

1_explore

December 18, 2020

1. Exploring natality dataset

This notebook illustrates:

Exploring a BigQuery dataset using AI Platform Notebooks.

```
[ ]: !sudo chown -R jupyter:jupyter /home/jupyter/training-data-analyst
```

```
[1]: # change these to try this notebook out
      BUCKET = 'cloud-training-demos-ml'
      PROJECT = 'cloud-training-demos'
      REGION = 'us-central1'
```

```
[2]: import os
      os.environ['BUCKET'] = BUCKET
      os.environ['PROJECT'] = PROJECT
      os.environ['REGION'] = REGION
```

```
[3]: %%bash
      if ! gsutil ls | grep -q gs://${BUCKET}/; then
        gsutil mb -l ${REGION} gs://${BUCKET}
      fi
```

Explore data

The data is natality data (record of births in the US). My goal is to predict the baby's weight given a number of factors about the pregnancy and the baby's mother. Later, we will want to split the data into training and eval datasets. The hash of the year-month will be used for that – this way, twins born on the same day won't end up in different cuts of the data.

```
[4]: # Create SQL query using natality data after the year 2000
      query = """
      SELECT
        weight_pounds,
        is_male,
        mother_age,
        plurality,
        gestation_weeks,
        FARM_FINGERPRINT(CONCAT(CAST(YEAR AS STRING), CAST(month AS STRING))) AS_
        ↪hashmonth
```

```
FROM
    publicdata.samples.natality
WHERE year > 2000
"""
```

```
[5]: # Call BigQuery and examine in dataframe
from google.cloud import bigquery
df = bigquery.Client().query(query + " LIMIT 100").to_dataframe()
df.head()
```

```
[5]:   weight_pounds  is_male  mother_age  plurality  gestation_weeks  \
0         7.063611     True         32          1             37.0
1         4.687028     True         30          3             33.0
2         7.561856     True         20          1             39.0
3         7.561856     True         31          1             37.0
4         7.312733     True         32          1             40.0

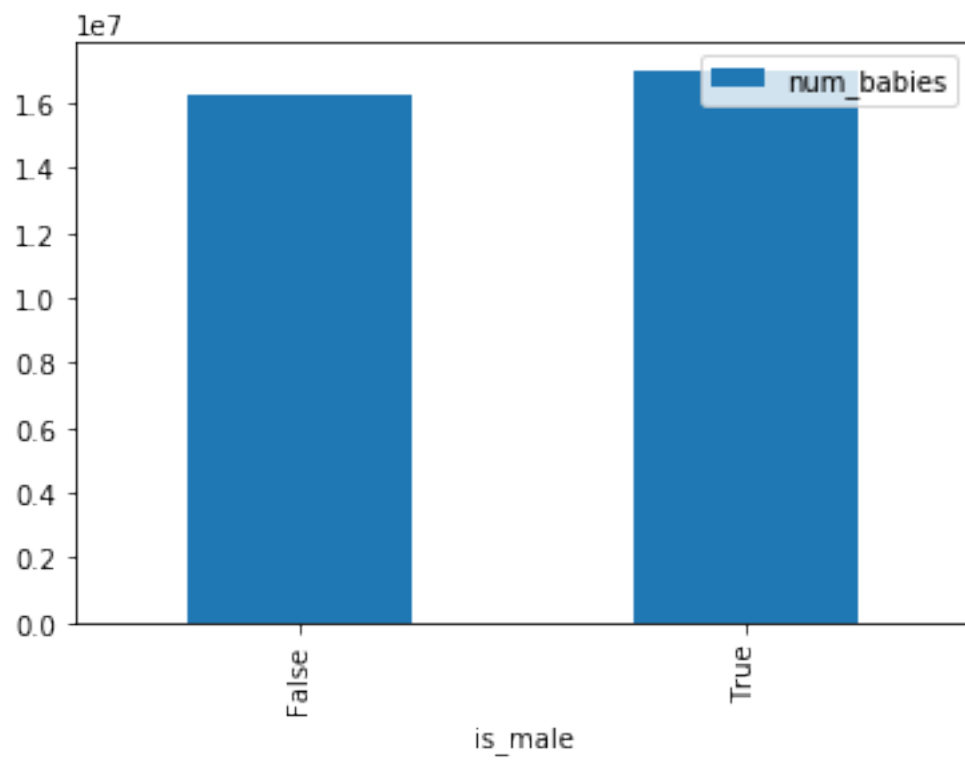
      hashmonth
0  7108882242435606404
1  7170969733900686954
2  6392072535155213407
3  2126480030009879160
4  3408502330831153141
```

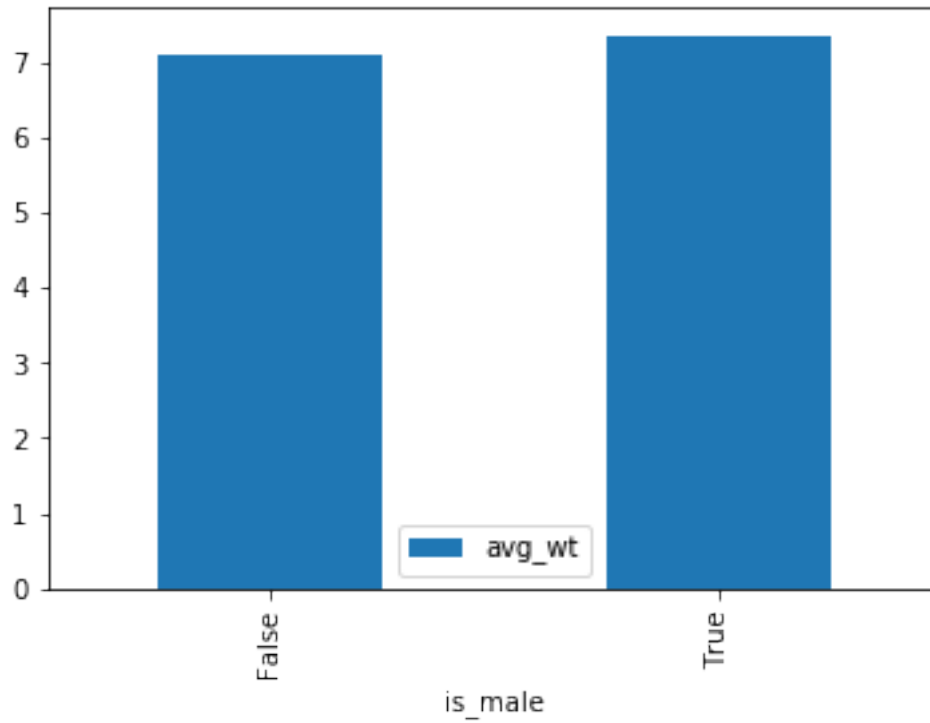
Let's write a query to find the unique values for each of the columns and the count of those values. This is important to ensure that we have enough examples of each data value, and to verify our hunch that the parameter has predictive value.

```
[10]: # Create function that finds the number of records and the average weight for
      ↪ each value of the chosen column
def get_distinct_values(column_name):
    sql = """
SELECT
    {0},
    COUNT(1) AS num_babies,
    AVG(weight_pounds) AS avg_wt
FROM
    publicdata.samples.natality
WHERE
    year > 2000
GROUP BY
    {0}
    """.format(column_name)
    return bigquery.Client().query(sql).to_dataframe()
```

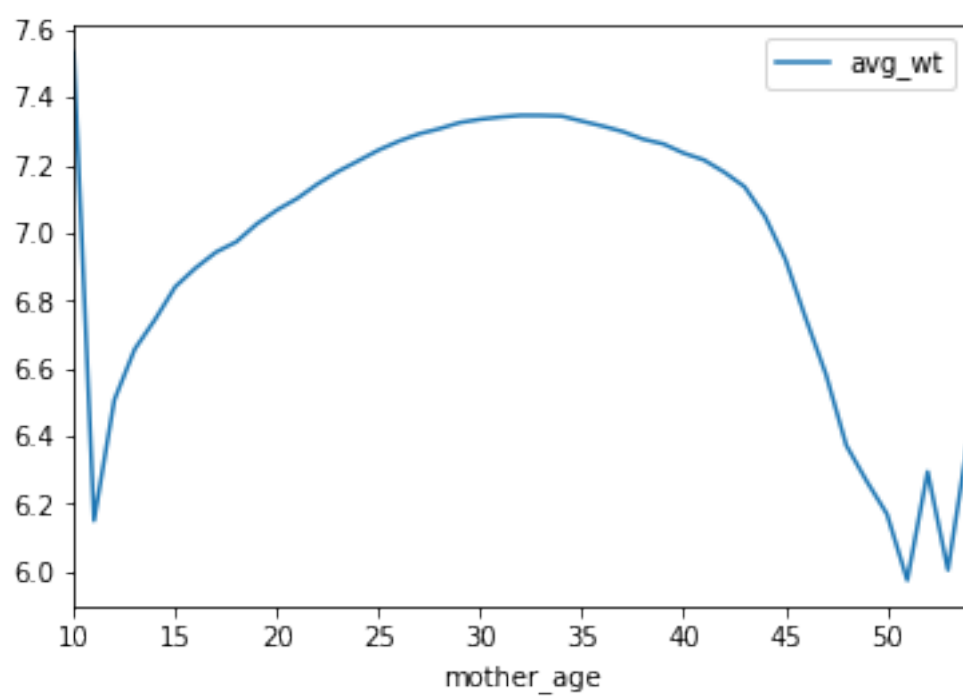
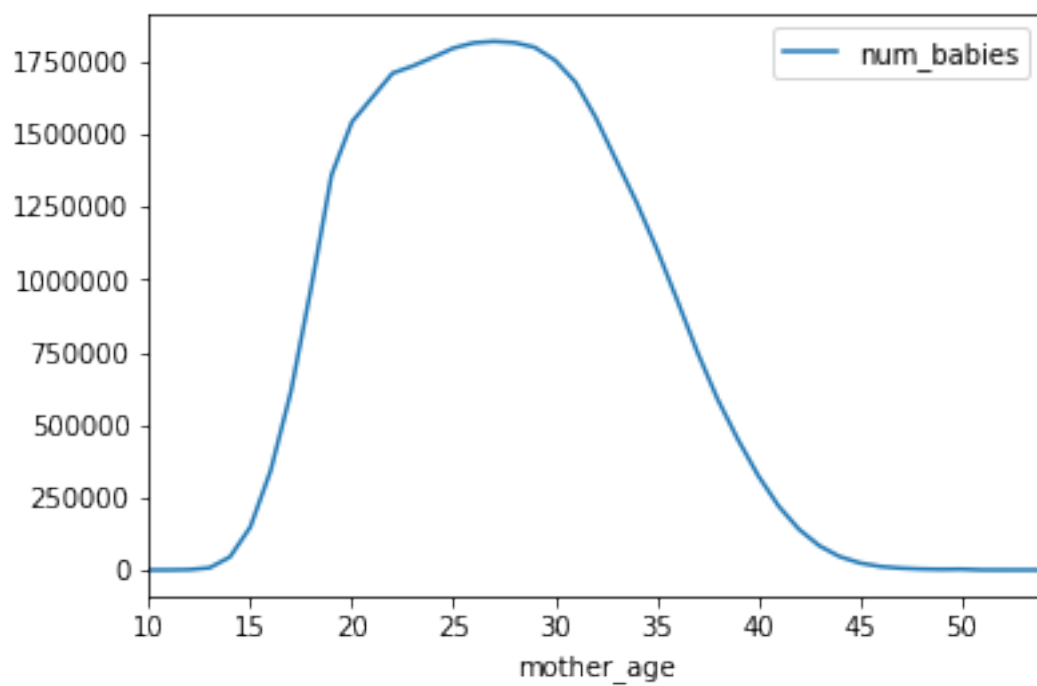
```
[11]: # Bar plot to see is_male with avg_wt linear and num_babies logarithmic
df = get_distinct_values('is_male')
```

```
df.plot(x='is_male', y='num_babies', kind='bar');  
df.plot(x='is_male', y='avg_wt', kind='bar');
```

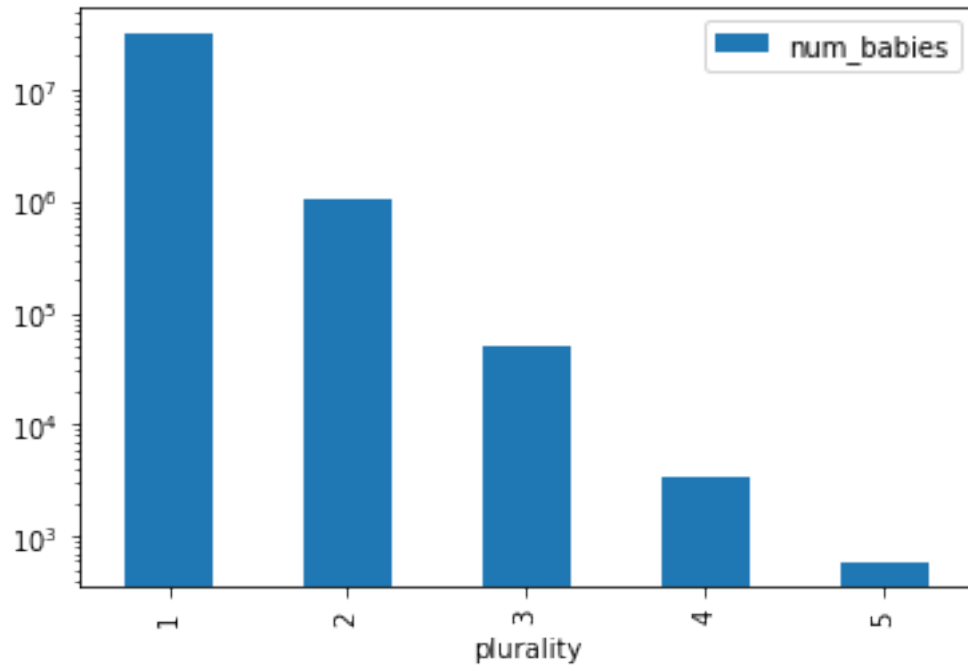


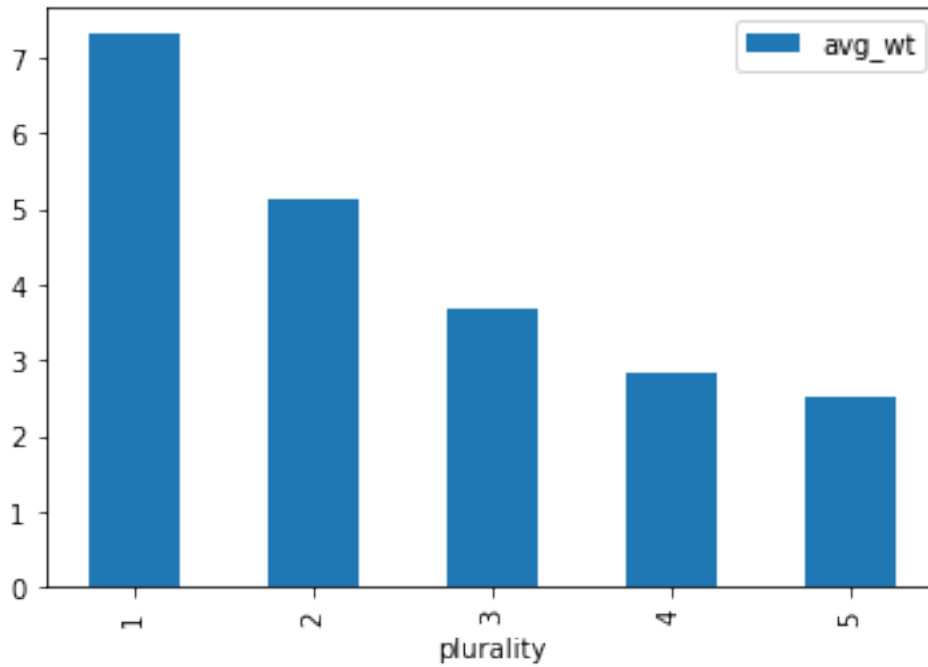


```
[12]: # Line plots to see mother_age with avg_wt linear and num_babies logarithmic
df = get_distinct_values('mother_age')
df = df.sort_values('mother_age')
df.plot(x='mother_age', y='num_babies');
df.plot(x='mother_age', y='avg_wt');
```

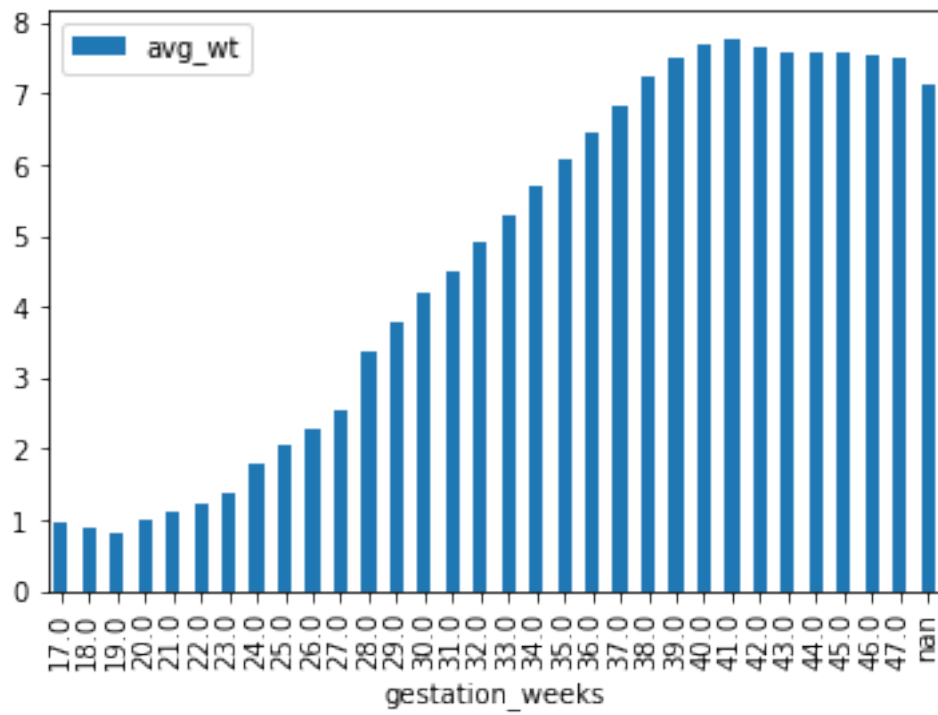
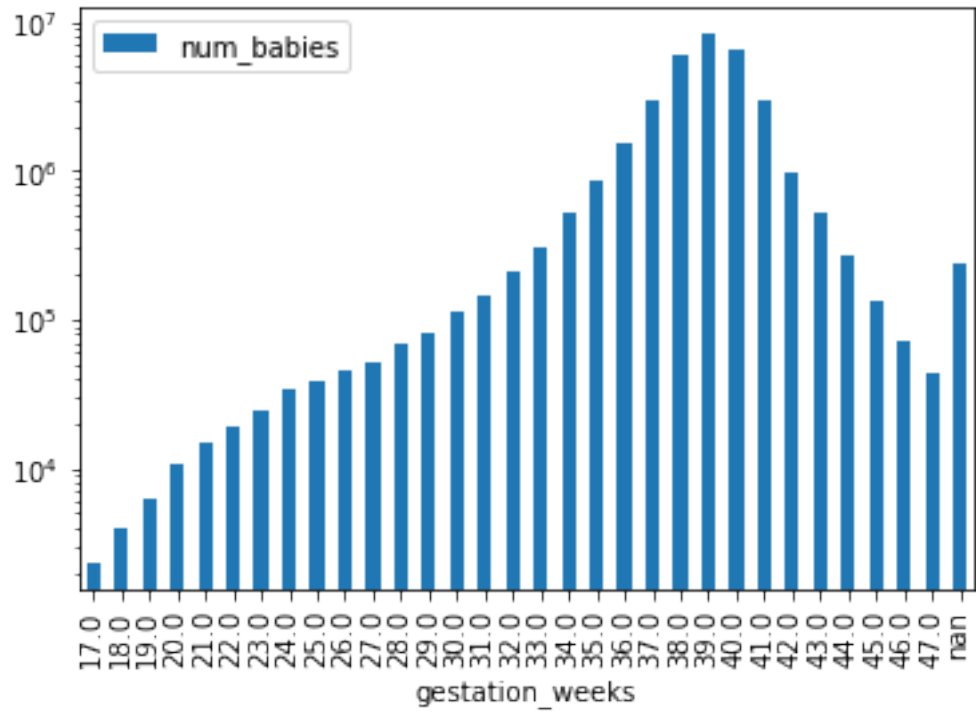


```
[13]: # Bar plot to see plurality(singleton, twins, etc.) with avg_wt linear and
      ↪ num_babies logarithmic
df = get_distinct_values('plurality')
df = df.sort_values('plurality')
df.plot(x='plurality', y='num_babies', logy=True, kind='bar');
df.plot(x='plurality', y='avg_wt', kind='bar');
```





```
[14]: # Bar plot to see gestation_weeks with avg_wt linear and num_babies logarithmic
df = get_distinct_values('gestation_weeks')
df = df.sort_values('gestation_weeks')
df.plot(x='gestation_weeks', y='num_babies', logy=True, kind='bar');
df.plot(x='gestation_weeks', y='avg_wt', kind='bar');
```



All these factors seem to play a part in the baby's weight. Male babies are heavier on average than female babies. Teenaged and older moms tend to have lower-weight babies. Twins, triplets, etc. are lower weight than single births. Premies weigh in lower as do babies born to single moms. In addition, it is important to check whether you have enough data (number of babies) for each input value. Otherwise, the model prediction against input values that doesn't have enough data may not be reliable.

In the next notebook, I will develop a machine learning model to combine all of these factors to come up with a prediction of a baby's weight.

Copyright 2017-2018 Google Inc. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License