STA 141B: Midterm (or Super-Homework if you prefer)

Worth 300 Points

Fall 2018

Information

After the colons (in the same line) please write just your first name, last name, and the 9 digit student ID number below.

First Name:Zack

Last Name: Liu

Student ID: 915026153

Instructions

Please print your answer notebook to pdf (make sure that it is not unnecessarily long due to long output) and submit as the homework solution with your zip file.

For readability you,

- MUST add cells in between the exercise statements and add answers within them and
- MUST NOT modify the existing cells, particularly not the problem statement
- you may add multiple cells between exercise cells

To make markdown, please switch the cell type to markdown (from code) - you can hit 'm' when you are in command mode - and use the markdown language. For a brief tutorial see:
https://daringfireball.net/projects/markdown/syntax (https://daringfireball.net/projects/markdown/syntax)

Trans-Atlantic Slave Trade

In this homework, we will uncover some of the numbers behind the Trans-atlantic slave trade (TAST), also known as the middle passage, that brought African slaves to the Americas. The middle passage is reported to have forcibly migrated over 10 million Africans to the Americas over a roughly 3 century time span. Many aspects of the TAST is little known by most people, such as the countries that constituted this network of slave ships, the regions from which the slaves were taken, and the number of slaves captured from Africa.

This last number is especially important since the number of slaves taken from Africa can impact other estimates that result from this. For example, when estimating the population of Africa in a given decade, demographers will use population growth models and more recent census data. For example, there are roughly X number of people in Africa and such populations tend to grow at rate M. Then if we want to calculate the population one century ahead then we just apply a simple formula that assumes that the population grows at this rate. But if the population is being drained by the slave trade, then this number will tend to be underestimated because the growth rate is overestimated. To account for this models need to take into account this drain on the population.

Throughout this homework you will need to follow the principles of graphical excellence and the grammar of graphics. **Use only Plotnine for your graphics**, do not use Pyplot, Seaborn, or Plotly since they do not follow closely the grammar of graphics. Be sure to include titles and necessary contextual captions.

Warning: The Trans-Atlantic Slave Trade remains one of the most horrific abuses of human rights in history. This homework deals with the numbers behind this forced migration, please be aware that this is a sensitive topic for possibly yourself and others. A suitable amount of respect and seriousness is required when dealing with this data.

Exercise 1. The data.

1. Read in the Trans-Atlantic Slave Trade database with Pandas. Hint: if you use the unix tool `file` you can find that this CSV is encoded with iso-8859-1 character set. Make sure that all missing values are encoded as NaN.
2. There is lots of missingness in this data, and some of these variables are imputed. We will be imputing some of these variables ourselves, so delete any variable that ends with 'imp'.
3. Open up the pdf file: TAST_codebook.pdf which is the data dictionary for this and other related datasets. Many of the variables in the codebook are not in this dataset because it is describing an updated dataset.
4. Create a list where you describe the meaning of the columns of your imported dataframe. You can group similar columns together when describing their rough meaning, such as ownera,...,ownerp are owners of the slave ships.

Throughout we will disregard all time variables other than year since they are unreliable.

In [1]:
```
import pandas as pd
import numpy as np
import plotnine as p9
import warnings
warnings.simplefilter('ignore')
```

In [2]:
```
#1.1 read data
data=pd.read_csv("tastdb-2010.csv",na_values=" ",encoding="iso-8859-1",low_memory=F

#1.2 delete columns ends with imp()
data.drop(columns=[col for col in data.columns if col[-3:]=="imp"], inplace=True)
print('In total {} columns'.format(len(data.columns)))
```

```
In total 89 columns
```

1.3&1.4

Variable Description

- **voyageid**: Voyage identification
- **evgreen**: Voyage in 1999 CD-ROM
- **shipname**: Name of vessel
- **national**: Country in which ship registered
- **placcons, yrcons**: Place and year of vessel's construction
- **placreg,yrreg**: Place and year of vessel's registration
- **rig**: Rig of vessel
- **tonnage**: Tonnage of vessel
- **tonmod**: Tonnage standardized on British measured tons
- **ownera, ownerb, ownerc, ownerd, ownere, ownerf, ownerg, ownerh,owneri, ownerj, ownerk, ownerl, ownerm, ownern, ownero, ownerp**: First to sixteeth owner of venture
- **fate, fate2, fate3, fate4**: Different outcomes

- **resistance**: African resistance
- **plac1tra, plac2tra, plac3tra**: Places of slave purchase
- **npafttra**: Port of call before Atlantic crossing
- **sla1port, adpsale1, adpsale1**: Places of slave landing
- **portret**: Place at which voyage ended
- **yearam**: Year of arrival at port of disembarkation
- **Date_dep**: Date that voyage began
- **Date_buy**: Date that slave purchase began
- **Date_leftAfr**: Date that vessel left last slaving port
- **Date_land1**: Date that slaves landed at first place
- **Date_depam**: Date ship left on return voyage
- **Date_end**: Date when voyage completed
- **captaina, captainb**: First and second captain's name
- **crew1, crew3** : Crew at voyage outset and first landing of slaves
- **crewdied**: Crew died during complete voyage
- **slintend**: Slaves intended from first port of purchase
- **ncar13, ncar15, ncar17**: Slaves carried from first, second and third port of purchase
- **tslavesd**: Total slaves on board at departure from last slaving
- **slaarriv**: Total slaves arrived at first port of disembarkation
- **slas32, slas36, slas39**: Slaves disembarked at first,second and third place
- **menrat7, womrat7, boyrat7, girlrat7, malrat7, chilrat7**: percentage of men, women, boys, girls, male, child at departure or arrival
- **jamcaspr**: Average price of slaves standardized on sterling cash price of prime slaves sold in Jamaica
- **vymrtrat**: Slave mortality rate (slave deaths / slaves embarked)
- **sourcea, sourceb, sourcec, sourced, sourcee, sourcef, sourceg, sourceh, sourcei, sourcej, sourcek, sourcel, sourcem, sourcen, sourceo, sourcep, sourceq, sourcer**: First to eighteenth source of information

Exercise 2. First pass at estimating the total number of captives.

1. We will ultimately try to estimate the number of people captured into slavery and forced through the middle passage. What variable would you use to estimate the total number of captives taken from Africa? Let me call this variable Var A in this problem statement. How much of the data for Var A is missing?
2. Create an initial estimate of the total number of captives taken from Africa by assuming that Var A is Missing Completely at Random.
3. What other variables do you expect to be associated with Var A and why? Give at least three possibilities. Which will probably be the most strongly associated with this variable? (I will be looking for a specific variable to be listed so be sure to think about the most strongly associated one.)

2.1

I would use `tslavesd` , which is the total slaves on board at departure from last slaving, to estimate the total number of captives taken from Africa. There are 26734 missing values of Var A.

In [3]:
```
#2.1 get number of NAs
sum(data["tslavesd"].isna())
```

Out[3]: 26734

2.2

To estimate the total number of captives, I first imputed the missing values as median of the non-missing values, then sum up all values. Thus, I get a estimate of 11298998 captives taken from Africa.

In [4]:
```
#2.2 get a estimate of total captives
VarA=data["tslavesd"].copy()
VarA[VarA.isna()]=VarA.median()
print(round(VarA.sum()))
```

```
11298998.0
```

2.3

Variables `slaarriv` , `ncar13` , `ncar15` are associated with Var A. `nar13` are `nar15` slaves carried from first and second port of purchase, so they are positively correlated to `tslavesd` . `slaarriv` is total slaves arrived at first port of disembarkation, so it should have a strong association with the total slaves departed at last port. The variable `slaarrive` should be the one that most strongly associated with Var A.

Exercise 3. The flag that the ships flew.

1. We want to understand the trends of the nationality of the slave ships (the flag that they flew under is in the national variable). Subselect the values of `national` that have more than 100 voyages with that value.
2. Create a DataFrame that filters out the voyages where `national` does not have one of these values. You should be retaining voyages with only these most common values.
3. Create a variable, `flag` , that is a string of easily readable names for these values by looking them up in the pdf codebook.
4. Using Plotnine, plot the counts of the voyages by flag as a function of voyage year. Think about how best to display the count of a voyage by year and then how should you be including the flag variable.
5. In this plot, what are the geometric elements and aesthetic mappings? What other components of the grammar of graphics are you using?
6. Do you observe any abrupt changes in the patterns of these counts for a given flag? Investigate the cause for this change (using Google, etc.).

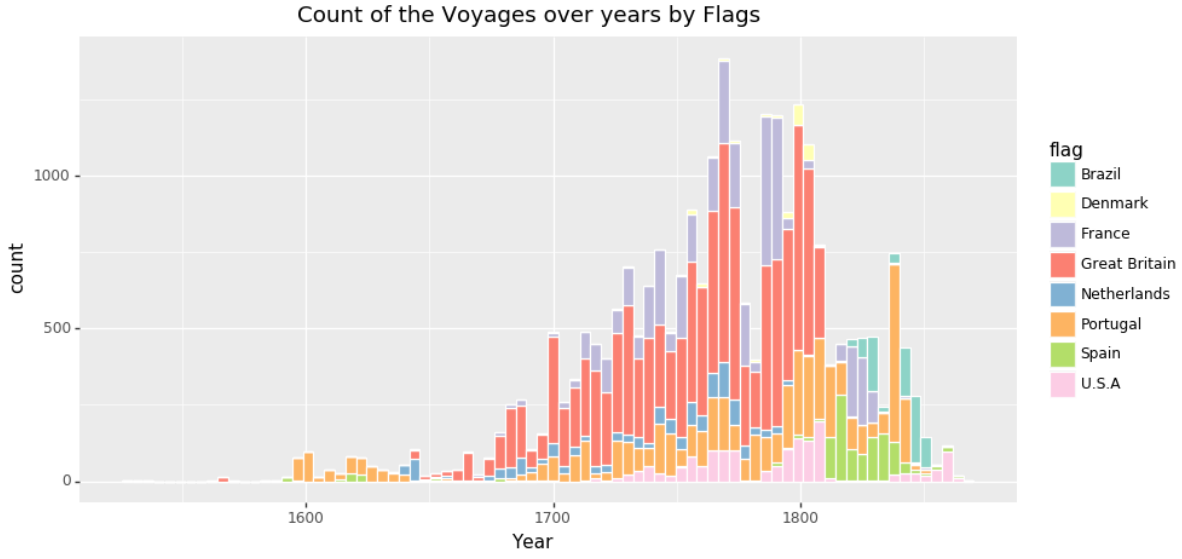
In [5]:
```
#3.1 subselect national with more than 100 voyages
CV=data['national'].value_counts()[data["national"].value_counts()>100].index.tolis

#3.2 create dataframe for selected nations.
CV_data=data[data.national.isin(CV)].copy()

#3.3 create variable flag
CV_data['national']=CV_data['national'].astype(int)
flag={7:"Great Britain", 4:"Portugal", 10:"France", 9:"U.S.A", 1:"Spain", 8:"Nether
     5:"Brazil", 11:"Denmark"}
CV_data["flag"]=CV_data['national'].map(flag)

#3.4 plot voyages by flag as a function of year
p9.ggplot(CV_data,p9.aes(x="yearam",fill="flag"))+p9.geom_histogram(color="white",b
    +p9.theme(figure_size=(10, 5))+p9.scale_fill_brewer(type='qual',palette=8)\
    +p9.labs(title="Count of the Voyages over years by Flags ",x="Year")

# print(p9.ggplot(CV_data,p9.aes(x="yearam",fill="flag"))+p9.geom_area(stat="bin")
#     p9.theme(figure_size=(10, 6))+p9.scale_fill_brewer(type='qual',palette=8) \
#     +p9.labs(title="Count of the Voyages over years by Flags ",x="Year"))
# print(p9.ggplot(CV_data,p9.aes(x="yearam",color="flag"))+p9.geom_freqpoly(stat="b
#     p9.theme(figure_size=(10, 6))+p9.scale_color_brewer(type='qual',palette=8) \
#     +p9.labs(title="Count of the Voyages over years by Flags ",x="Year"))
```



Out[5]: <ggplot: (8761611518917)>

3.5

The geometric elements in the plot above is histogram, and the aesthetic mapping is `yearam` to x-position and `flag` to color. I am also using scale, theme and labs.

3.6

The counts of voyages for Great Britain had suddenly drop to 0 after early 1800s. The cause is that Great Britain issued several acts to abolish slave trade in the early 1800s.

Exercise 4. Looking at some of these ships.

1. Search for the slave ship mentioned in the following wikipedia article: https://en.wikipedia.org/wiki/Brookes_(ship) (https://en.wikipedia.org/wiki/Brookes_(ship)) Hint: Look at all

records of ships with 'Brook' in the name and try to match the characteristics to those described. How many voyages for this ship are in the data (try to exclude ships with the same name)?

2. Create a variable that is True if there was a resistance (like a slave revolt) on the ship. Plot the density of ships as a function of year with and without revolts and compare these distributions.
3. The movie Amistad was based on a real slave ship and slave uprising. Read about it here: https://en.wikipedia.org/wiki/La_Amistad (https://en.wikipedia.org/wiki/La_Amistad) Try to find this ship by searching for it by name and also searching for ships in the same 10 year period as this event with a slave resistance. If you think you found it describe it, otherwise describe the events of another voyage that you did find.

4.1

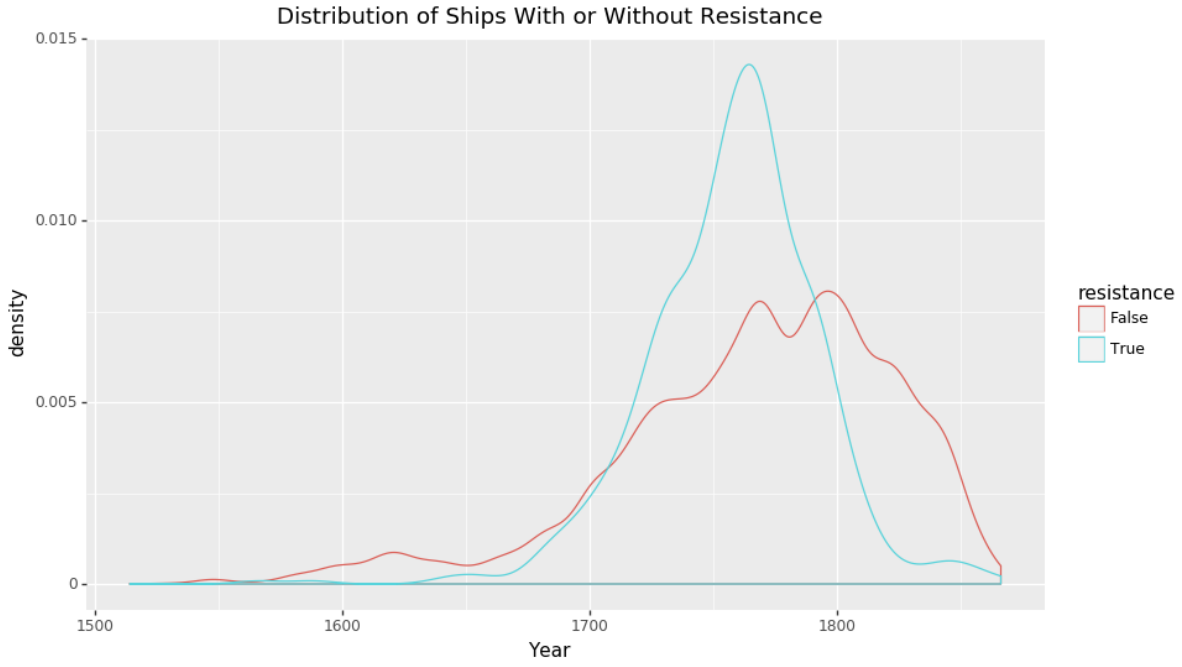
There are total 4 voyages in the data. From the article, I found the that the voyages of Brookes began in late 1700s. Also, by matching the number of slaves held in the Brookes with `tslavesd`, we can see that the Brookes mentioned in the article is co-owned by Joesph Brooks. By searching the data, I found that there are 4 voyages of the Brookes, which belonged to Joesph Brooks.

In [6]:
```
#4.1
#extract voyages contain "Brook" and find the specific Brookes
data.loc[data['shipname'].isna(),"shipname"]=" "
brookes=data[data["shipname"].str.contains("Brook")]
brookes[["shipname","ownera","ownerb","ownerc","ownerd","yearam","tslavesd"]].sort_
brookes.loc[29534:29537,["shipname","ownera","ownerc","tslavesd","yearam"]]
```

Out[6]:

	shipname	ownera	ownerc	tslavesd	yearam
29534	Brooks	Brooks, Joseph (Jr)	Rumbold, Thomas	650.0	1782
29535	Brooks	Noble, Clement	Brooks, Joseph (Jr)	619.0	1784
29536	Brooks	Noble, Clement	Brooks, Joseph (Jr)	740.0	1785
29537	Brooks (a) Brookes	Brooks, Joseph (Jr)	Molyneux, Thomas	609.0	1787

In [7]:
```
#4.2
data["revolt"]=data["resistance"].notnull()
data["revolt"]
p9.ggplot(data,p9.aes(x="yearam",color="revolt")) + p9.theme(figure_size=(10, 6))\
    +p9.geom_density()+p9.labs(x="Year",title="Distribution of Ships With or Withou
```



Out[7]: <ggplot: (-9223363275243449884)>

4.2

By Comparing two distributions, we can see that both distribution are left skewed. There lots of resistance occurred in the mid 1700s relatively, and most of the voyages occurred between 1700 and 1800

4.3

By searching the ship name that contains "Amistad", I didn't find any record of Amistad that matches the description on Wikipedia. After searching the ships in the same 10 year period with a slave resistance, I found that there are two ships matching the criteria. The interest thing is that these two ships both landing in Sierra Leone. Sierra Leone is the place where Joseph Cinqué, the leader of the slave revolt on Amistad, came from. Maybe, Joseph Cinqué's revolt was influenced by these two events.

In [8]:
```
#4.3
#searching for Amsitad
amistad=data[data["shipname"].str.contains("Amistad")].sort_values("yearam")
amistad[["shipname","yearam","tslavesd","slaarriv","Date_land1","plac1tra","sla1por

#searching same 10 year period with resisitance
result=data[data["yearam"].isin(range(1830,1840))& data["resistance"].notnull()]
result[["shipname","yearam","tslavesd","slaarriv","Date_land1","plac1tra","sla1port
```

Out[8]:

	shipname	yearam	tslavesd	slaarriv	Date_land1	plac1tra	sla1port	resistance	national
2504	Temerário	1837	352.0	254.0	2/22/1837	60605.0	60207.0	1.0	4.0
2789	Virginie	1831	92.0	92.0	3/20/1831	60217.0	60207.0	1.0	10.0

Exercise 5. Other patterns.

1. The arrival and departure locations are quite detailed. Look in the appendix of the codebook for the location codes. Make a coarser version of both arrival and departure port variables (select just the last departure and first arrival) so that for example,

   ```
   30000 Caribbean 36100 Martinique 36101 Fort-Royale
   ```

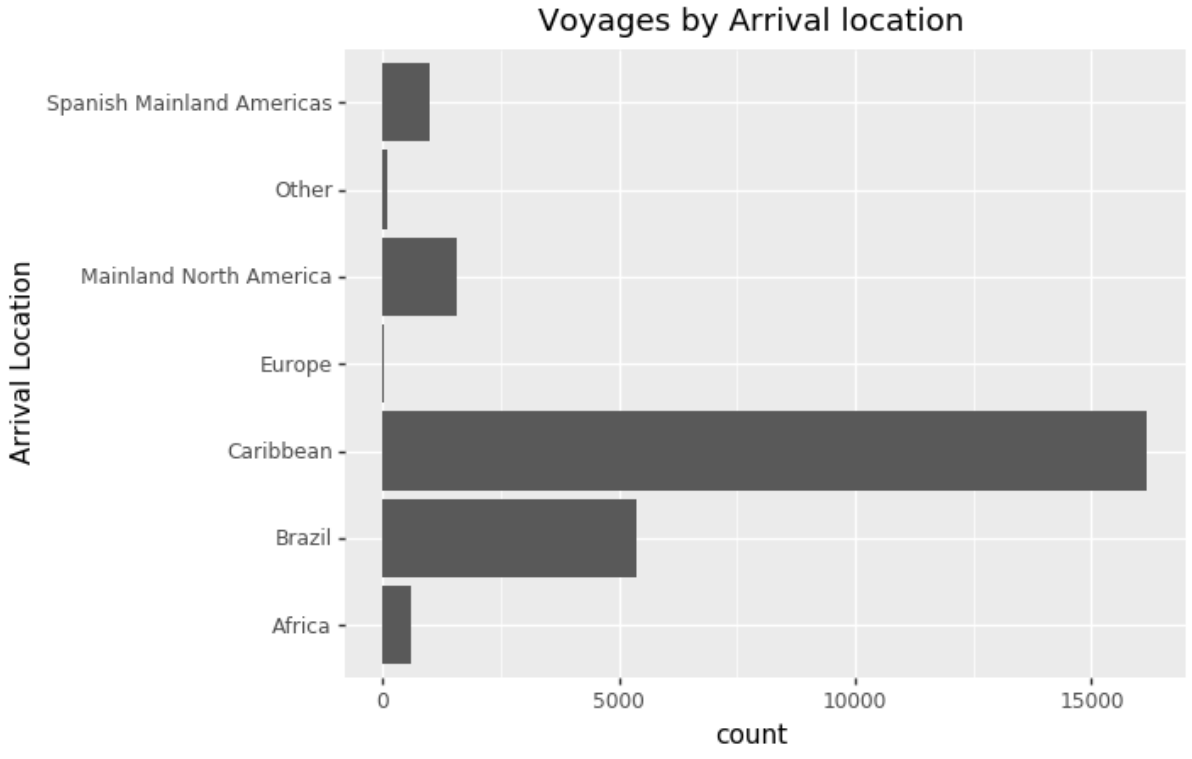
   is just encoded as '3' or Caribbean.
2. Plot the trend of voyages as a function of arrival location. What trends do you see?
3. Do the same for departure location.
4. Plot the ratio of captives that are men as a function of year. Include a smoother to describe the over all trend. Also include in the plot another possible confounding variable.
5. Describe the geoms, aesthetic mappings, and other aspects of the plot.

In [9]:
```
#5.1 make coarser version of arrival and departure variable
location_map={1:"Europe",2:"Mainland North America",3:"Caribbean", 4:"Spanish Mainl
              6:"Africa",8:"Other"}
data["arri_port"]=(data["sla1port"]//10000).map(location_map)
data["dep_port"]=(data["plac3tra"]//10000).map(location_map)
```

5.2

From the plot below, we can see that most of the voyages were landing in Caribbean and Brazil. The trend is that most of slaves were shipped to America.

In [10]:
```
#5.2 plot voyages vs arrival location
p9.ggplot(data[data["arri_port"].notnull()],p9.aes(x="arri_port"))+p9.geom_bar()+p9
    +p9.labs(x="Arrival Location",title="Voyages by Arrival location")
```



Out[10]: <ggplot: (-9223632752438459 58)>

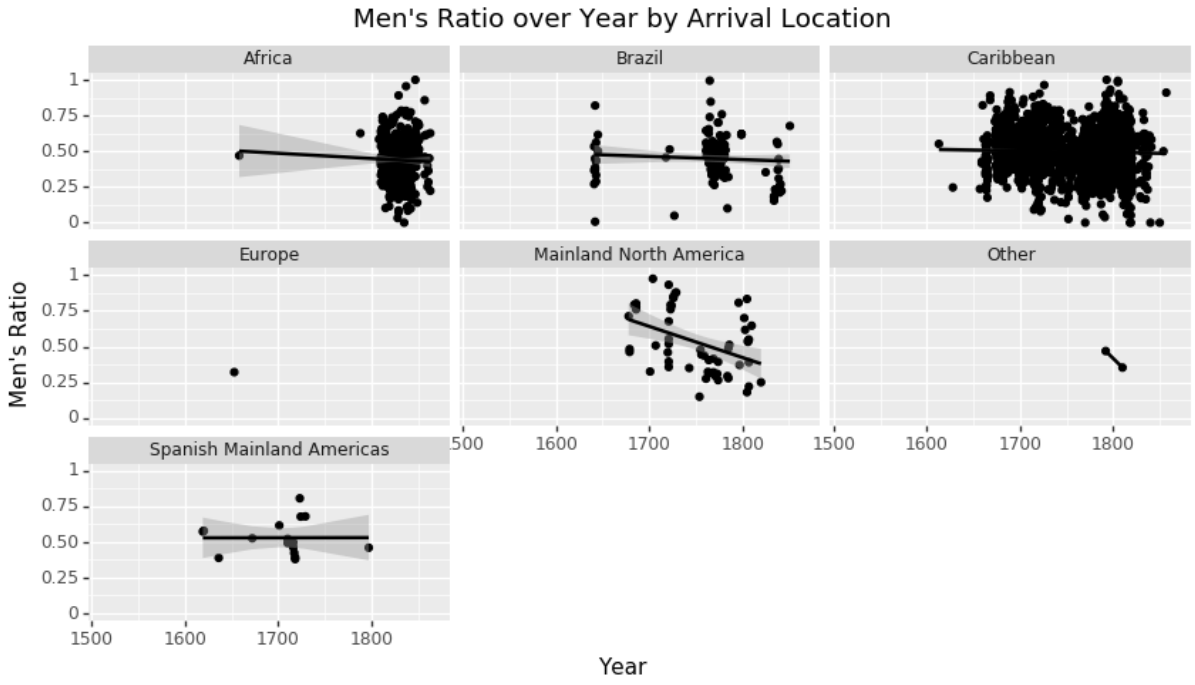
5.3

Skipped

5.4

When plotting the men's ratio as a function of year before including the possible confounding variable, the trend of men's ratio is decreasing over time. However, after including arrival location,which we created in **Exercise 5.2**, as the confounding variable,the men's ratio only decreases for "Mainland North America" and "other". The men's ratio stays about the same over time for the rest of the locations.

In [11]:
```
#5.4 plot men's ratio over year with arri_port
p9.ggplot(data[data["arri_port"].notnull()],p9.aes(x="yearam",y="menrat7"))+p9.geom
    +p9.facet_wrap("arri_port")+p9.geom_smooth(method="lm")+p9.theme(figure_size=(1
    +p9.labs(x="Year",y="Men's Ratio",title="Men's Ratio over Year by Arrival Locat
```

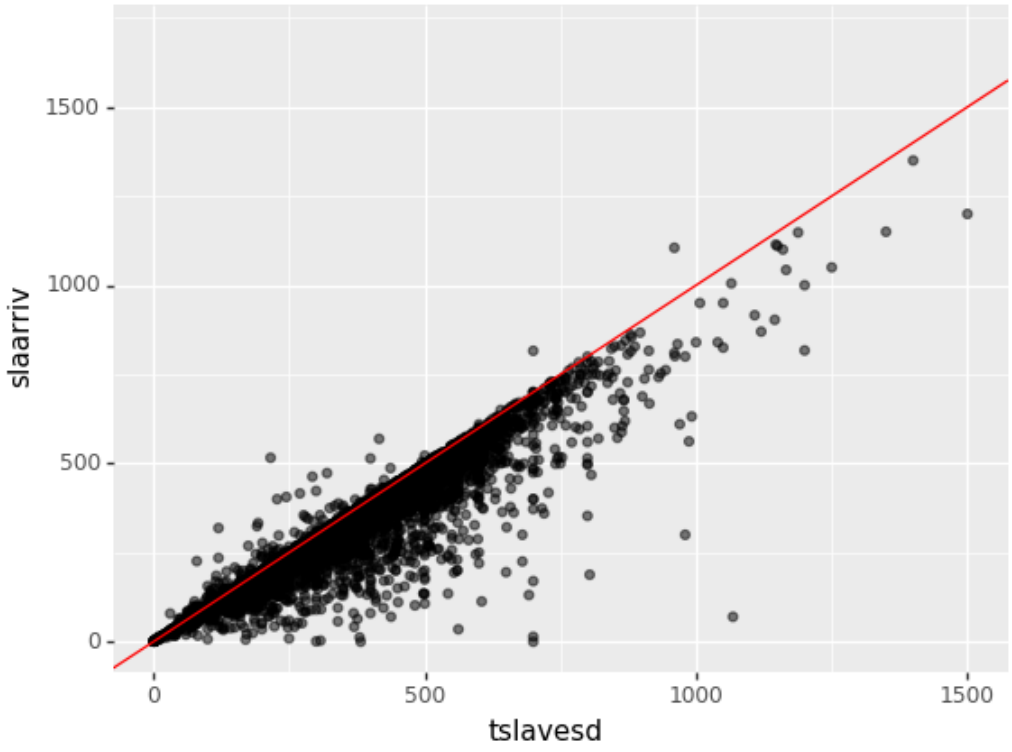


Out[11]: <ggplot: (8761611217998)>

Exercise 6. Estimating total captives count I.

1. Plot the number of slaves at departure from last slaving and the number that arrived at the first port of disembarkation.
2. Why do these variables concentrate near a line? Most of the data lies on one side of the line, but a small fraction do not, what could have happened for those that did not? Separate the data into these two populations.
3. For those voyages where the number of captives decreased (through death), calculate a death rate (ratio of captives that died).
4. Plot this death rate as a function of year. Make any appropriate scale transformations, and include a trend line or other smoother. What is your interpretation?
5. Guess at two possible predictors and create visualizations that give you a sense of their association with the death rate. Make sure that one of these is categorical, and plot it with the death rate and year so that there are three variables used in the same plot.
6. For these plots, list the geoms, aesthetic mappings, scales, and other notable aspects used.

In [12]:
```
#6.1 plot "tslavesd" vs "slaarriv"
p9.ggplot(data,p9.aes("tslavesd","slaarriv"))+p9.geom_point(alpha=0.5)+p9.geom_abli
```



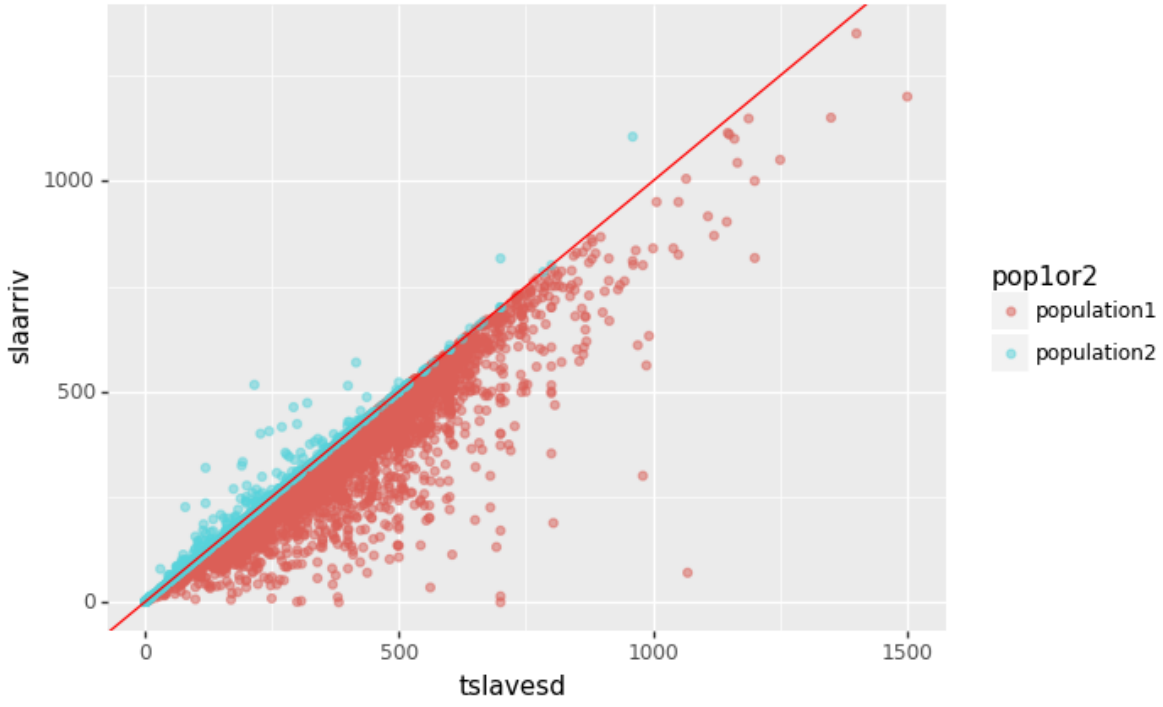
Out[12]: <ggplot: (-9223363275243255534)>

6.2

Ideally, number of slaves at departure and number of slaves at arrival should be the same, and they should be on the line Y=X . However, lots of captives probably died during the voyages, so most of the data lies on the bottom side of the line. The small fraction above the line is probably due to inaccurate numbers of slaves at departure were provided. Due to inhumanity in the middle passage, slave traders may hold more slaves in ships than the numbers provided in the data to maximize their profits.

In [13]:
```
#6.2
#separate two populations using a new variable pop1or2
slave_data=data[data["slaarriv"].notnull()&data["tslavesd"].notnull()].copy()
slave_data["pop1or2"]=slave_data["slaarriv"]<slave_data["tslavesd"]
pop_map={True:"population1",False:"population2"}
slave_data["pop1or2"]=slave_data["pop1or2"].map(pop_map)

#plot separate two populations by color
p9.ggplot(slave_data,p9.aes("tslavesd","slaarriv",color="pop1or2"))+p9.geom_point(a
          +p9.geom_abline(color="red")
```



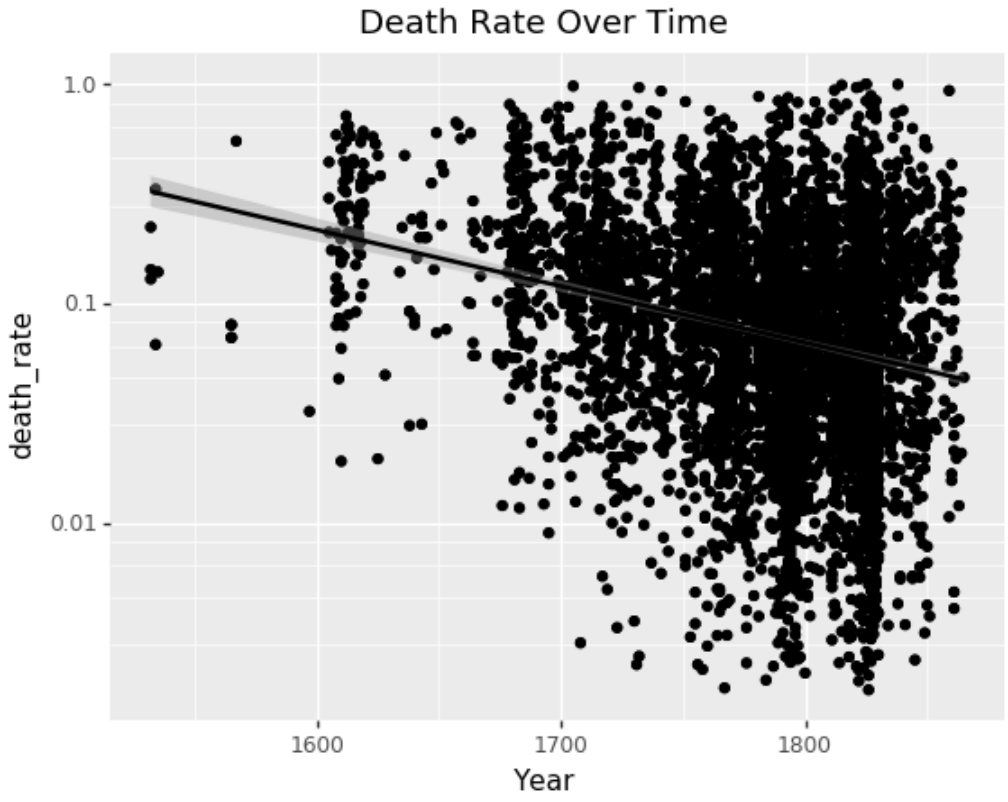
Out[13]: <ggplot: (-9223363275245887267)>

In [14]:
```
#6.3
#calculates death rate for captives decreased and store in "death_rate"
pop1=slave_data[slave_data["pop1or2"]=="population1"].copy()
pop1["death_rate"]=(pop1["tslavesd"]-pop1["slaarriv"])/pop1["tslavesd"]
```

6.4

From the plot below we can see that the captives death rates were decreasing over years.

In [15]:
```
#6.4 plot death rate as function of year with log transformation on death_rate
p9.ggplot(pop1,p9.aes("yearam","death_rate"))+p9.geom_point()+p9.geom_point()+p9.sc
    +p9.labs(x="Year",title="Death Rate Over Time")
```

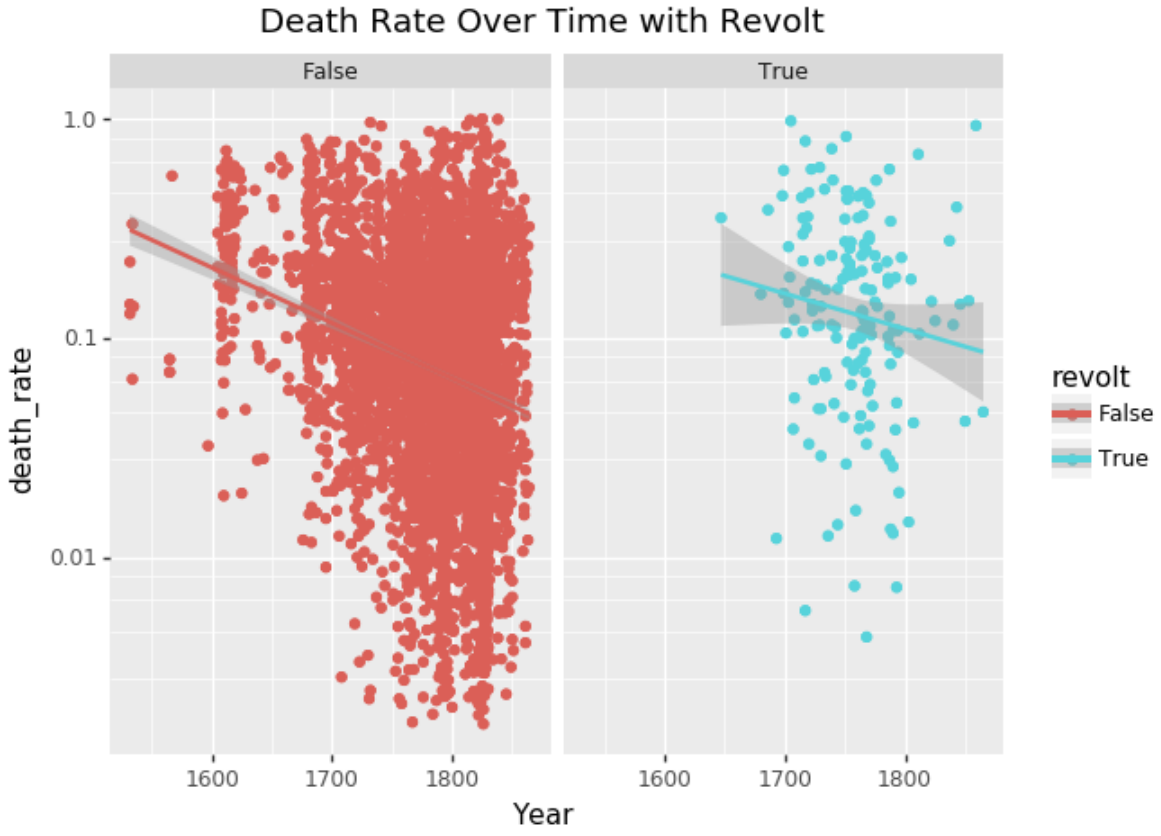


Out[15]: <ggplot: (8761608888328)>

6.5

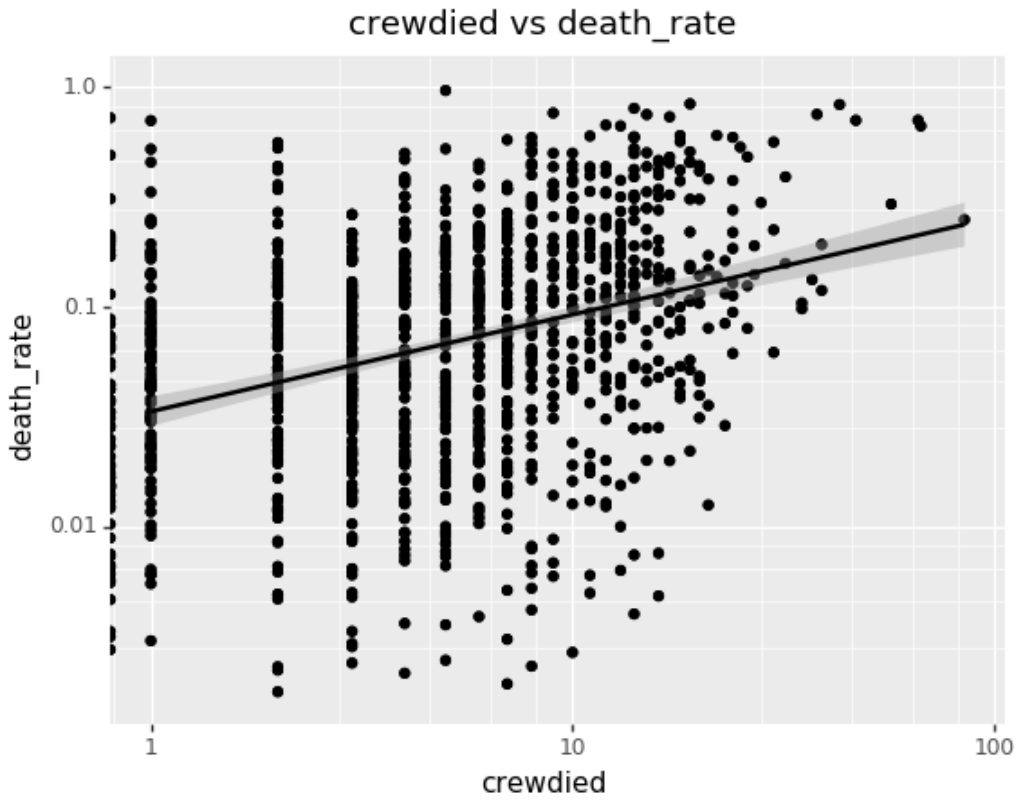
The two predictor variables I choose is `crewdied` and `revolt`, which is created in **Exercise 4.2**.

In [16]:
```
#6.5.1
#plot plot death rate as function of year with revolt
p9.ggplot(pop1,p9.aes("yearam","death_rate",color="revolt"))+p9.geom_point()+p9.geo
    +p9.facet_wrap("revolt")+p9.geom_smooth()+p9.scale_y_log10()\
    +p9.labs(x="Year",title="Death Rate Over Time with Revolt")
```



Out[16]: <ggplot: (-9223363275245835176)>

In [17]:
```
#6.5.2
#plot crewdied vs death_rate with log transformations
p9.ggplot(pop1,p9.aes("crewdied","death_rate"))+p9.geom_point()+p9.geom_point()\
    +p9.scale_x_log10()+p9.geom_smooth()+p9.scale_y_log10()+p9.labs(title="crewdied
```



Out[17]: <ggplot: (8761611202231)>

6.6

For the above plot, I used
geoms:point, line
aesthetic mapping: x-position y-position, color
scale: scale_x_log10, scale_y_log10, others: labs,facet_wrap

Exercise 7. Estimating total captives count II.

1. Using the variables selected, fit a prediction of death rate using linear regression and some simple transformations/scales. You do not need to spend time doing automated model selection, just include the most likely predictor or predictors from the previous question.
2. We will impute the number of captives on board at departure, so enumerate the cases for missingness of the variables involved in the prediction.
3. We would like to predict the number of captives on departure from the number that arrived with the following formula,
$$\text{arrived} = \pi(1 - \text{death rate}) \cdot (\text{captives taken}) + (1 - \pi)(\text{average increase})$$
where π is the proportion of voyages with no increase in captives, and the average increase is over those that did see an increase. For those voyages with captives taken missing and arrived not missing, use the estimated death rate to predict the captives taken.
4. Impute the remainder by predicting the captives taken with the year variable, you can use simple linear regression.
5. With the same formula and method impute the number of captives upon arrival whenever it is missing.

6. Give a new estimate of the predicted total number of captives taken on the middle passage, the total number of deaths, and the overall death rate. How does this differ from your first estimate obtained by assuming MCAR?

7.1

I fit a linear model for death rate using `yearam` , `revolt` . Before fitting I did log transformation on `death_rate` to make it more linear.

In [18]:
```
#7.1 fit a linear model for death rate
from sklearn.linear_model import LinearRegression
train_data=pop1.copy()
train_data.loc[train_data["death_rate"]==0,"death_rate"]=0.000001
train_data["log_death_rate"]=np.log(train_data["death_rate"])
lm_death = LinearRegression()
lm_death.fit(train_data[["revolt","yearam"]],train_data["log_death_rate"])
```

Out[18]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [19]:
```
#7.2
print("Number of missing values for yearam is {}".format(data.yearam.isna().sum()))
print("Number of missing values for revolt is {}".format(data.revolt.isna().sum()))
```

```
Number of missing values for yearam is 0
Number of missing values for revolt is 0
```

In [20]:
```
#7.3
#get data for calculating pi and average increase
estCap_data=data[data["slaarriv"].notnull()].copy()
estCap_data["death_rate"]=(estCap_data["tslavesd"]-estCap_data["slaarriv"])/estCap_

#calcalate pi and average increase
pi=sum(estCap_data["slaarriv"]<=estCap_data["tslavesd"])/estCap_data["death_rate"].
increasing=estCap_data[estCap_data["slaarriv"]>estCap_data["tslavesd"]]
ave_increase=(increasing["slaarriv"]-increasing["tslavesd"]).sum()/len(increasing.i

#predict captive taken
taken_missing=estCap_data[estCap_data["tslavesd"].isna()].copy()

taken_missing["death_rate"]=np.exp(lm_death.predict(taken_missing[["revolt","yearam

taken_missing["tslavesd"]=(taken_missing["slaarriv"]-(1-pi)*ave_increase)/(pi*(1-ta
```

In [21]:
```
#7.4
#get data for predict
taken_exist=data[data["tslavesd"].notnull()]
remain_data=data[data["slaarriv"].isna()&data["tslavesd"].isna()].copy()

#fit a linear model and predict
lm_remain = LinearRegression()
lm_remain.fit(taken_exist[["yearam"]],taken_exist["tslavesd"])
remain_data["tslavesd"]=lm_remain.predict(remain_data[["yearam"]])
```

In [22]:
```
#7.5
#get data for predict
arrival_exist=data[data["slaarriv"].notnull()]
arrival_missing=data[data["slaarriv"].isna()].copy()

#fit a linear model and predict
lm_arrival = LinearRegression()
lm_arrival.fit(arrival_exist[["yearam"]],arrival_exist["slaarriv"])
arrival_missing["slaarriv"]=lm_arrival.predict(arrival_missing[["yearam"]])
```

7.6

The estimated total number of captives 11027897.0
The estimated total number of death 1534948.0
The estimated overall death rate 0.1391877013580286
Comparing to the first estimate of total captives, this estimate is smaller.

In [23]:
```
#7.6
# get the data the both "tslavesd" and "slaarriv" exists
both_exist=data[data["tslavesd"].notnull()&data["slaarriv"].notnull()]
#calculate required values
total_captive=both_exist["tslavesd"].sum()+remain_data["tslavesd"].sum()+arrival_mi
              +taken_missing["tslavesd"].sum()
total_arrival=both_exist["slaarriv"].sum()+remain_data["slaarriv"].sum()+arrival_mi
              +taken_missing["slaarriv"].sum()
total_death=total_captive-total_arrival
death_rate=total_death/total_captive

print("The total number of captive {}".format(round(total_captive)))
print("The total number of death {}".format(round(total_death)))
print("The overall death rate {}".format(death_rate))
```

```
The total number of captive 11027897.0
The total number of death 1534948.0
The overall death rate 0.1391877013580286
```