Names Project

Course CSE 250

Christian Lira Gonzalez

Elevator pitch

In this Project we are trying to identify certain values during a certain time frame. We have a set of data with over hundreds lines of data that we want to transform into visualization.

GRAND QUESTION 1

How does your name at your birth year compare to its use historically?

The Name Christian has been used Historically for a really long time. In the United States of America, a nation founded in the Chridtian principles the name has also been highly famous and used but it was not until the 90's that the name had a Boom and was used if not frequent, more frequently. It is inthe 2000's when the name reaches it's highest peak and where I position myself as well, even though i was not born in the states it appears such booming influenced not the states but other neighboring countries such as Mexico, country where I was born.
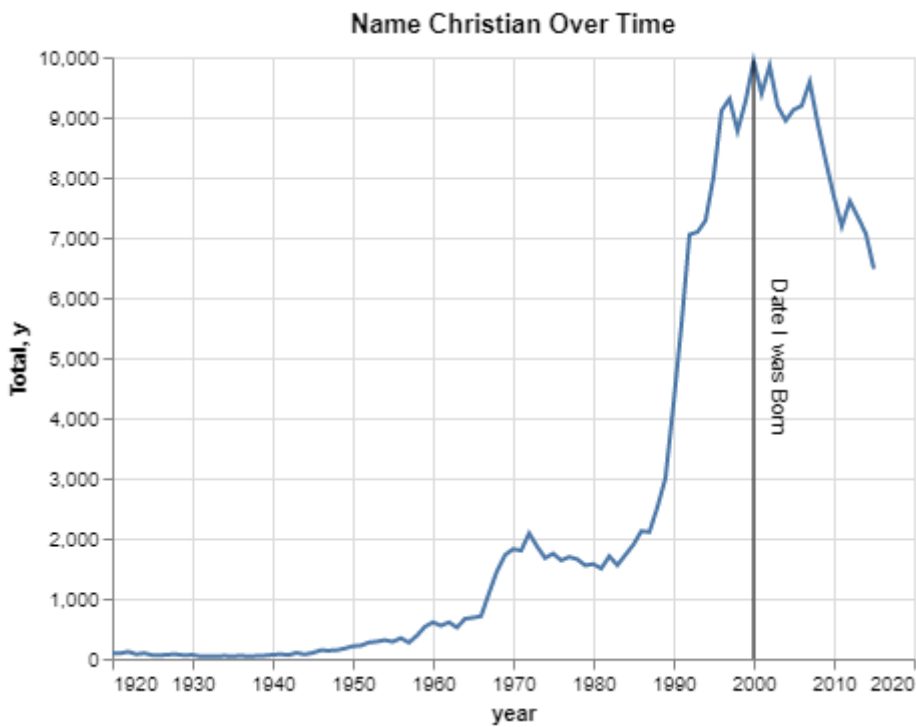
TECHNICAL DETAILS

```python
#%%
from turtle import title
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#%%
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#Then we code for the information we want to specifically find in the database, in this case we
Christian  = dat.query("name == 'Christian'")
Christian  = (dat
 .query("name == 'Christian'")
 .query("year >= 1920 & year <= 2015")
)
#%%
#Because we did a query through a function "Christian" we can graph the information we found thr
Christian_chart_name = (alt.Chart(Christian ).mark_line().encode(
  x = alt.X("year",
    title= "year",
    axis = alt.Axis(format ='d')),

  y = alt.Y ("Total")
).properties(title="Name Christian Over Time"))

Christian_chart_name.show()
#%%
#we make another query through a function "Christian" and we can graph the information we found
date_of_birth = Christian .query("year == 2000")
Christian_chart_year = (alt.Chart(date_of_birth).mark_rule()
  .encode(
    x= "year",
    y = "Total")
)
text5 = alt.Chart({'values':[{'x': 2003, 'y': 5000}]}).mark_text(
    text='Date I was Born', angle=90
).encode(
    x='x:Q', y='y:Q'
)
Christian_chart_year.show()
#now we combine the graphs so the information can be visualized both at the same time
Christian_charts = Christian_chart_name + Christian_chart_year + text5
Christian_charts.show()
#%%
```

Name Christian Over Time

GRAND QUESTION 2

If you talked to someone named Brittany on the phone, what is your guess of their age? What ages would you not guess?

If we talked to someone named Brittany our most correct age to guess would be somewhere between 27 years old and 37 years old with a highest possiblity to find an average of 32 years old. I proceed to explain this trend:

If we pay close attention to our chart we can notice that between the years of 1985 and 1995 there is an increase of the use of such name but it is in the midterm between these years we encounter a higher rate of it usage with an increase of almost 1.5 more times than in 1985 or 1995.

TECHNICAL DETAILS

```python
#%%
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#%%
url =('names_year.csv')
dat = pd.read_csv(url)
# print(dat)

brittany = dat.query("name == 'Brittany'")

brit_chart_name = (alt.Chart(brittany).mark_line()
    .encode(
        x = "year".replace(", "," "),
        y = "Total")
)
brit_chart_name.show()


#%%
date_of_birth = brittany.query("year == 1990")

brit_chart_year = (alt.Chart(date_of_birth).mark_rule().encode(
    x= alt.X ("year",
        title = "year",
        axis = alt.Axis(format = 'd')),

    y = alt.Y ("Total")
).properties(title = 'Name Brittany Over Time'))

brit_chart_year.show()
#%%
total_brits = brittany.Total.sum()
print("I printed" , total_brits)
# %%
textbrit = alt.Chart({'values':[{'x': 1992, 'y': 10000}]}).mark_text(
    text='Guess of the age for Brittany', angle= 90
).encode(
    x='x:Q', y='y:Q'
)
brit_charts = brit_chart_name + brit_chart_year +textbrit
brit_charts.show()
```
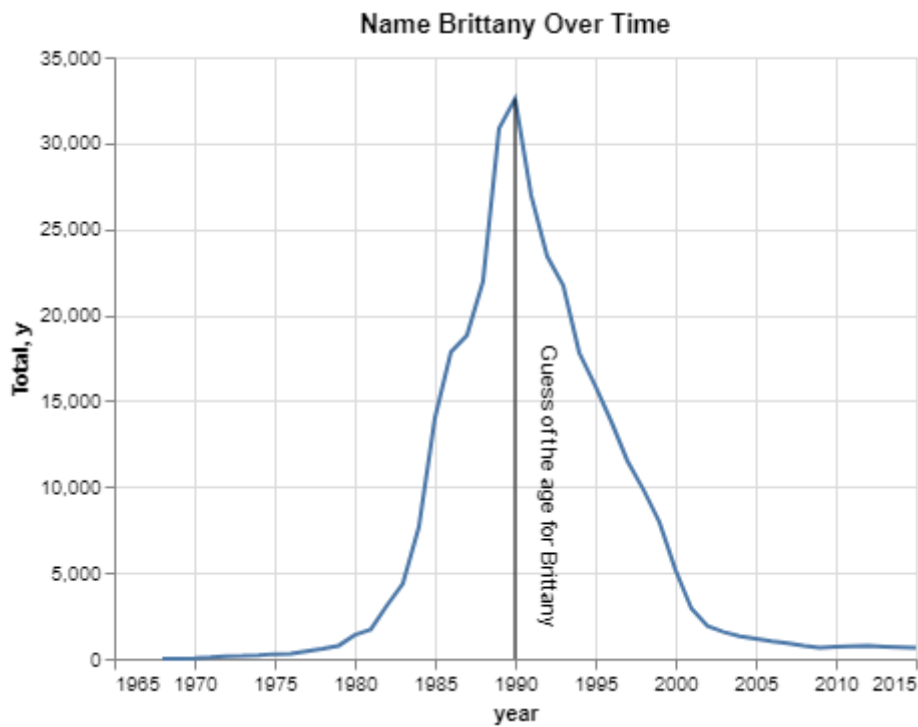
Name Brittany Over Time

GRAND QUESTION 3

Mary, Martha, Peter, and Paul are all Christian names. From 1920 - 2000, compare the name usage of each of the four names.

Here we can see a comparison of the Four names: Mary, Martha, Paul and Peter. all four Christian Names But we can compare their frequency to determine their popularity throught the last 100 years.

TECHNICAL DETAILS

```python
#This is the code for the 3rd grand question:
#The first part is the code for Mary
#%%
# First we have to  import every package that will play an essential role in the activity
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#%%

comparisson = (dat
   .query('(name == "Mary" or name == "Martha" or name == "Peter" or name == "Paul") and year > 1
   .filter(['name', 'year', 'Total'])
   .sort_values('year')
   .reset_index()
)

chart3 = alt.Chart(comparisson, title="Comparison of names").mark_line().transform_fold(
   fold=['name', 'year', 'Total'],
   ).encode(
     x = alt.X('year', title="Birth Year name"),
     y = alt.Y('Total', title="Amount of Newborns"),
     color='name'
)
chart3.show()
# %%
```


Comparison of names
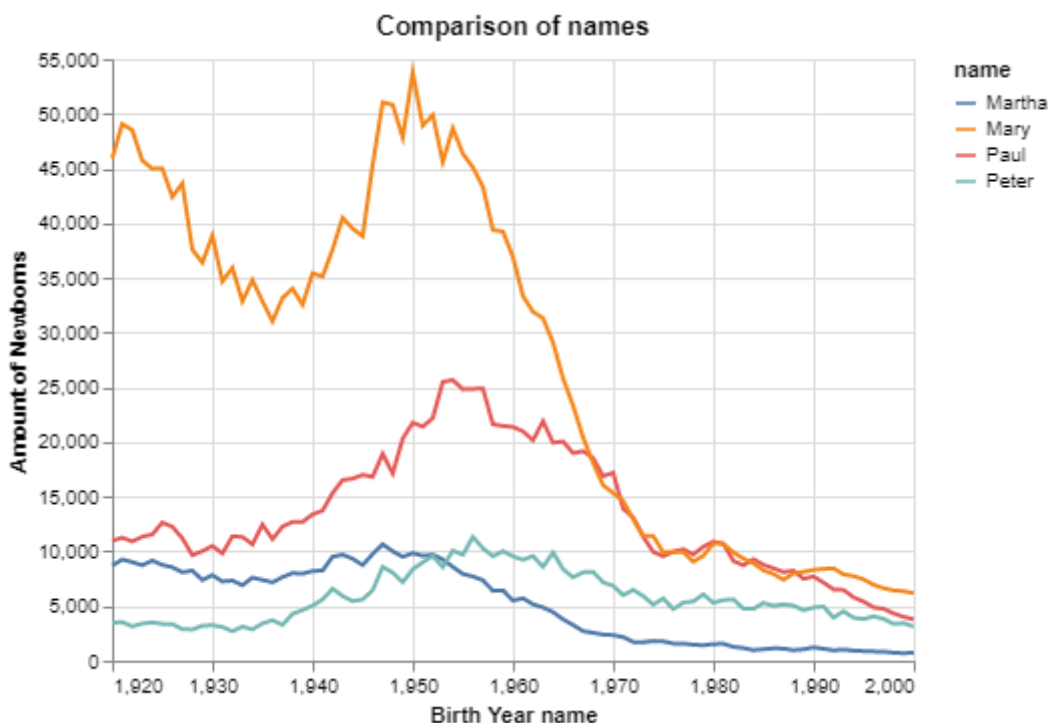
GRAND QUESTION 4

Think of a unique name from a famous movie. Plot that name and see how increases line up with the movie release.

The chosen Name was "Marlon" in honor of Marlon Brando the famous movie star, well known due to his character in "The Godfather" as The Goddfather himself. When we ponder about the grand question and reflect it's idea on the name Marlon we can not deny the popularity such movie had and the impact of the names of newborns at such time. The movie was launched during the year of 1972 and due to it's popularity and awesome reception we notice that the name's usage seem to be rocketed to its highest peak in that very year (1972).
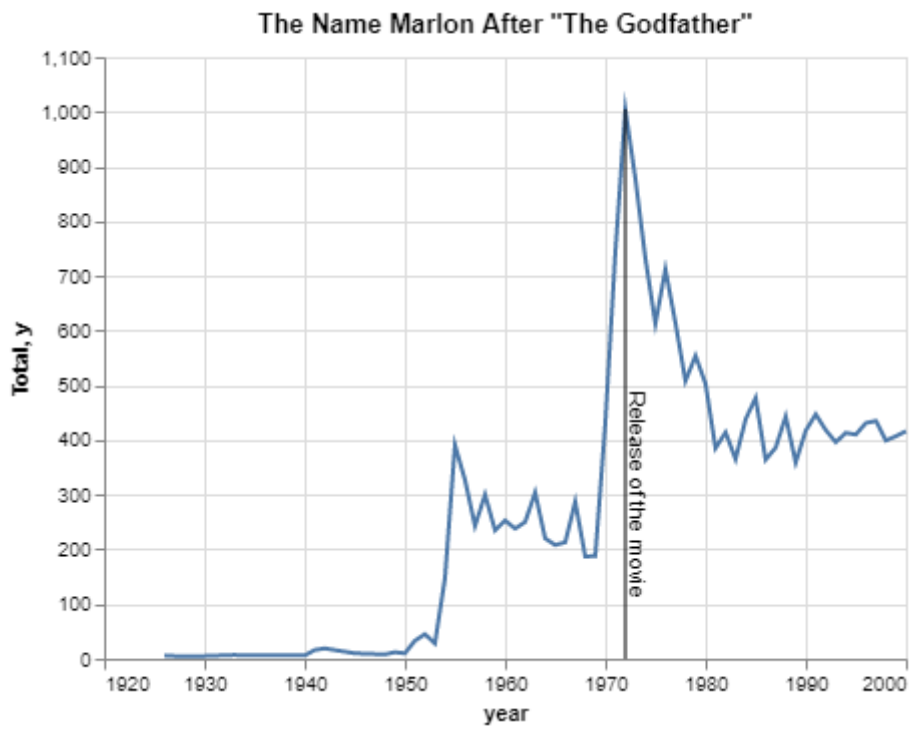
TECHNICAL DETAILS

```python
#%%
# First we have to  import every package that will play an essential role in the activity
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#Then we code for the information we want to specifically find in the database, in this case we
Marlon  = dat.query("name == 'Marlon'")
Marlon  = (dat
 .query("name == 'Marlon'")
 .query("year >= 1920 & year <= 2000")
)
#Because we did a query through a function "Marlon " we can graph the information we found throu
Marlon_chart_name = (alt.Chart(Marlon ).mark_line()
   .encode(
     x = alt.X ("year",
     title= 'year',
     axis = alt.Axis(format = 'd')),
     y = alt.Y ("Total")
).properties(title = 'The Name Marlon After "The Godfather"'))
Marlon_chart_name.show()

#we make another query through a function "Marlon " and we can graph the information we found th
date_of_birth = Marlon .query("year == 1972")
Marlon_chart_year = (alt.Chart(date_of_birth).mark_rule()
   .encode(
     x= "year",
     y = "Total")
)
Marlon_chart_year.show()
#now we combine the graphs so the information can be visualized both at the same time
textmar = alt.Chart({'values':[{'x': 1973, 'y': 300}]}).mark_text(
     text='Release of the movie', angle=90
).encode(
     x='x:Q', y='y:Q'
)
Marlon_charts = Marlon_chart_name + Marlon_chart_year +textmar
Marlon_charts.show()
# %%
```

The Name Marlon After "The Godfather"

APPENDIX A (PYTHON CODE)

```python
#Grand Question 1
%%
from turtle import title
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#%%
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#Then we code for the information we want to specifically find in the database, in this case we
Christian  = dat.query("name == 'Christian'")
Christian  = (dat
 .query("name == 'Christian'")
 .query("year >= 1920 & year <= 2015")
)
#%%
#Because we did a query through a function "Christian" we can graph the information we found thr
Christian_chart_name = (alt.Chart(Christian ).mark_line().encode(
  x = alt.X("year",
    title= "year",
    axis = alt.Axis(format ='d')),

  y = alt.Y ("Total")
).properties(title="Name Christian Over Time"))

Christian_chart_name.show()
#%%
#we make another query through a function "Christian" and we can graph the information we found
date_of_birth = Christian .query("year == 2000")
Christian_chart_year = (alt.Chart(date_of_birth).mark_rule()
  .encode(
    x= "year",
    y = "Total")
)
text5 = alt.Chart({'values':[{'x': 2003, 'y': 5000}]}).mark_text(
    text='Date I was Born', angle=90
).encode(
    x='x:Q', y='y:Q'
)
Christian_chart_year.show()
#now we combine the graphs so the information can be visualized both at the same time
Christian_charts = Christian_chart_name + Christian_chart_year + text5
Christian_charts.show()
#%%
#%%
```

```python
#Grand Question 2


import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#%%
url =('names_year.csv')
dat = pd.read_csv(url)
# print(dat)

brittany = dat.query("name == 'Brittany'")

brit_chart_name = (alt.Chart(brittany).mark_line()
  .encode(
    x = "year".replace(", "," "),
    y = "Total")
)
brit_chart_name.show()


#%%
date_of_birth = brittany.query("year == 1990")

brit_chart_year = (alt.Chart(date_of_birth).mark_rule().encode(
  x= alt.X ("year",
    title = "year",
    axis = alt.Axis(format = 'd')),

  y = alt.Y ("Total")
).properties(title = 'Name Brittany Over Time'))

brit_chart_year.show()
#%%
total_brits = brittany.Total.sum()
print("I printed" , total_brits)
# %%
textbrit = alt.Chart({'values':[{'x': 1992, 'y': 10000}]}).mark_text(
    text='Guess of the age for Brittany', angle= 90
).encode(
    x='x:Q', y='y:Q'
)
brit_charts = brit_chart_name + brit_chart_year +textbrit
brit_charts.show()




#Grand Question 3
```

```python
#This is the code for the 3rd grand question:
#The first part is the code for Mary
#%%
# First we have to  import every package that will play an essential role in the activity
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#%%
q3_1 = (dat
   .query('(name == "Mary" or name == "Martha" or name == "Peter" or name == "Paul") and year > 1
   .filter(['name', 'year', 'Total'])
   .sort_values('year')
   .reset_index()
)

chart3 = alt.Chart(q3_1, title="Comparison of names").mark_line().transform_fold(
   fold=['name', 'year', 'Total'],
   ).encode(
     x = alt.X('year', title="Birth Year name"),
     y = alt.Y('Total', title="Amount of Newborns"),
     color='name'
)
chart3.show()



#Grand Question 4


# First we have to  import every package that will play an essential role in the activity
import pandas as pd
import altair as alt
import numpy as np
import tabulate as tab
import requests as req
#Once we have imported every packge we code so the database is read through csv we create functi
url =('names_year.csv')
dat = pd.read_csv(url)
#Then we code for the information we want to specifically find in the database, in this case we
Marlon  = dat.query("name == 'Marlon'")
Marlon  = (dat
 .query("name == 'Marlon'")
 .query("year >= 1920 & year <= 2000")
)
```

```python
#Because we did a query through a function "Marlon " we can graph the information we found throu
Marlon_chart_name = (alt.Chart(Marlon ).mark_line()
  .encode(
    x = alt.X ("year",
    title= 'year',
    axis = alt.Axis(format = 'd')),
    y = alt.Y ("Total")
).properties(title = 'The Name Marlon After "The Godfather"'))
Marlon_chart_name.show()

#we make another query through a function "Marlon " and we can graph the information we found th
date_of_birth = Marlon .query("year == 1972")
Marlon_chart_year = (alt.Chart(date_of_birth).mark_rule()
  .encode(
    x= "year",
    y = "Total")
)
Marlon_chart_year.show()
#now we combine the graphs so the information can be visualized both at the same time
textmar = alt.Chart({'values':[{'x': 1973, 'y': 300}]}).mark_text(
    text='Release of the movie', angle=90
).encode(
    x='x:Q', y='y:Q'
)
Marlon_charts = Marlon_chart_name + Marlon_chart_year +textmar
Marlon_charts.show()
# %%
```