid
group by 1, 2
order by revenue desc
```

Result:

Number of Records: 74

CustomerID	CustomerName	revenue
20	Ernst Handel	35631.21000000001
51	Mère Paillarde	23362.600000000002
71	Save-a-lot Markets	22500.06
65	Rattlesnake Canyon Grocery	18421.42
63	QUICK-Stop	18178.8
62	Queen Cozinha	17880.6
59	Piccolo und mehr	16040.75
37	Hungry Owl All-Night Grocers	15301.03