SQL DATA ANALYSIS PROJECT

Samantha Watson

January 2024

The following project was completed using PostgresSQL and the uncleaned “Data Science Job Posting on Glassdoor” dataset on Kaggle and involves exploratory data analysis and data cleaning/wrangling. The dataset can be found at the following link: https://www.kaggle.com/datasets/rashikrahmanpritom/data-science-job-posting-on-glassdoor?select=Uncleaned_DS_jobs.csv

Please note that due to the size and number of records in this data set, data output shown has been limited to a certain number of records (i.e. first ten records).

EXPLORATORY DATA ANALYSIS

1) Create a database to allow for table creation and data import.

```
CREATE DATABASE projects_2024;
```

2) Create table to import data into.

```
CREATE TABLE ds_salaries(
        index int PRIMARY KEY,
        job_title text,
        salary_estimate text,
        job_description text,
        rating numeric,
        company_name text,
        location text,
        headquarters text,
        size text,
        founded int,
        ownership text,
        industry text,
        sector text,
        revenue text,
        competitors text
);
```

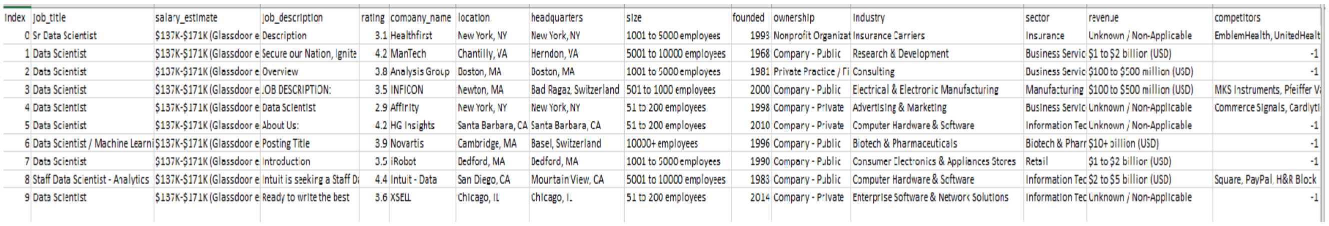
3) Import csv file.

```
COPY ds_salaries
FROM 'C:\Users\Public\ds_salaries_project.csv'
```

```
WITH(FORMAT CSV, HEADER);
```

4) View output to verify import and display table.

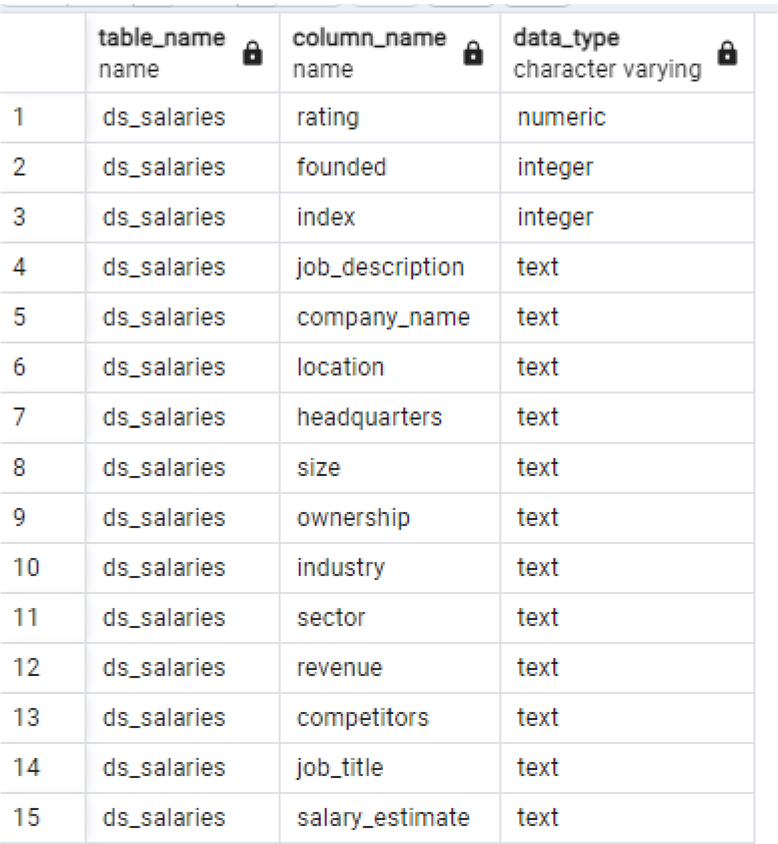
```
SELECT * FROM ds_salaries
LIMIT 10;
```



index	job_title	salary_estimate	job_description	rating	company_name	location	headquarters	size	founded	ownership	industry	sector	revenue	competitors
0	Sr Data Scientist	$137K-$171K (Glassdoor e	Description	3.1	Healthfirst	New York, NY	New York, NY	1001 to 5000 employees	1993	Nonprofit Organizat	Insurance Carriers	Insurance	Unknown / Non-Applicable	EmblemHealth, UnitedHealt
1	Data Scientist	$137K-$171K (Glassdoor e	Secure our Nation, Ignite	4.2	ManTech	Chantilly, VA	Herndon, VA	5001 to 10000 employees	1968	Company - Public	Research & Development	Business Servic	$1 to $2 billion (USD)	-1
2	Data Scientist	$137K-$171K (Glassdoor e	Overview	3.8	Analysis Group	Boston, MA	Boston, MA	1001 to 5000 employees	1981	Private Practice / Fi	Consulting	Business Servic	$100 to $500 million (USD)	-1
3	Data Scientist	$137K-$171K (Glassdoor e	JOB DESCRIPTION:	3.5	INFICON	Newton, MA	Bad Ragaz, Switzerland	501 to 1000 employees	2000	Company - Public	Electrical & Electronic Manufacturing	Manufacturing	$100 to $500 million (USD)	MKS Instruments, Pfeiffer Va
4	Data Scientist	$137K-$171K (Glassdoor e	Data Scientist	2.9	Affinity	New York, NY	New York, NY	51 to 200 employees	1998	Company - Private	Advertising & Marketing	Business Servic	Unknown / Non-Applicable	Commerce Signals, Cardlyti
5	Data Scientist	$137K-$171K (Glassdoor e	About Us:	4.2	HG Insights	Santa Barbara, CA	Santa Barbara, CA	51 to 200 employees	2010	Company - Private	Computer Hardware & Software	Information Tec	Unknown / Non-Applicable	-1
6	Data Scientist / Machine Learni	$137K-$171K (Glassdoor e	Posting Title	3.9	Novartis	Cambridge, MA	Basel, Switzerland	10000+ employees	1996	Company - Public	Biotech & Pharmaceuticals	Biotech & Pharr	$10+ billion (USD)	-1
7	Data Scientist	$137K-$171K (Glassdoor e	Introduction	3.5	iRobot	Bedford, MA	Bedford, MA	1001 to 5000 employees	1990	Company - Public	Consumer Electronics & Appliances Stores	Retail	$1 to $2 billion (USD)	-1
8	Staff Data Scientist - Analytics	$137K-$171K (Glassdoor e	Intuit is seeking a Staff Da	4.4	Intuit - Data	San Diego, CA	Mountain View, CA	5001 to 10000 employees	1983	Company - Public	Computer Hardware & Software	Information Tec	$2 to $5 billion (USD)	Square, PayPal, H&R Block
9	Data Scientist	$137K-$171K (Glassdoor e	Ready to write the best	3.6	XSELL	Chicago, IL	Chicago, IL	51 to 200 employees	2014	Company - Private	Enterprise Software & Network Solutions	Information Tec	Unknown / Non-Applicable	-1

5) Verify column datatypes.

```
SELECT table_name, column_name, data_type
FROM information_schema.columns
WHERE table_name = 'ds_salaries';
```



	table_name name	column_name name	data_type character varying
1	ds_salaries	rating	numeric
2	ds_salaries	founded	integer
3	ds_salaries	index	integer
4	ds_salaries	job_description	text
5	ds_salaries	company_name	text
6	ds_salaries	location	text
7	ds_salaries	headquarters	text
8	ds_salaries	size	text
9	ds_salaries	ownership	text
10	ds_salaries	industry	text
11	ds_salaries	sector	text
12	ds_salaries	revenue	text
13	ds_salaries	competitors	text
14	ds_salaries	job_title	text
15	ds_salaries	salary_estimate	text

6) Create a backup table now that data import and accurate format has been verified.

```
CREATE TABLE ds_salaries_backup AS
    SELECT * FROM ds_salaries;
```

7) Count the number of records in the dataset.

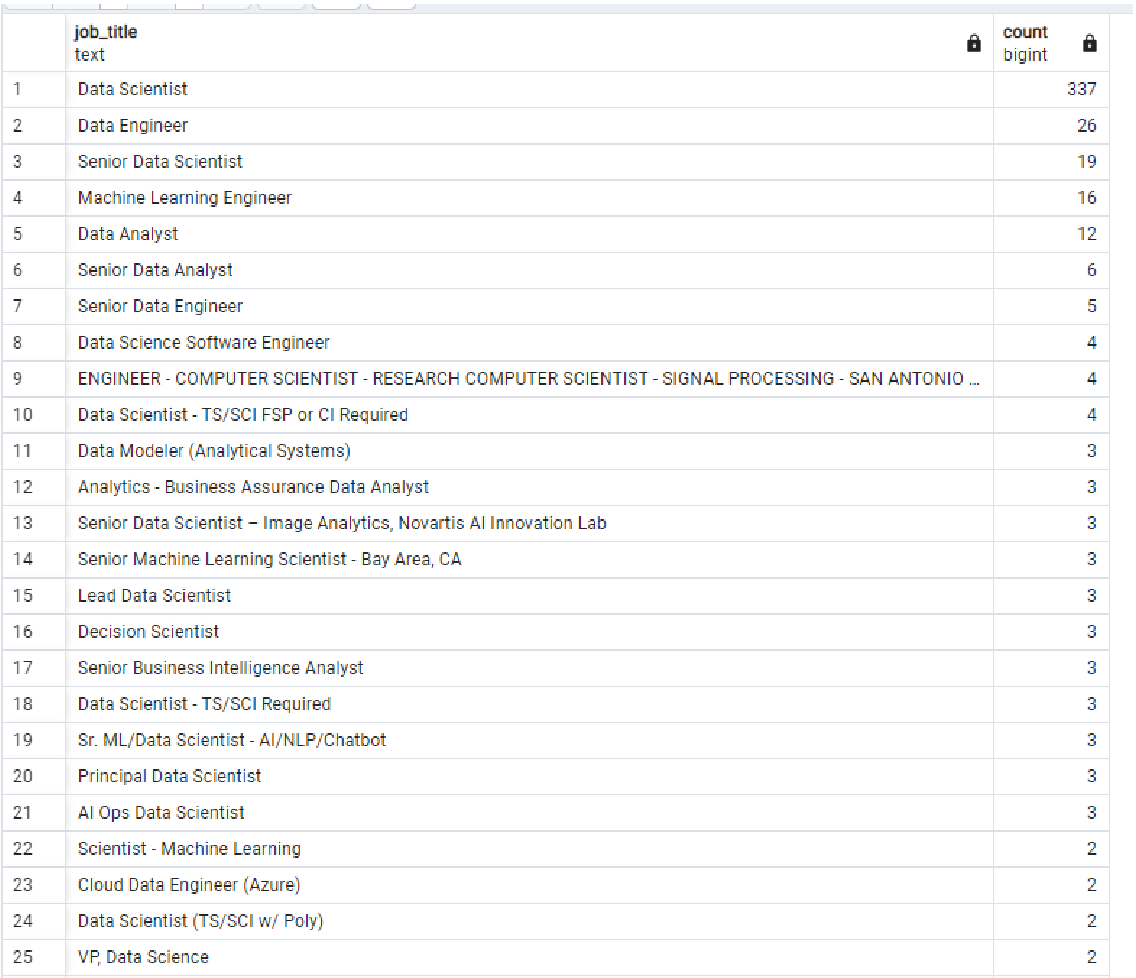
```
SELECT COUNT(*) FROM ds_salaries;
```

8) Perform a quick data inspection of the head and tail of dataset.

```
SELECT * FROM ds_salaries
ORDER BY index ASC
LIMIT 5;
SELECT * FROM ds_salaries
ORDER BY index DESC
LIMIT 5;
```

9) Retrieve counts of various job titles and possible spelling/format variations of similar job titles.

```
SELECT job_title, COUNT(job_title) FROM ds_salaries
GROUP BY job_title
ORDER BY COUNT(job_title) DESC;
```



	job_title text	count bigint
1	Data Scientist	337
2	Data Engineer	26
3	Senior Data Scientist	19
4	Machine Learning Engineer	16
5	Data Analyst	12
6	Senior Data Analyst	6
7	Senior Data Engineer	5
8	Data Science Software Engineer	4
9	ENGINEER - COMPUTER SCIENTIST - RESEARCH COMPUTER SCIENTIST - SIGNAL PROCESSING - SAN ANTONIO ...	4
10	Data Scientist - TS/SCI FSP or CI Required	4
11	Data Modeler (Analytical Systems)	3
12	Analytics - Business Assurance Data Analyst	3
13	Senior Data Scientist – Image Analytics, Novartis AI Innovation Lab	3
14	Senior Machine Learning Scientist - Bay Area, CA	3
15	Lead Data Scientist	3
16	Decision Scientist	3
17	Senior Business Intelligence Analyst	3
18	Data Scientist - TS/SCI Required	3
19	Sr. ML/Data Scientist - AI/NLP/Chatbot	3
20	Principal Data Scientist	3
21	AI Ops Data Scientist	3
22	Scientist - Machine Learning	2
23	Cloud Data Engineer (Azure)	2
24	Data Scientist (TS/SCI w/ Poly)	2
25	VP, Data Science	2

***Observations: There are many variations of the same job title. For example, “Senior Data Scientist” vs. “Sr. Data Scientist.”**

10) Investigation and exploration (code shown below) of the remaining text columns reveal similar formatting issues. Additional observations include:

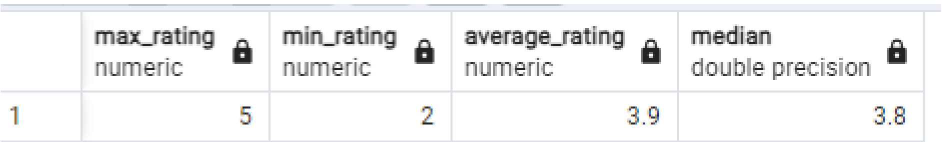
- **The need to change the salary_estimate column to a numerical format so that mathematical and aggregate calculations may be performed. The salary_estimate column has text that needs to be removed and the salary range needs to be split into a lower range and upper range.**
- **It appears that NULL values are coded as a "-1" in the following columns: size, ownership, industry, sector, revenue, and competitors columns, as "Unknown" in the ownership and size columns, and as "Unknown/Non-Applicable" in the revenue column. These could be converted to NULL to aid in later analysis.**
- **Both city and state are listed in the location column, this can be split into separate columns to aid in later analysis.**

```
SELECT salary_estimate, COUNT(salary_estimate) FROM ds_salaries
GROUP BY salary_estimate
ORDER BY COUNT(salary_estimate) DESC;


SELECT company_name, COUNT(company_name) FROM ds_salaries
GROUP BY company_name
ORDER BY company_name DESC;


SELECT ownership, COUNT(ownership) FROM ds_salaries
GROUP BY ownership
ORDER BY ownership DESC;


SELECT location, COUNT(location) FROM ds_salaries
GROUP BY location
ORDER BY location DESC;


SELECT size, COUNT(size) FROM ds_salaries
GROUP BY size
ORDER BY COUNT(size) DESC;


SELECT industry, COUNT(industry) FROM ds_salaries
GROUP BY industry
ORDER BY COUNT(industry) DESC;
```

```
SELECT sector, COUNT(sector) FROM ds_salaries
GROUP BY sector
ORDER BY COUNT(sector) DESC;


SELECT revenue, COUNT(revenue) FROM ds_salaries
GROUP BY revenue
ORDER BY COUNT(revenue) DESC;


SELECT competitors, COUNT(competitors) FROM ds_salaries
GROUP BY competitors
ORDER BY competitors;
```

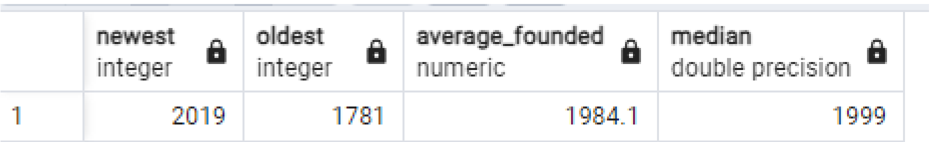
11) Explore min/max, mean and median of rating and founded columns (excluding NULLS).

```
SELECT MAX(rating) AS max_rating, MIN(rating) AS min_rating, ROUND(AVG(rating), 1) AS average_rating,
        PERCENTILE_CONT(.5) WITHIN GROUP(ORDER BY rating) AS median  FROM ds_salaries
        WHERE rating<>-1;
```



	max_rating numeric	min_rating numeric	average_rating numeric	median double precision
1	5	2	3.9	3.8

```
SELECT MAX(founded) AS newest, MIN(founded) AS oldest, ROUND(AVG(founded), 1) AS average_founded,
        PERCENTILE_CONT(.5) WITHIN GROUP(ORDER BY founded) AS median  FROM ds_salaries
        WHERE founded<>-1;
```



	newest integer	oldest integer	average_founded numeric	median double precision
1	2019	1781	1984.1	1999

***Observation: Founding date of 1781 is questionable; this may skew average.**

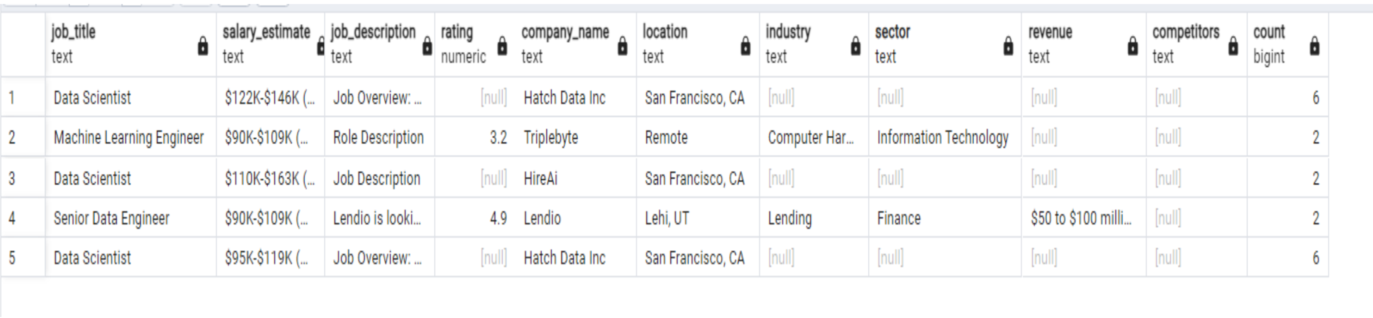
DATA CLEANING & WRANGLING

1) After investigation of each column, it appears that NULL values are coded as "-1", "Unknown", or "Unknown / Non-Applicable" and will need to be updated for consistency purposes.

```
START TRANSACTION;

UPDATE ds_salaries
SET rating = NULL
WHERE rating = -1;

UPDATE ds_salaries
SET headquarters = NULL
WHERE headquarters = '-1';

UPDATE ds_salaries
SET size = NULL
WHERE size = '-1' OR size = 'Unknown';

UPDATE ds_salaries
SET competitors = NULL
WHERE competitors = '-1';

UPDATE ds_salaries
SET founded = NULL
WHERE founded = -1;

UPDATE ds_salaries
SET industry = NULL
WHERE industry = '-1';

UPDATE ds_salaries
SET sector = NULL
WHERE sector = '-1';

UPDATE ds_salaries
SET ownership = NULL
WHERE ownership = '-1' OR ownership = 'Unknown';

UPDATE ds_salaries
SET revenue = NULL
WHERE revenue = '-1'

UPDATE ds_salaries
SET revenue = NULL
WHERE revenue ILIKE '%unknown%';

COMMIT;
```

2) Find duplicate records.

```
SELECT job_title, salary_estimate, job_description, rating, company_name, location, industry, sector,
    revenue, competitors, COUNT(*)  FROM ds_salaries
GROUP BY job_title, salary_estimate, job_description, rating, company_name, location, industry, sector,
    revenue, competitors
HAVING COUNT(location_state) >1;
```



	job_title text	salary_estimate text	job_description text	rating numeric	company_name text	location text	industry text	sector text	revenue text	competitors text	count bigint
1	Data Scientist	$122K-$146K (...	Job Overview: ...	[null]	Hatch Data Inc	San Francisco, CA	[null]	[null]	[null]	[null]	6
2	Machine Learning Engineer	$90K-$109K (...	Role Description	3.2	Triplebyte	Remote	Computer Har...	Information Technology	[null]	[null]	2
3	Data Scientist	$110K-$163K (...	Job Description	[null]	HireAi	San Francisco, CA	[null]	[null]	[null]	[null]	2
4	Senior Data Engineer	$90K-$109K (...	Lendio is looki...	4.9	Lendio	Lehi, UT	Lending	Finance	$50 to $100 milli...	[null]	2
5	Data Scientist	$95K-$119K (...	Job Overview: ...	[null]	Hatch Data Inc	San Francisco, CA	[null]	[null]	[null]	[null]	6

*** It would be noteworthy to know whether these duplicate records are erroneous or are due to multiple job postings due to the actual number of positions available. Therefore, for the purposes of this project, duplicate records will be kept.**

3) The salary_estimate column needs to be:

- **Split into two columns, low range and high range columns containing the lower range of the salary and higher range of salary**
- **The low and high range columns need to be rid of text values (i.e. the "Glassdoor estimate" at the end of each salary range)**
- **The low and high range columns need to be converted to numeric format to allow for further analysis**

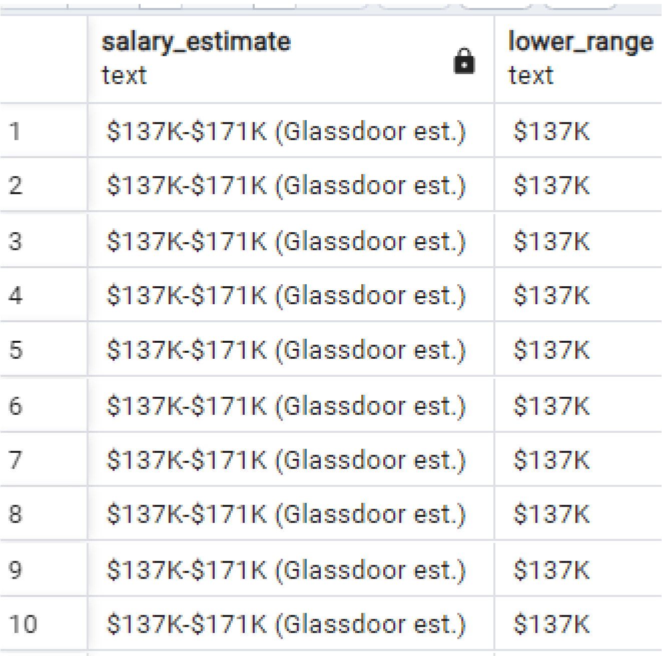
```
SELECT DISTINCT salary_estimate FROM ds_salaries
ORDER BY salary_estimate;
```

Add lower_range and higher_range columns, will originate as text data type and convert to numeric later.

```
ALTER TABLE ds_salaries
ADD COLUMN lower_range text;

ALTER TABLE ds_salaries
ADD COLUMN higher_range text;
```

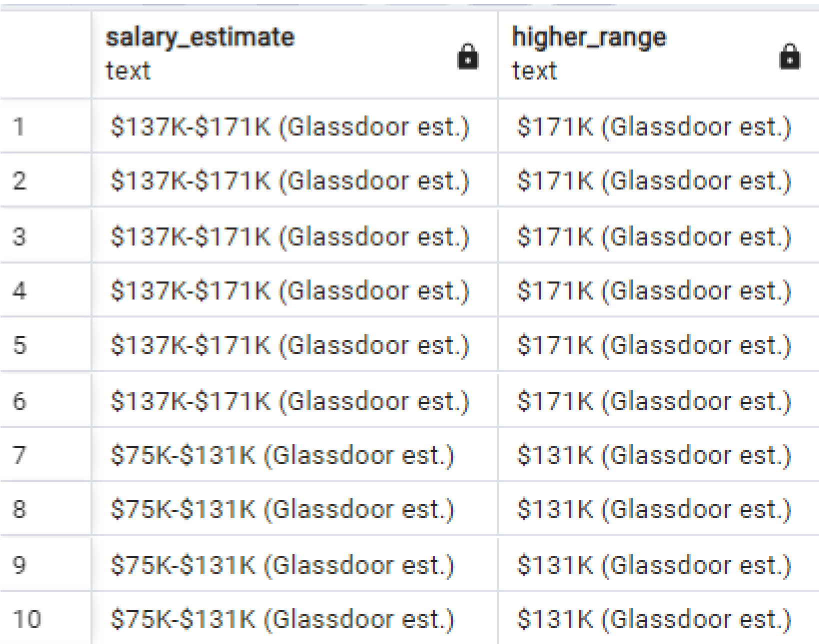
Split salary_estimate column into ranges using "-" as a delimiter.

```
START TRANSACTION;

UPDATE ds_salaries
SET lower_range = split_part(salary_estimate, '-', 1)
RETURNING salary_estimate, lower_range;
```



	salary_estimate text	lower_range text
1	$137K-$171K (Glassdoor est.)	$137K
2	$137K-$171K (Glassdoor est.)	$137K
3	$137K-$171K (Glassdoor est.)	$137K
4	$137K-$171K (Glassdoor est.)	$137K
5	$137K-$171K (Glassdoor est.)	$137K
6	$137K-$171K (Glassdoor est.)	$137K
7	$137K-$171K (Glassdoor est.)	$137K
8	$137K-$171K (Glassdoor est.)	$137K
9	$137K-$171K (Glassdoor est.)	$137K
10	$137K-$171K (Glassdoor est.)	$137K

```
UPDATE ds_salaries
SET higher_range = split_part(salary_estimate, '-', 2)
RETURNING salary_estimate, higher_range;
```



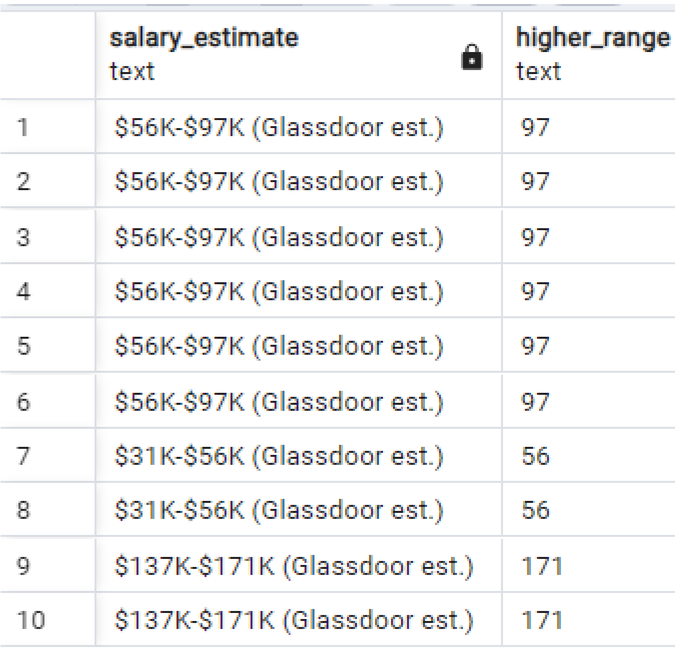
	salary_estimate text	higher_range text
1	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
2	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
3	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
4	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
5	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
6	$137K-$171K (Glassdoor est.)	$171K (Glassdoor est.)
7	$75K-$131K (Glassdoor est.)	$131K (Glassdoor est.)
8	$75K-$131K (Glassdoor est.)	$131K (Glassdoor est.)
9	$75K-$131K (Glassdoor est.)	$131K (Glassdoor est.)
10	$75K-$131K (Glassdoor est.)	$131K (Glassdoor est.)

Remove text from end of higher_range column.

```
UPDATE ds_salaries
SET higher_range = SUBSTRING(higher_range,2,3)
RETURNING salary_estimate, higher_range;
```

Remove "K" from end of higher_range values <100K.

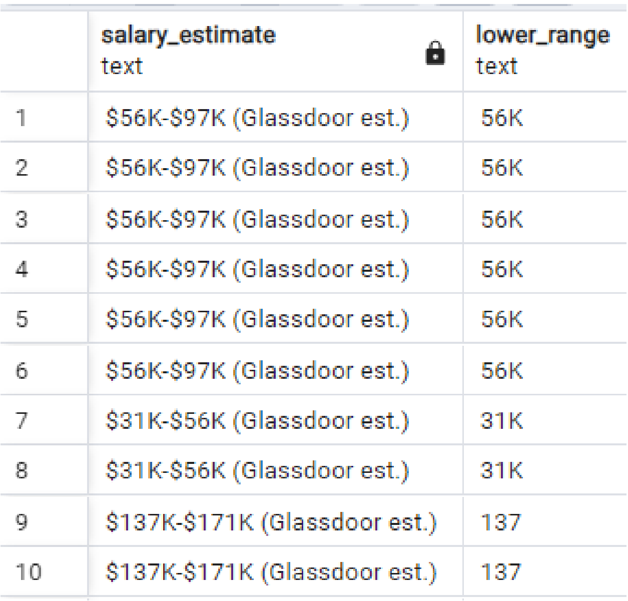
```
UPDATE ds_salaries
SET higher_range = LEFT(higher_range,2)
WHERE POSITION ('K' IN higher_range)>0
RETURNING salary_estimate, higher_range;
```



	salary_estimate text	higher_range text
1	$56K-$97K (Glassdoor est.)	97
2	$56K-$97K (Glassdoor est.)	97
3	$56K-$97K (Glassdoor est.)	97
4	$56K-$97K (Glassdoor est.)	97
5	$56K-$97K (Glassdoor est.)	97
6	$56K-$97K (Glassdoor est.)	97
7	$31K-$56K (Glassdoor est.)	56
8	$31K-$56K (Glassdoor est.)	56
9	$137K-$171K (Glassdoor est.)	171
10	$137K-$171K (Glassdoor est.)	171

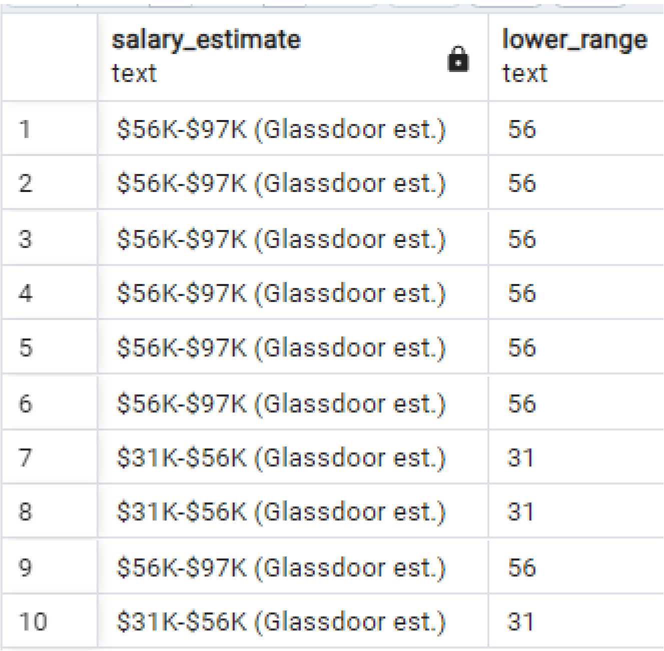
Remove "$" from beginning of lower_range values and "K" from lower_range values <100K.

```
UPDATE ds_salaries
SET lower_range = SUBSTRING(lower_range,2,3)
RETURNING salary_estimate, lower_range;
```



	salary_estimate text	lower_range text
1	$56K-$97K (Glassdoor est.)	56K
2	$56K-$97K (Glassdoor est.)	56K
3	$56K-$97K (Glassdoor est.)	56K
4	$56K-$97K (Glassdoor est.)	56K
5	$56K-$97K (Glassdoor est.)	56K
6	$56K-$97K (Glassdoor est.)	56K
7	$31K-$56K (Glassdoor est.)	31K
8	$31K-$56K (Glassdoor est.)	31K
9	$137K-$171K (Glassdoor est.)	137
10	$137K-$171K (Glassdoor est.)	137

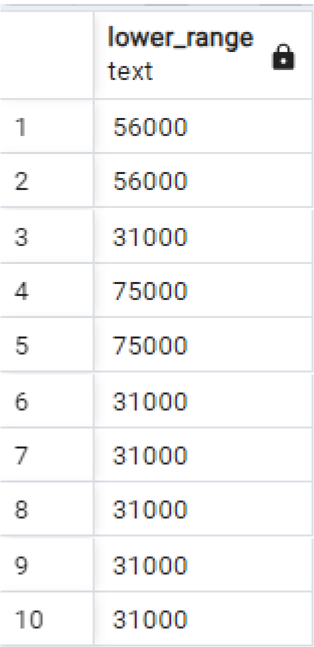
```
UPDATE ds_salaries
SET lower_range = LEFT(lower_range,2)
WHERE POSITION ('K' IN lower_range)>0
RETURNING salary_estimate, lower_range;
```



	salary_estimate text	lower_range text
1	$56K-$97K (Glassdoor est.)	56
2	$56K-$97K (Glassdoor est.)	56
3	$56K-$97K (Glassdoor est.)	56
4	$56K-$97K (Glassdoor est.)	56
5	$56K-$97K (Glassdoor est.)	56
6	$56K-$97K (Glassdoor est.)	56
7	$31K-$56K (Glassdoor est.)	31
8	$31K-$56K (Glassdoor est.)	31
9	$56K-$97K (Glassdoor est.)	56
10	$31K-$56K (Glassdoor est.)	31

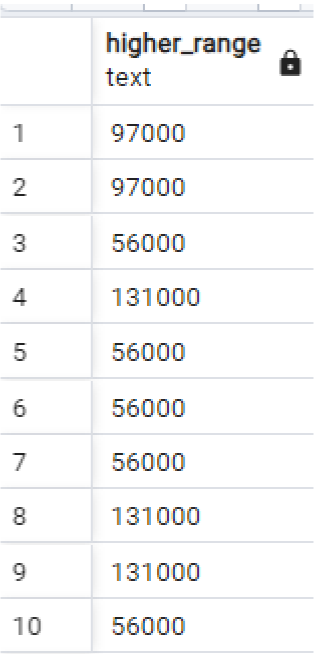
Add trailing zeros to prepare for converting data type to integer.

```
UPDATE ds_salaries
SET lower_range = lower_range||'000'
RETURNING lower_range;
```



	lower_range text
1	56000
2	56000
3	31000
4	75000
5	75000
6	31000
7	31000
8	31000
9	31000
10	31000

```
UPDATE ds_salaries
SET higher_range = higher_range||'000'
RETURNING higher_range;
```



	higher_range text
1	97000
2	97000
3	56000
4	131000
5	56000
6	56000
7	56000
8	131000
9	131000
10	56000

Change data type to integer to allow for further analysis.

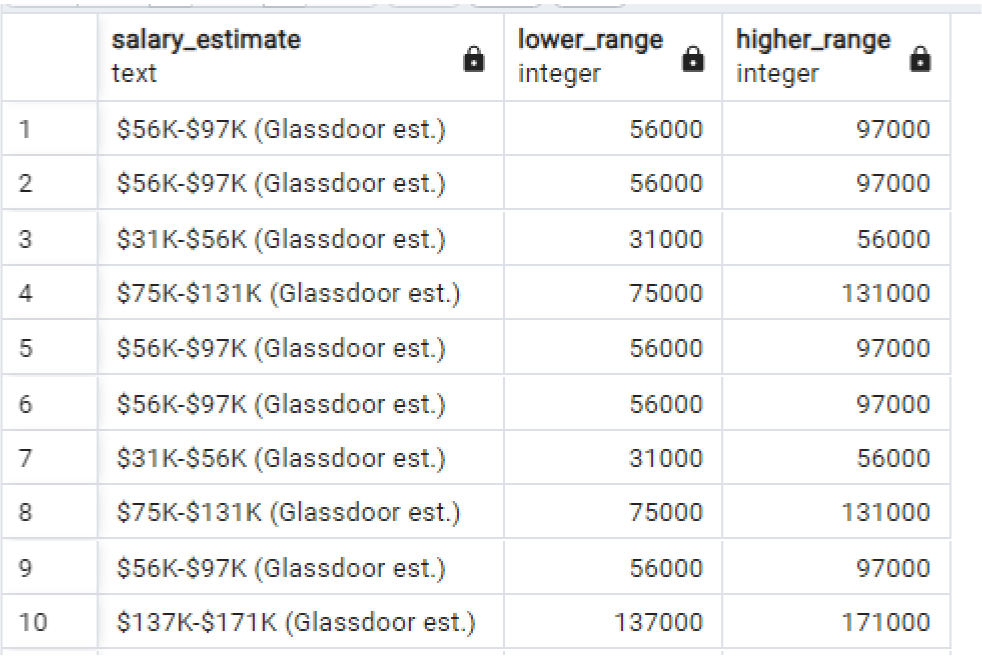
```
ALTER TABLE ds_salaries
ALTER COLUMN lower_range
SET DATA TYPE integer
USING lower_range::integer;
```

```
ALTER TABLE ds_salaries
ALTER COLUMN higher_range
SET DATA TYPE integer
USING higher_range::integer;
```

```
SELECT salary_estimate, lower_range, higher_range
FROM ds_salaries;
```

```
COMMIT;
```

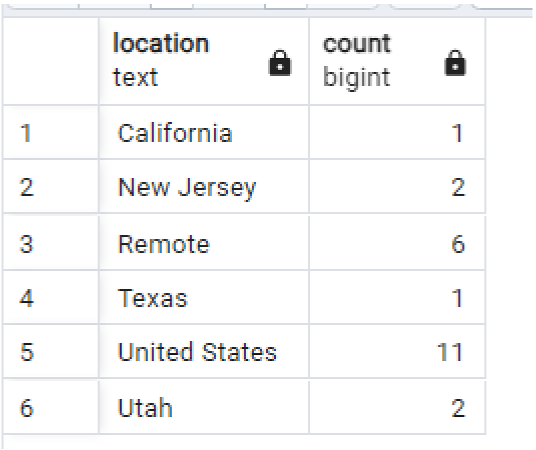
Final output shows new columns reflecting lower and higher salary ranges with no unnecessary text and converted to integer format.



	salary_estimate text	lower_range integer	higher_range integer
1	$56K-$97K (Glassdoor est.)	56000	97000
2	$56K-$97K (Glassdoor est.)	56000	97000
3	$31K-$56K (Glassdoor est.)	31000	56000
4	$75K-$131K (Glassdoor est.)	75000	131000
5	$56K-$97K (Glassdoor est.)	56000	97000
6	$56K-$97K (Glassdoor est.)	56000	97000
7	$31K-$56K (Glassdoor est.)	31000	56000
8	$75K-$131K (Glassdoor est.)	75000	131000
9	$56K-$97K (Glassdoor est.)	56000	97000
10	$137K-$171K (Glassdoor est.)	137000	171000

4) To split the location column into separate city and state columns, first check to make sure each record has both a city and state listed by checking for a "," delimiter. This allows us to see which records will not be transformed when using a delimiter to split the column.

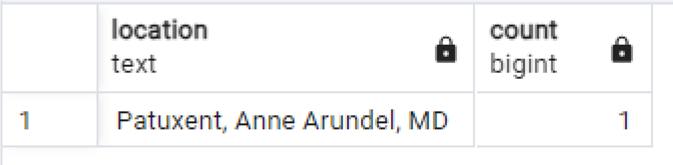
```
SELECT location, COUNT(location) FROM ds_salaries
WHERE location NOT LIKE '%,%'
GROUP BY location;
```



	location text	count bigint
1	California	1
2	New Jersey	2
3	Remote	6
4	Texas	1
5	United States	11
6	Utah	2

***Some records only list the state, country or are listed as "Remote". We can set the city to NULL for those records only listing the state, set the state as NULL for those only listing the country and have both city and state listed as "Remote" for remote jobs.**

```
SELECT location, COUNT(location) FROM ds_salaries
WHERE location ILIKE '%,%,%'
GROUP BY location;
```



	location text	count bigint
1	Patuxent, Anne Arundel, MD	1

***There is one record that contains more than one comma delimiter, it appears this record lists the city, county and state. This record will be changed to only list the city and state.**

START TRANSACTION;

Create new columns for city and state.

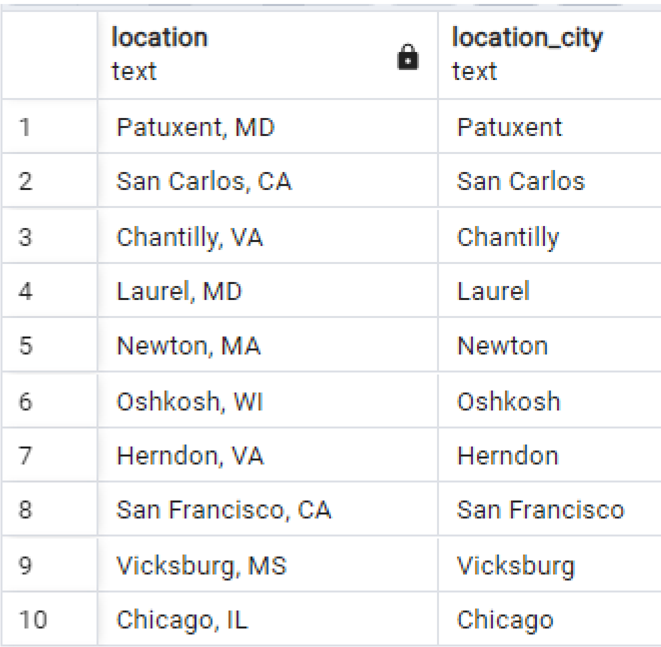
```
ALTER TABLE ds_salaries
ADD COLUMN location_city text;

ALTER TABLE ds_salaries
ADD COLUMN location_state text;
```

Update record containing two commas to reflect only city and state.

```
UPDATE ds_salaries
SET location = 'Patuxent, MD'
WHERE location = 'Patuxent, Anne Arundel, MD';
```

Extract only city from location column by using "," as a delimiter.

```
UPDATE ds_salaries
SET location_city = split_part(location, ',',1)
RETURNING location, location_city;
```



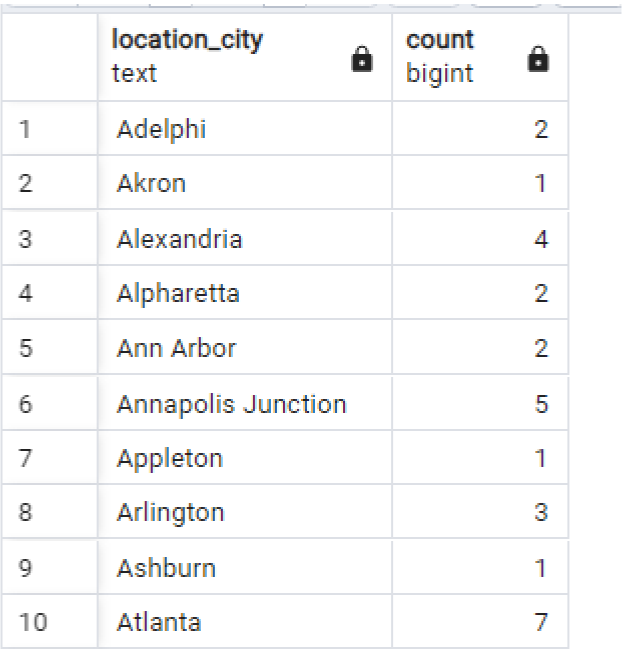
	location text	location_city text
1	Patuxent, MD	Patuxent
2	San Carlos, CA	San Carlos
3	Chantilly, VA	Chantilly
4	Laurel, MD	Laurel
5	Newton, MA	Newton
6	Oshkosh, WI	Oshkosh
7	Herndon, VA	Herndon
8	San Francisco, CA	San Francisco
9	Vicksburg, MS	Vicksburg
10	Chicago, IL	Chicago

Set city to NULL where city was not listed in location column.

```
UPDATE ds_salaries
SET location_city = NULL
WHERE location IN('California', 'New Jersey', 'Texas', 'United States', 'Utah');
```

Verify results.

```
SELECT location_city, COUNT(location_city) FROM ds_salaries
GROUP BY location_city ORDER BY location_city;
```



	location_city text	count bigint
1	Adelphi	2
2	Akron	1
3	Alexandria	4
4	Alpharetta	2
5	Ann Arbor	2
6	Annapolis Junction	5
7	Appleton	1
8	Arlington	3
9	Ashburn	1
10	Atlanta	7

Extract only state from location column.

```
UPDATE ds_salaries
SET location_state = split_part(location, ',',2)
RETURNING location, location_state;
```

Set state column to NULL where state was not listed in location column.

```
UPDATE ds_salaries
SET location_state = NULL
WHERE location IN('United States');
```

-Set state column to correct state where only state was listed with no comma delimiter.

```
UPDATE ds_salaries
SET location_state = split_part(location, ',',1)
WHERE location IN('California', 'New Jersey', 'Texas', 'Utah', 'Remote')
RETURNING location, location_state;
```

Transform full state name to two letter abbreviation.

```
UPDATE ds_salaries
SET location_state = 'CA'
WHERE location_state = 'California';
```

```
UPDATE ds_salaries
SET location_state = 'NJ'
WHERE location_state = 'New Jersey';
```

```
UPDATE ds_salaries
SET location_state = 'TX'
WHERE location_state = 'Texas';
```

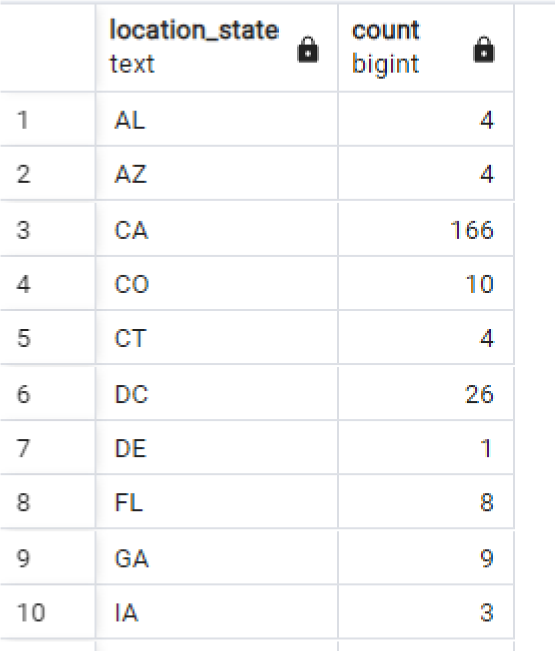
```
UPDATE ds_salaries
SET location_state = 'UT'
WHERE location_state = 'Utah';
```

Trim whitespace.

```
UPDATE ds_salaries
SET location_state = TRIM(location_state);
```

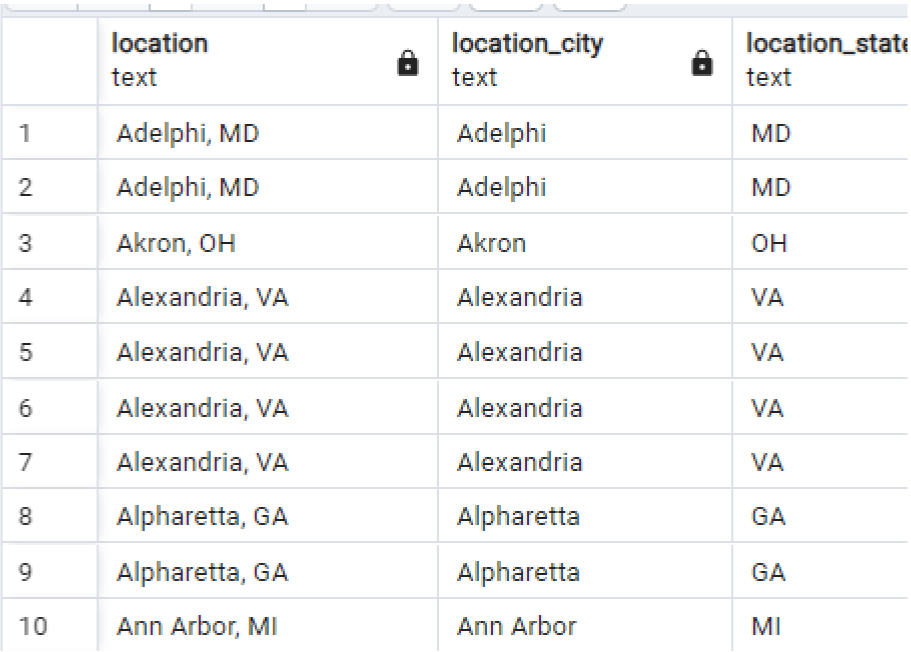
Verify results.

```
SELECT location_state, COUNT(location_state) FROM ds_salaries
GROUP BY location_state
ORDER BY location_state;
```



	location_state text	count bigint
1	AL	4
2	AZ	4
3	CA	166
4	CO	10
5	CT	4
6	DC	26
7	DE	1
8	FL	8
9	GA	9
10	IA	3

```
SELECT location, location_city, location_state FROM ds_salaries
ORDER BY location;
```



	location text	location_city text	location_state text
1	Adelphi, MD	Adelphi	MD
2	Adelphi, MD	Adelphi	MD
3	Akron, OH	Akron	OH
4	Alexandria, VA	Alexandria	VA
5	Alexandria, VA	Alexandria	VA
6	Alexandria, VA	Alexandria	VA
7	Alexandria, VA	Alexandria	VA
8	Alpharetta, GA	Alpharetta	GA
9	Alpharetta, GA	Alpharetta	GA
10	Ann Arbor, MI	Ann Arbor	MI

```
COMMIT;
```

5) Many job titles have different variations that are similar enough that they can be grouped together to make further analysis more meaningful. For example, Senior Data Scientist is also listed as Sr Data Scientist and Sr. Data Scientist. Some job titles have the company or location listed after the actual job title. This can be removed so that we are only left with the actual job title. The following steps merge variations of job titles into singular, simplified job titles.

```
START TRANSACTION;

UPDATE ds_salaries
SET job_title = 'Data Analyst'
WHERE job_title IN('Data Analyst - Unilever Prestige', 'In-Line Inspection Data Analyst', 'Data Science
    Analyst',  'Report Writer-Data Analyst', 'Data Analyst I', 'Global Data Analyst', 'Diversity and Inclusion
    Data Analyst', 'E-Commerce Data Analyst', 'Enterprise Data Analyst (Enterprise Portfolio Management
    Office',  'Operations Data Analyst', 'RFP Data Analyst');

UPDATE ds_salaries
SET job_title = 'Senior Data Analyst'
WHERE job_title IN('Health Plan Data Analyst, Sr', 'Senior Data Analyst - Finance & Platform Analytics',
     'Sr. Data Analyst', 'Sr Data Analyst');

UPDATE ds_salaries
SET job_title = 'Data Scientist'
WHERE job_title IN('Data Scientist, Kinship - NYC/Portland', 'Real World Science, Data Scientist',  'Data
    Scientist - Intermediate', 'Data Scientist - Statistics, Mid-Career',
    'Product Data Scientist - Ads Data Science', 'Data Scientist/Data Analytics Practitioner');


UPDATE ds_salaries
SET job_title = 'Senior Data Scientist'
WHERE job_title IN('Senior Data Scientist - Image Analytics, Novartis AI Innovation Lab', 'Sr Data Scientist',
     'Sr. Data Scientist', 'Sr. Data Scientist II', 'Senior Data Scientist - R&D Oncology',
    'Experienced Data Scientist', '(Sr.) Data Scientist - ', 'Senior Data Scientist - Algorithms',
    'Senior Clinical Data Scientist Programmer');

UPDATE ds_salaries
SET job_title = 'Associate Data Scientist'
WHERE job_title IN('Data Scientist - Statistics, Early Career', 'Patient Safety- Associate Data Scientist');

UPDATE ds_salaries
SET job_title = 'Data Scientist - TS/SCI Required'
WHERE job_title = 'Data Scientist (TS/SCI)';

UPDATE ds_salaries
SET job_title = 'Staff Data Scientist'
WHERE job_title IN('Staff Data Scientist - Analytics', 'Staff Data Scientist - Pricing', 'Staff Scientist-
    Upstream PD');

UPDATE ds_salaries
SET job_title = 'Data Scientist - Machine Learning'
WHERE job_title IN('Data & Machine Learning Scientist', 'Data Scientist, Applied Machine Learning - Bay
    Area', 'Scientist - Machine Learning', 'Data Scientist / Machine Learning Expert', 'Data Scientist
    Machine Learning',  'Machine Learning Scientist - Bay Area, CA');

UPDATE ds_salaries
SET job_title = 'Business Data Analyst'
WHERE job_title IN('Analytics - Business Assurance Data Analyst', 'Say Business Data Analyst');

UPDATE ds_salaries
SET job_title = 'Machine Learning Engineer'
WHERE job_title IN('Machine Learning Engineer/Scientist', 'Machine Learning Scientist / Engineer');

UPDATE ds_salaries
SET job_title = 'Senior Machine Learning Engineer'
WHERE job_title IN('Machine Learning Engineer, Sr.', 'Senior Machine Learning Scientist - Bay Area, CA');

UPDATE ds_salaries
SET job_title = 'Data Engineer'
WHERE job_title IN('Data Engineer - Kafka', 'Data Engineer (Analytics, SQL, Python, AWS)',
    'Data Engineer, Digital & Comp Pathology', 'Data Analytics Engineer',
    'Data Engineer, Enterprise Analytics', 'Tableau Data Engineer 20-0117',
    'Cloud Data Engineer (Azure)', 'Data Engineer (Remote)');

UPDATE ds_salaries
SET job_title = 'Senior Data Engineer'
WHERE job_title = 'Sr Data Engineer (Sr BI Developer)';

UPDATE ds_salaries
SET job_title = 'Computer Scientist - Engineer'
WHERE job_title IN('ENGINEER - COMPUTER SCIENTIST - RESEARCH COMPUTER SCIENTIST - SIGNAL
    PROCESSING - SAN ANTONIO OR',
    'COMPUTER SCIENTIST - ENGINEER - RESEARCH COMPUTER SCIENTIST - TRANSPORTATION
    TECHNOLOGY', 'COMPUTER SCIENTIST - ENGINEER - RESEARCH COMPUTER SCIENTIST - SIGNAL
    PROCESSING');

UPDATE ds_salaries
SET job_title = 'Data Science Manager'
WHERE job_title IN('Data Science Manager, Payment Acceptance - USA', 'Manager / Lead, Data Science &
    Analytics');

UPDATE ds_salaries
SET job_title = 'Software Engineer'
WHERE job_title IN('Software Engineer - Data Science', 'Software Engineer - Machine Learning & Data
    Science (Applied Intelligence Services Team)', 'Software Engineer (Data Scientist, C,C++,Linux, Unix) -
    SISW - MG');

UPDATE ds_salaries
SET job_title = 'Principal Data Scientist - Machine Learning'
WHERE job_title = 'Principal Machine Learning Scientist';

UPDATE ds_salaries
SET job_title = 'Senior Business Intelligence Analyst'
WHERE job_title = 'Intelligence Data Analyst, Senior';

UPDATE ds_salaries
SET job_title = 'Business Intelligence Analyst'
WHERE job_title = 'Business Intelligence Analyst I- Data Insights';

UPDATE ds_salaries
SET job_title = 'Lead Data Scientist'
WHERE job_title = 'Lead Data Scientist - Network Analysis and Control';

UPDATE ds_salaries
SET job_title = 'Data Scientist - AI'
WHERE job_title IN('AI Data Scientist', 'AI Ops Data Scientist');
```

```
UPDATE ds_salaries
SET job_title = 'Data Modeler'
WHERE job_title = 'Data Modeler (Analytical Systems)';
```

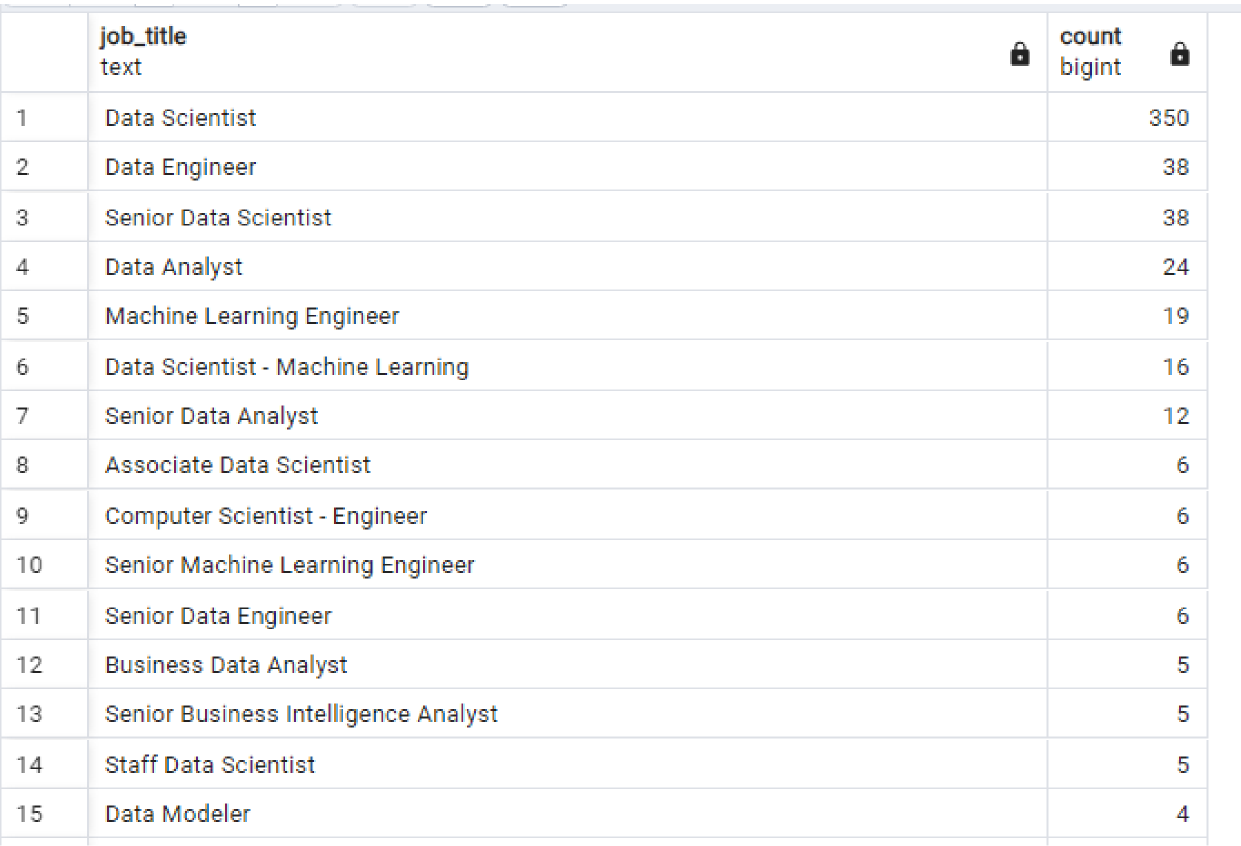
```
UPDATE ds_salaries
SET job_title = 'Computational Scientist'
WHERE job_title IN('Computational Behavioral Scientist', 'Computational Scientist, Machine Learning');
```

```
UPDATE ds_salaries
SET job_title = 'Analytics Manager'
WHERE job_title = 'Analytics Manager - Data Mart';
```

```
UPDATE ds_salaries
SET job_title = 'Lead Data Scientist'
WHERE job_title = 'Lead Data Scientist - Network Analysis and Control';
```

Verify results.

```
SELECT job_title, COUNT(job_title) FROM ds_salaries
GROUP BY job_title
ORDER BY COUNT(job_title) DESC;
```



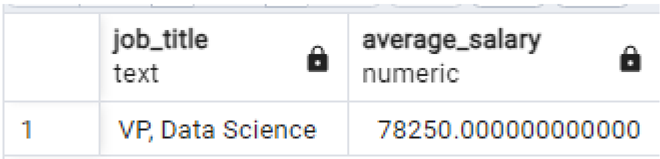
	job_title text	count bigint
1	Data Scientist	350
2	Data Engineer	38
3	Senior Data Scientist	38
4	Data Analyst	24
5	Machine Learning Engineer	19
6	Data Scientist - Machine Learning	16
7	Senior Data Analyst	12
8	Associate Data Scientist	6
9	Computer Scientist - Engineer	6
10	Senior Machine Learning Engineer	6
11	Senior Data Engineer	6
12	Business Data Analyst	5
13	Senior Business Intelligence Analyst	5
14	Staff Data Scientist	5
15	Data Modeler	4

```
COMMIT;
```

FURTHER ANALYSIS ON CLEANED DATA

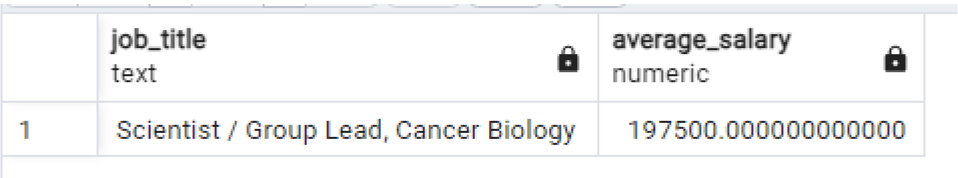
1) Which job titles (having a count greater than 1) pays the least? The most?

```
SELECT job_title, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY job_title
HAVING COUNT(job_title)>1
ORDER BY average_salary ASC
LIMIT 1;
```



	job_title text	average_salary numeric
1	VP, Data Science	78250.000000000000

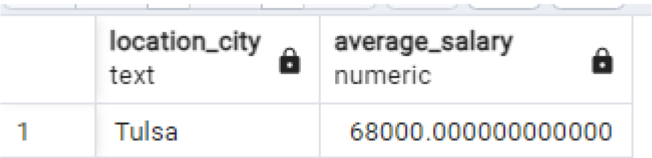
```
SELECT job_title, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY job_title
HAVING COUNT(job_title)>1
ORDER BY average_salary DESC
LIMIT 1;
```



	job_title text	average_salary numeric
1	Scientist / Group Lead, Cancer Biology	197500.000000000000

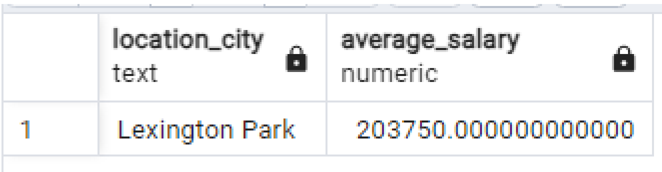
2) Which city pays the least/most? Which state pays the least/most?

```
SELECT location_city, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY location_city
HAVING COUNT(location_city)>1
ORDER BY average_salary ASC
LIMIT 1;
```



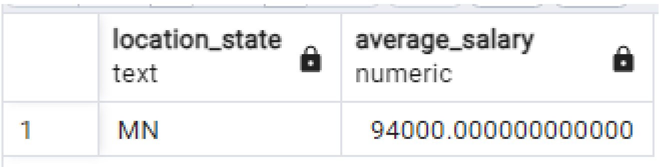
	location_city text	average_salary numeric
1	Tulsa	68000.000000000000

```
SELECT location_city, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY location_city
HAVING COUNT(location_city)>1
ORDER BY average_salary DESC
LIMIT 1;
```



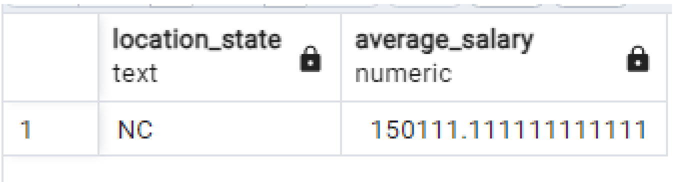
	location_city text	average_salary numeric
1	Lexington Park	203750.000000000000

```
SELECT location_state, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY location_state
HAVING COUNT(location_state)>1
ORDER BY average_salary ASC
LIMIT 1;
```



	location_state text	average_salary numeric
1	MN	94000.000000000000

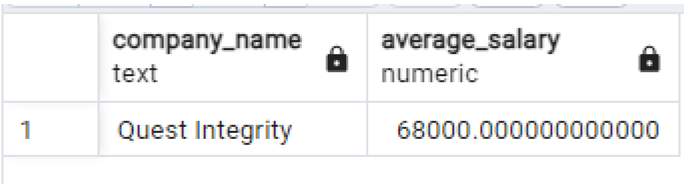
```
SELECT location_state, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY location_state
HAVING COUNT(location_state)>1
ORDER BY average_salary DESC
LIMIT 1;
```



	location_state text	average_salary numeric
1	NC	150111.111111111111

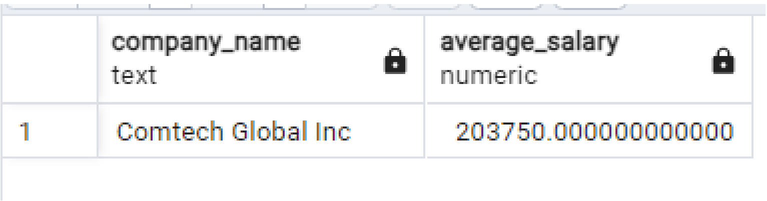
3) Which company pays the lowest/highest average salaries? Which industry pays the lowest/highest?

```
SELECT company_name, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY company_name
HAVING COUNT(company_name)>1
ORDER BY average_salary ASC
LIMIT 1;
```



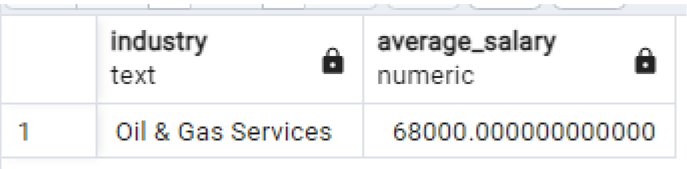
	company_name text	average_salary numeric
1	Quest Integrity	68000.000000000000

```
SELECT company_name, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY company_name
HAVING COUNT(company_name)>1
ORDER BY average_salary DESC
LIMIT 1;
```



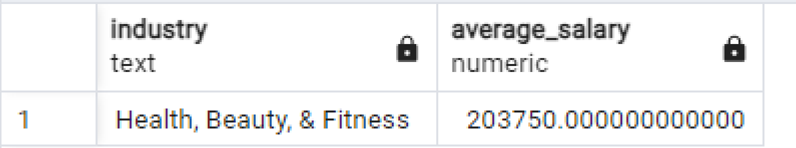
	company_name text	average_salary numeric
1	Comtech Global Inc	203750.000000000000

SELECT industry, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY industry
HAVING COUNT(industry)>1
ORDER BY average_salary ASC
LIMIT 1;



	industry text	average_salary numeric
1	Oil & Gas Services	68000.000000000000

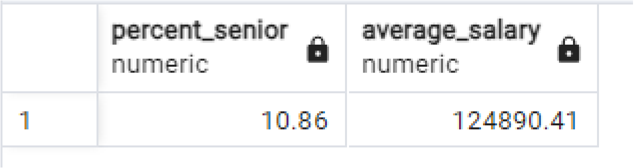
SELECT industry, AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
GROUP BY industry
HAVING COUNT(industry)>1
ORDER BY average_salary DESC
LIMIT 1;



	industry text	average_salary numeric
1	Health, Beauty, & Fitness	203750.000000000000

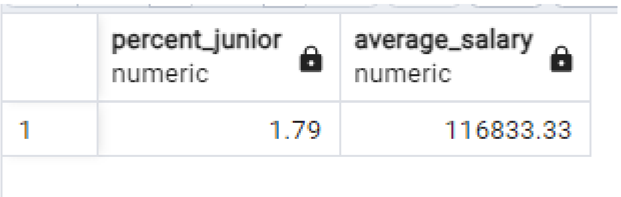
4) What percent of jobs are senior roles vs junior/associate role and what is the salary difference?

```
SELECT ROUND((SELECT SUM(number_senior) AS total_senior FROM
   (SELECT COUNT(job_title) AS number_senior FROM ds_salaries
   WHERE job_title ILIKE '%senior%'))/(SELECT COUNT(*) FROM ds_salaries)*100,2) AS percent_senior,
      (SELECT ROUND(AVG((lower_range+higher_range)/2),2) AS average_salary FROM ds_salaries
      WHERE job_title ILIKE '%senior%');
```



	percent_senior numeric	average_salary numeric
1	10.86	124890.41

```
SELECT ROUND((SELECT SUM(number_junior) AS total_junior FROM
    (SELECT COUNT(job_title) AS number_junior FROM ds_salaries
    WHERE job_title ILIKE '%jr%' OR job_title ILIKE '%associate%'))/(SELECT COUNT(*) FROM
ds_salaries)*100,2) ;
    AS percent_junior,
        (SELECT ROUND(AVG((lower_range+higher_range)/2),2) AS average_salary FROM ds_salaries
        WHERE job_title ILIKE '%jr%' OR job_title ILIKE '%associate%')
```

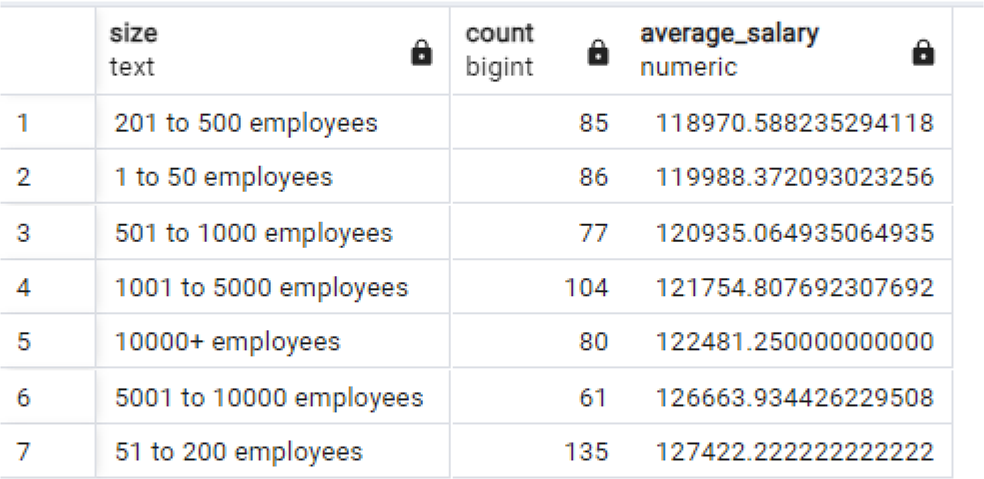


	percent_junior numeric	average_salary numeric
1	1.79	116833.33

*** Senior positions made up almost 11% of the data set, at an average salary of almost $125,000/year. While junior/associate roles made up almost 2% of the data at an average salary of almost $117,000/year.**

5) Do smaller (500 or less employees) or large companies (over 5000 employees) pay higher salaries?

```
SELECT size, COUNT(size), AVG((lower_range+higher_range)/2) AS average_salary FROM ds_salaries
WHERE size IS NOT NULL
GROUP BY size ORDER BY average_salary ASC;
```



	size text	count bigint	average_salary numeric
1	201 to 500 employees	85	118970.588235294118
2	1 to 50 employees	86	119988.372093023256
3	501 to 1000 employees	77	120935.064935064935
4	1001 to 5000 employees	104	121754.807692307692
5	10000+ employees	80	122481.250000000000
6	5001 to 10000 employees	61	126663.934426229508
7	51 to 200 employees	135	127422.222222222222

***Two of the three small size categories pay the lowest average salary, however one of the three small categories pays the highest average salary. Both large size categories are in the top three highest average salary listings.**