**2022-2023学年第二学期《大数据技术基础》考查报告**

**课题名称：2020年美国新冠肺炎疫情数据分析**

班 级： 21大数据2班 啊

学 号： 2115905027 啊

姓 名： 何东海

成 绩： 啊

2022年12月

目录

[一、大数据平台的搭建](#_Toc9803_WPSOffice_Level1) [1](#_Toc9803_WPSOffice_Level1)

[1. 架构设计](#_Toc28329_WPSOffice_Level2) [1](#_Toc28329_WPSOffice_Level2)

[2 服务器集群的搭建](#_Toc4653_WPSOffice_Level2) [1](#_Toc4653_WPSOffice_Level2)

[（1）安装Linux操作系统、](#_Toc3321_WPSOffice_Level3) [1](#_Toc3321_WPSOffice_Level3)

[（2）安装大数据处理框架Hadoop 2](#_Toc6907_WPSOffice_Level3)

[3 工具软件安装 3](#_Toc10868_WPSOffice_Level2)

[（1）安装关系型数据库MySQL 3](#_Toc29052_WPSOffice_Level3)

[（2）安装列族数据库HBase 4](#_Toc5697_WPSOffice_Level3)

[（3）安装Zookeepper 4](#_Toc27593_WPSOffice_Level3)

[（4）安装数据仓库Hive 5](#_Toc19290_WPSOffice_Level3)

[（5）安装Sqoop 5](#_Toc18713_WPSOffice_Level3)

[（6）安装Eclipse 5](#_Toc953_WPSOffice_Level3)

[二、数据导入 6](#_Toc23725_WPSOffice_Level1)

[1. 数据库、表清单 6](#_Toc2564_WPSOffice_Level2)

[2. 数据对象的定义 7](#_Toc8540_WPSOffice_Level2)

[（1） 创建数据库 7](#_Toc25966_WPSOffice_Level3)

[（2） 创建数据表 8](#_Toc598_WPSOffice_Level3)

[2. 数据集处理与导入 8](#_Toc6644_WPSOffice_Level2)

[（1） 数据集的预处理 8](#_Toc19213_WPSOffice_Level3)

[（2） 数据集导入 9](#_Toc18199_WPSOffice_Level3)

[3. 数据分析](#_Toc32621_WPSOffice_Level2) [1](#_Toc32621_WPSOffice_Level2)1

[（1） 简单查询分析](#_Toc444_WPSOffice_Level3) [1](#_Toc444_WPSOffice_Level3)1

[（2） 查询条数统计分析](#_Toc9929_WPSOffice_Level3) [1](#_Toc9929_WPSOffice_Level3)1

[（3） 关键字条件查询分析](#_Toc27113_WPSOffice_Level3) [1](#_Toc27113_WPSOffice_Level3)2

[（4） 根据用户行为分析](#_Toc10014_WPSOffice_Level3) [1](#_Toc10014_WPSOffice_Level3)2

[（5） 用户实时查询分析](#_Toc17261_WPSOffice_Level3) [1](#_Toc17261_WPSOffice_Level3)3

[三、 数据可视化展示](#_Toc31655_WPSOffice_Level1) [1](#_Toc31655_WPSOffice_Level1)3

[四、个人总结 1](#_Toc10453_WPSOffice_Level1)6

**一、大数据平台的搭建**

**1. 架构设计**

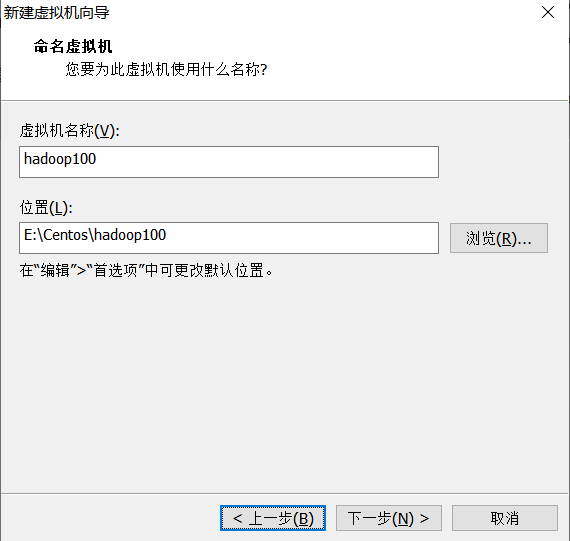
此处使用VMware12pro创建虚拟机（使用VMware16时有些操作会导致系统崩溃重启，个人认为12版本更为稳定一些），安装的操作系统为Centos7.5，之前使用为ubuntu18.04，其主要差别在于ubuntu18.04的图形化界面较为完善，但完成课设主要进行的操作基本都在终端上靠命令完成，所以此处使用Centos进行操作。网络使用NAT模式将其ip配置为192.168.8.0同时更改NAT中网关为192.168.8.2，此处同时更改win10中VMnet8中的IP地址、网关、DNS。使用xshell连接linux服务器进行操作，连接端口号默认为22。其数据存储位置在HDFS上，具体位置为 /opt/module/hadoop-3.1.3/data，以及mysql的bigdata数据库中。

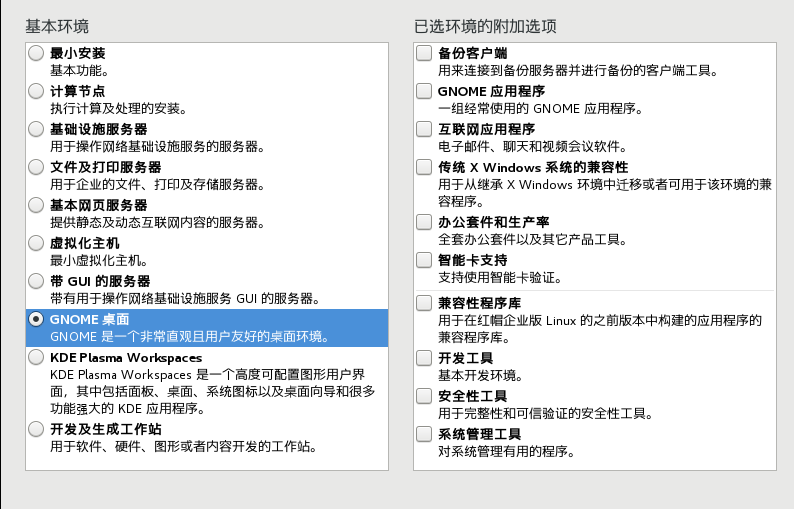


**2 服务器集群的搭建**

1. 安装Linux操作系统

提前在网上下载好centos7.x光盘映像文件，在VMware中创建新的虚拟机，选择自定义安装将虚拟机命名为hadoop100，放在E盘的文件夹下为其分配4G的内存和两个处理器，以及50G最大可用空间。完成配置后选择光盘映像文件，然后打开虚拟机进行安装。进入安装后此处选择桌面版安装。



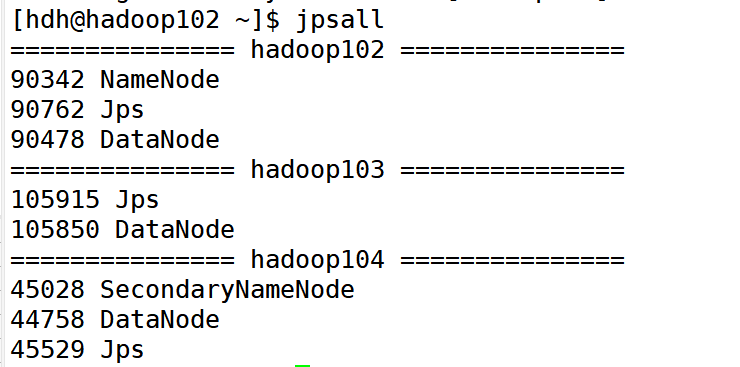


1. 安装大数据处理框架Hadoop
2. 在网上找到hadoop安装压缩包，此处使用hadoop-3.1.3版本，使用Bitvise SSH Client 连接虚拟机将其传入到hadoop102中/opt/software目录（以下简称software目录）下，此处将所有需要用到的软件传入到此目录下，以下省略此步骤。
3. 将jdk-8u212-linux-x64.tar.gz、hadoop-3.1.3.tar.gz解压到/opt/module目录（以下简称module目录）。
4. 在/etc/profile.d/my\_env.sh文件中配置环境变量，此处区别于系统自带的环境变量配置文件，使用自己新建的脚本文件配置环境变量。
5. 配置ssh免密登录，此处是为了以后在hadoop102上访问其他服务器更加方便。
6. 更改hadoop目录下etc/hadoop中的core-site.xml、hdfs-site.xml、yarn-site.xml、

mapred-site.xml、workers文件中的配置

6、使用编写好的脚本attri分发已经安装好的hadoop和jdk（脚本主要使用rsync进行在不同服务器之间的数据传输）

7、使用start-dfs.sh启动hadoop，然后使用编写好的shell脚本文件jpsall（使用ssh连接各个服务器同时使用jps命令查看进程即可）查看个服务器上的进程，出现如下进程显示，则说明安装配置成功。



**3 工具软件安装**

1. 安装关系型数据库MySQL
2. 首先使用rpm -qa|grep mariadb命令检查系统是否曾安装过数据库，如有则使用sudo rpm -e --nodeps mariadb-libs命令进行卸载
3. 使用tar -xf mysql-5.7.28-1.el7.x86\_64.rpm-bundle.tar命令解压mysql压缩包
4. sudo rpm -ivh mysql-community-common-5.7.28-1.el7.x86\_64.rpm

sudo rpm -ivh mysql-community-libs-5.7.28-1.el7.x86\_64.rpm

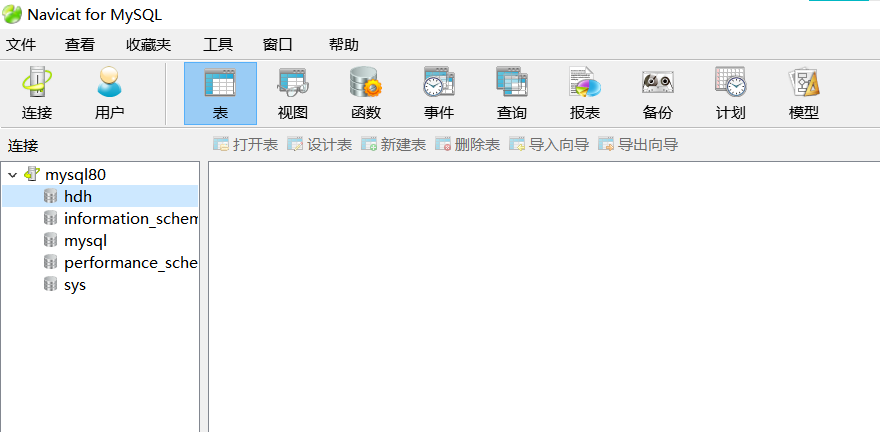
sudo rpm -ivh mysql-community-libs-compat-5.7.28-1.el7.x86\_64.rpm

sudo rpm -ivh mysql-community-client-5.7.28-1.el7.x86\_64.rpm

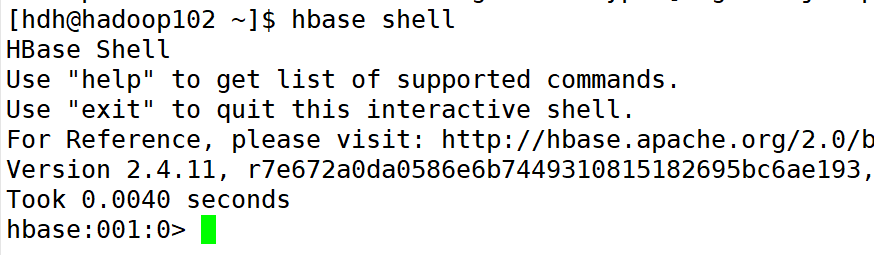
sudo rpm -ivh mysql-community-server-5.7.28-1.el7.x86\_64.rpm

分别安装以上安装包

1. sudo mysqld --initialize --user=mysql 初始化数据库
2. sudo cat /var/log/mysqld.log查看生成的root用户的临时密码
3. start mysqld 启动mysql服务
4. set password = password("root"); 进入mysql修改root密码
5. update mysql.user set host='%' where user='root'; 修改 mysql 库下的 user 表中的 root 用户允许任意 ip 连接
6. flush privileges; 刷新权限
7. 能够使用Navicat连接上说明已经配置好了



1. 安装列族数据库HBase
2. tar -zxvf hbase-2.4.11-bin.tar.gz -C /opt/module/ 将hbase解压到module目录上
3. mv /opt/module/hbase-2.4.11 /opt/module/hbase 重命名
4. sudo vim /etc/profile.d/my\_env.sh 将hbase路径添加到环境变量
5. 更改hbase/conf目录下的hbase-env.sh、hbase-site.xml、regionservers
6. 使用attri脚本分发到hadoop103、104上
7. start-hbase.sh 启动hbase服务
8. hbase shell 进入hbase命令行界面



（3）安装Zookeepper

1、然后使用tar命令将其解压安装到/opt/module目录下，

2、安装完成后将其重命名为zookeeper，

3、进入zookeeper目录创建myData目录，在myData目录下创建myid 文件，在文件中添加与 server 对应的编号2

4、重命名/opt/module/zookeeper-3.5.7/conf 这个目录下的 zoo\_sample.cfg 为 zoo.cfg，修改数据存储位置dataDir=/opt/module/zookeeper/zkData，并添加相关配置后分发到其他服务器，

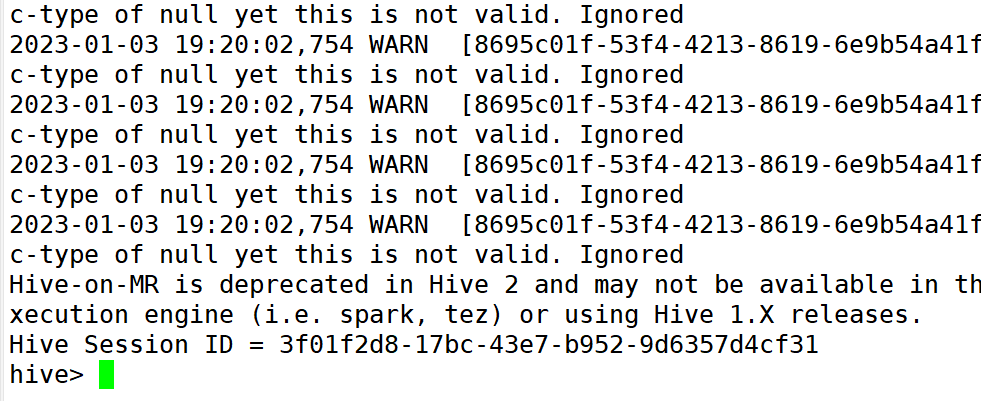
5、更改每台服务器中myid文件内容

6、使用bin/zkServer.sh start 将其启动，jps查看进程，看到此进程则说明zookeeper启动成功



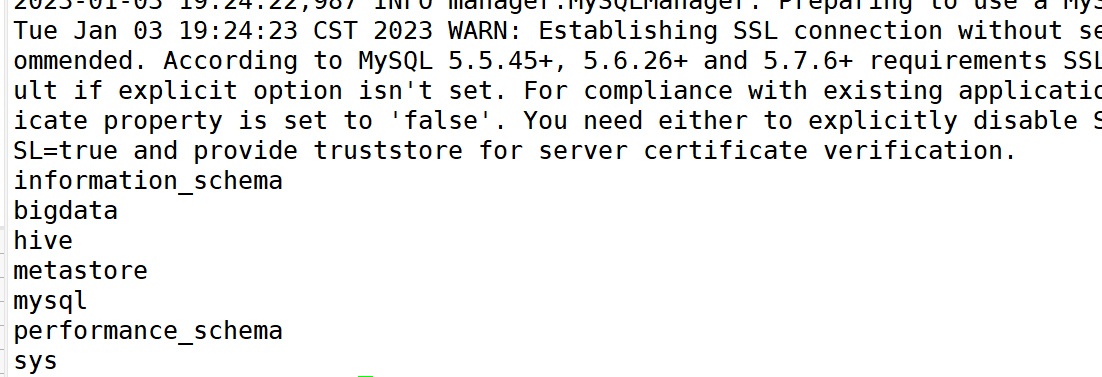
（4）安装数据仓库Hive

1. 解压apache-hive-3.1.2-bin.tar.gz到/opt/module/目录下面
2. 修改apache-hive-3.1.2-bin.tar.gz的名称为hive
3. 修改/etc/profile.d/my\_env.sh，添加环境变量
4. 将MySQL的JDBC驱动拷贝到Hive的lib目录下
5. 在hive的conf目录下新建hive-site.xml文件，添加配置
6. hive 启动hive命令行，进入此页面则安装成功



（5）安装Sqoop

1. 解压sqoop安装包到/opt/module 目录
2. 重命名该文件夹为sqoop
3. mv sqoop-env-template.sh sqoop-env.sh 进入到/opt/module/sqoop/conf目录，重命名配置该文件
4. 修改配置文件sqoop-env.sh
5. 进入到/opt/software/路径，拷贝jdbc驱动（mysql-connector-java-5.1.48.jar ）到sqoop的lib目录下。
6. bin/sqoop list-databases --connect jdbc:mysql://hadoop102:3306/ --username root --password root 使用该命令看能否连接到mysql数据库，出现数据库中的数据库名称则说明能够正常连接



1. 安装Eclipse
2. tar -zxvf ./eclipse-4.7.0-linux.gtk.x86\_64.tar.gz -C /opt/module 将eclipse解压到module目录
3. 进入到该目录使用该命令./eclipse 能正常启动其图形化界面即可

**二、数据导入**

**1. 数据库、表清单**

此部分主要介绍数据库所有数据表的清单，包括表的类型、表的名称和中文含义

1. mysql

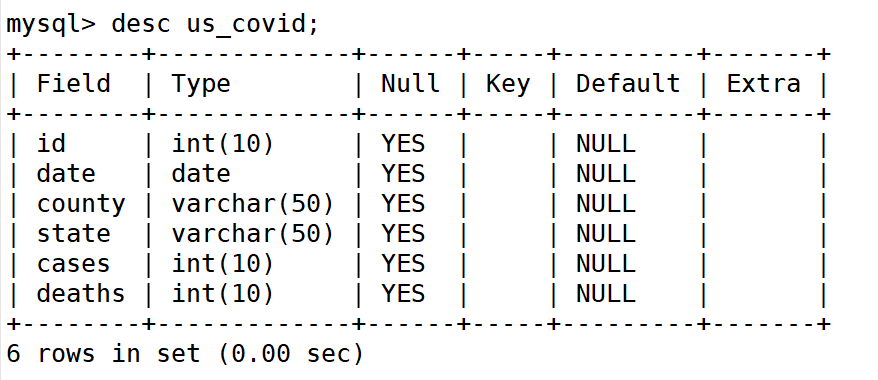
数据库：

information\_schema、bigdata、hive、mysql、performance\_schema、sys

此处只有bigdata是个人创建用于存放数据的

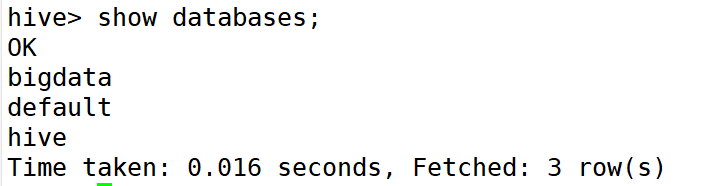
bigdata中的表格：us\_covid

存放的数据包括id、日期、城市、州名、确诊人数、死亡人数



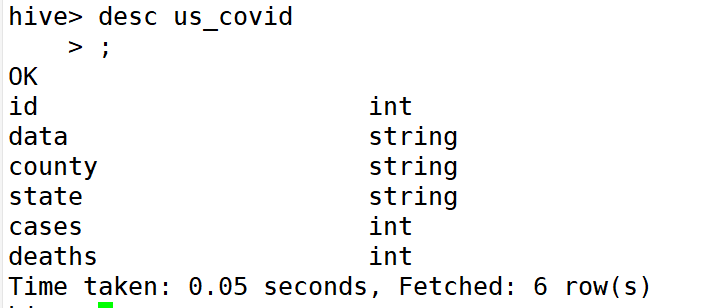
1. hive

数据库：



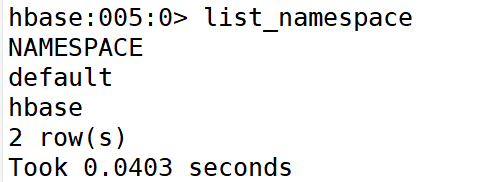
hive中只有默认数据库和创建的bigdata数据库

表格：



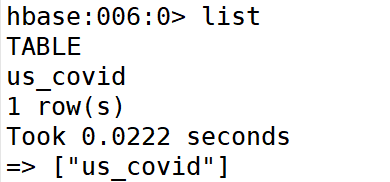
1. hbase

数据库：



hbase中包含两个命名空间（类似mysql中数据库），其中直接创建的表格会存储在default命名空间中。

表格：



hbase中只有一个us\_covid表格，其拥有一个列族，列名与mysql中的us\_covid一样，但没有id，取而代之的是行号。

1. **数据对象的定义**
2. 创建数据库
3. mysql

create database bigdata; 创建bigdata数据库

1. hive

create database bigdata; 创建bigdata数据库

1. hbase

未创建新命名空间，数据存储在default命名空间中

1. 创建数据表；
2. mysql

create database dblab; 创建dblab数据库

1. hive

create database dblab; 创建dblab数据库

1. hbase

未创建新命名空间，数据存储在default命名空间中

1. 创建数据表；
2. mysql

create table us\_covid(id int(10),date date,county varchar(50),state varchar(50),cases int(10),deaths int(10));

在mysql的bigdata数据库中创建us\_covid表格，包括id、日期、城市、州名、确诊病例、死亡人数

1. hive

CREATE EXTERNAL TABLE dblab.bigdata\_user(id INT,uid STRING,item\_id STRING,behavior\_type INT,item\_category STRING,visit\_date DATE,province STRING) COMMENT 'Welcome to xmu dblab!' ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' STORED AS TEXTFILE LOCATION '/bigdata';

创建外部表，其数据来源于hdfs中的/bigdatat目录下

1. hbase

create 'us\_covid', { NAME => 'f1', VERSIONS => 5}

创建表格us\_covid，创建一个列族‘f1’，保留5个版本

1. **数据集处理与导入**
2. 数据集的预处理

数据集较为规则，只需要删除其第一行，并简单编写shell脚本，将其转换为txt格式

1. 数据集导入

在hive中创建bigdata数据库，进入后使用以下代码将数据从hdfs中导入到hive中 的us\_covid表格中：

hive> create external table bigdata.us\_covid(id int,data string,county string,state string,cases int,deaths int) comment 'welcome to bigdata!' row format delimited fields terminated by '\t' stored as textfile location '/bigdata';

在mysql中创建bigdata数据库，并在此数据库中使用下列代码创建表格us\_covid：

mysql> create table us\_covid(id int(10),date date,county varchar(50),state varchar(50),cases int(10),deaths int(10));

使用sqoop将数据从hive导入到mysql

./bin/sqoop export --connect jdbc:mysql://localhost:3306/bigdata --username root --password root --table us\_covid --export-dir '/user/hive/warehouse/bigdata.db/user\_action' --fields-terminated-by '\t';

使用idea编写程序连接hbase将数据导入到hbase中

*package* com.hdh;

*import* org.apache.hadoop.hbase.TableName;

*import* org.apache.hadoop.hbase.client.*Connection*;

*import* org.apache.hadoop.hbase.client.Put;

*import* org.apache.hadoop.hbase.client.*Table*;

*import* org.apache.hadoop.hbase.util.Bytes;

*import* java.io.BufferedReader;

*import* java.io.FileReader;

*import* java.io.IOException;

*import* java.util.ArrayList;

*import* java.util.*List*;

*public class* HBaseDML {

*public static Connection connection* = HbaseConnection.*connection*;

*public static void* putCell(String namespace, String tablename, String rowkey, String columnf,

String columnn, String value) *throws* IOException {

*Table* table = *connection*.getTable(TableName.*valueOf*(namespace, tablename));

Put put = *new* Put(Bytes.*toBytes*(rowkey));

put.addColumn(Bytes.*toBytes*(columnf),Bytes.*toBytes*(columnn), Bytes.*toBytes*(value));

table.put(put);

table.close();

}

*public static void* main(String[] args) *throws* IOException {

String File = "E:\\hadoop\\考查作业\\2020年美国新冠肺炎疫情数据分析\\us-counties.csv";

String line = "";

String SplitBy = ",";

String[] Line;

*List*<String[]> BikeDataList = *new* ArrayList<>();

*int* i =1;

*try* (BufferedReader br = *new* BufferedReader(*new* FileReader(File))) {

*while* ((line = br.readLine()) != *null*) {

Line = line.split(SplitBy);

BikeDataList.add(Line);

*putCell*("default", "us\_covid", String.*valueOf*(i), "info",

"date", Line[0]);

*putCell*("default", "us\_covid", String.*valueOf*(i), "info",

"county", Line[1]);

*putCell*("default", "us\_covid", String.*valueOf*(i), "info",

"state", Line[2]);

*putCell*("default", "us\_covid", String.*valueOf*(i), "info",

"cases", Line[3]);

*putCell*("default", "us\_covid", String.*valueOf*(i), "info",

"deaths", Line[4]);

i++;

}

} *catch* (IOException e) {

e.printStackTrace();

}

System.out.println("导入完成！");

}

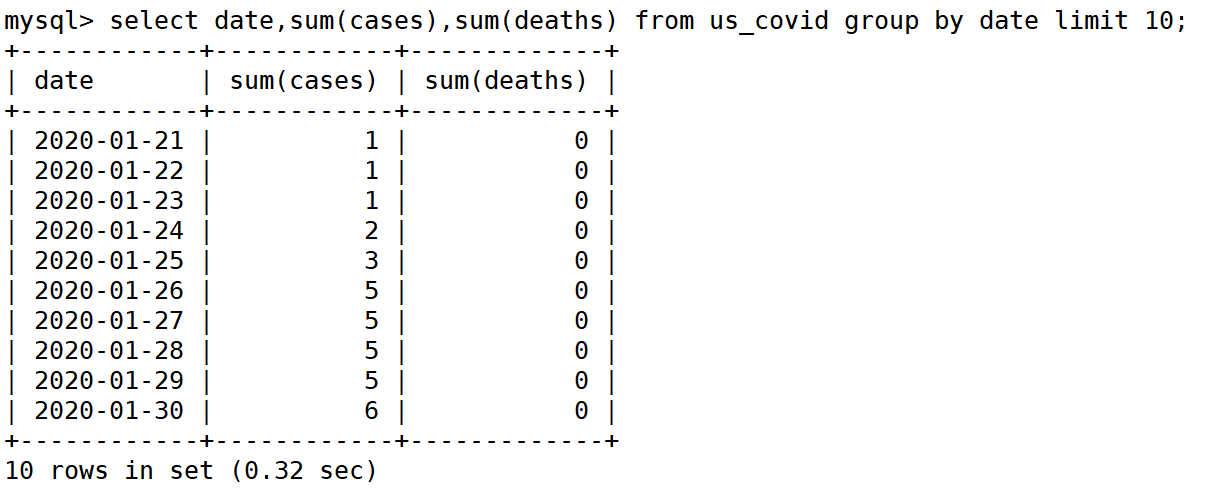
}

1. **数据分析**

（此处均使用查询效率较高的mysql对数据进行分析）

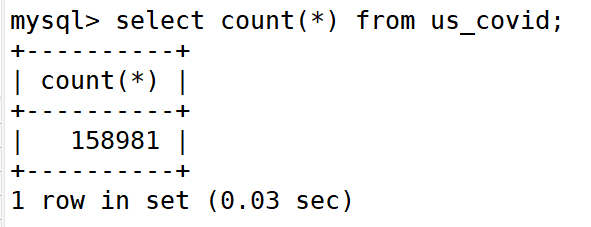
1. 简单查询分析

select date,sum(cases),sum(deaths) from us\_covid group by date limit 10; 查看前十天确诊和死亡人数

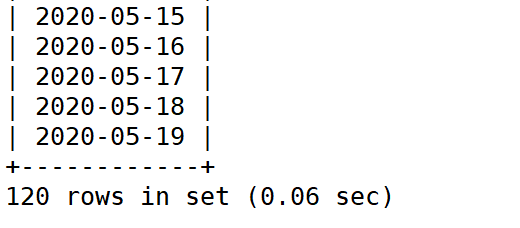


1. 查询条数统计分析

1、select count(\*) from us\_covid; 查看表格中一共有多少条数据



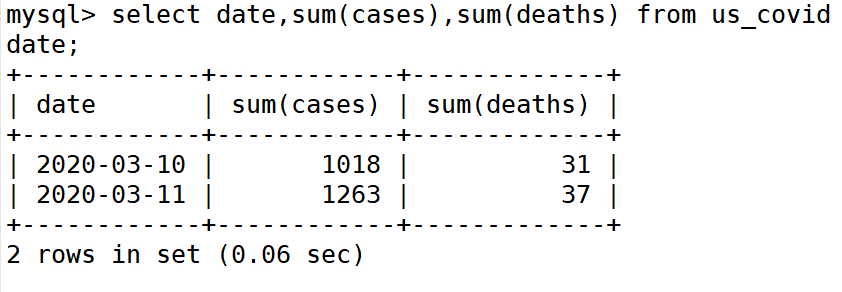
2、select date from us\_covid; 查看表格共统计了多少天的数据



1. 关键字条件查询分析

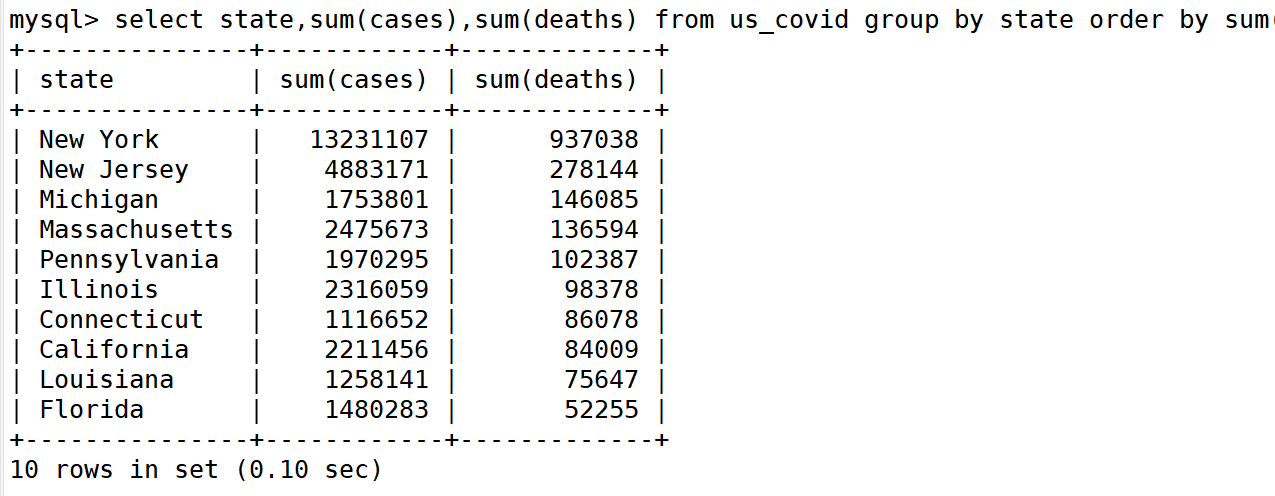
1、select date,sum(cases),sum(deaths) from us\_covid where date >'2020-03-09' and date<'2020-03-12' group by date;

查看2020年3月9号到3月12号的确诊和死亡人数



2、select state,sum(cases),sum(deaths) from us\_covid group by state order by sum(deaths) desc limit 10;

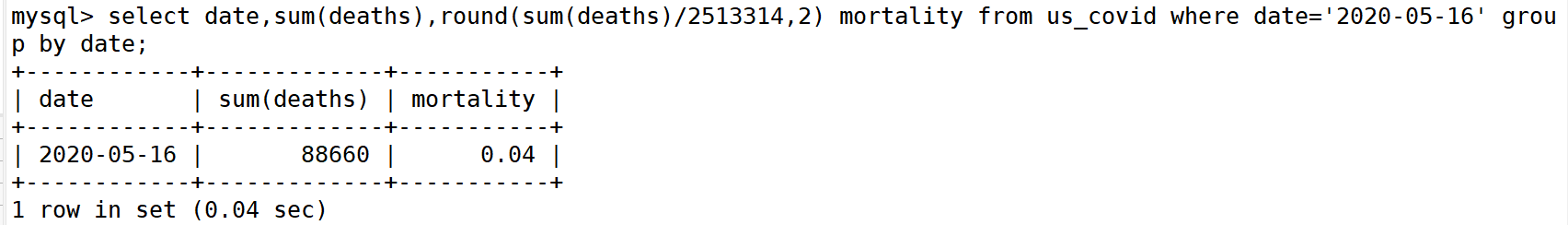
查看死亡人数最多的十个州



1. 根据用户行为分析

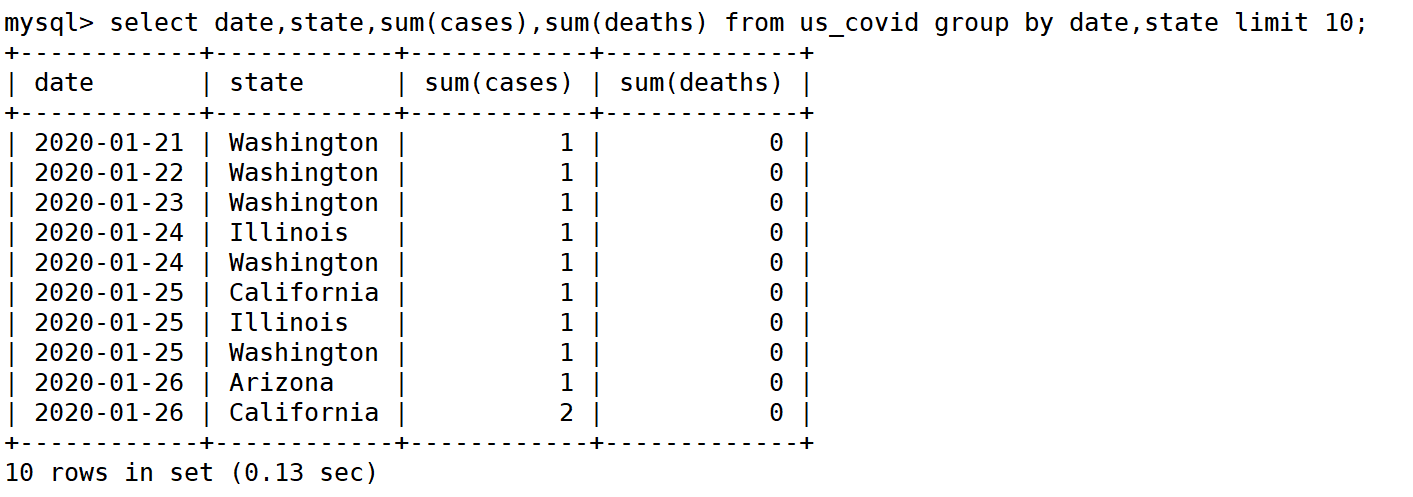
select date,sum(deaths),round(sum(deaths)/2513314,2) mortality from us\_covid where date='2020-05-16' group p by date;

查看2020年5月16日死亡人数占总死亡人数的比例



1. 用户实时查询分析

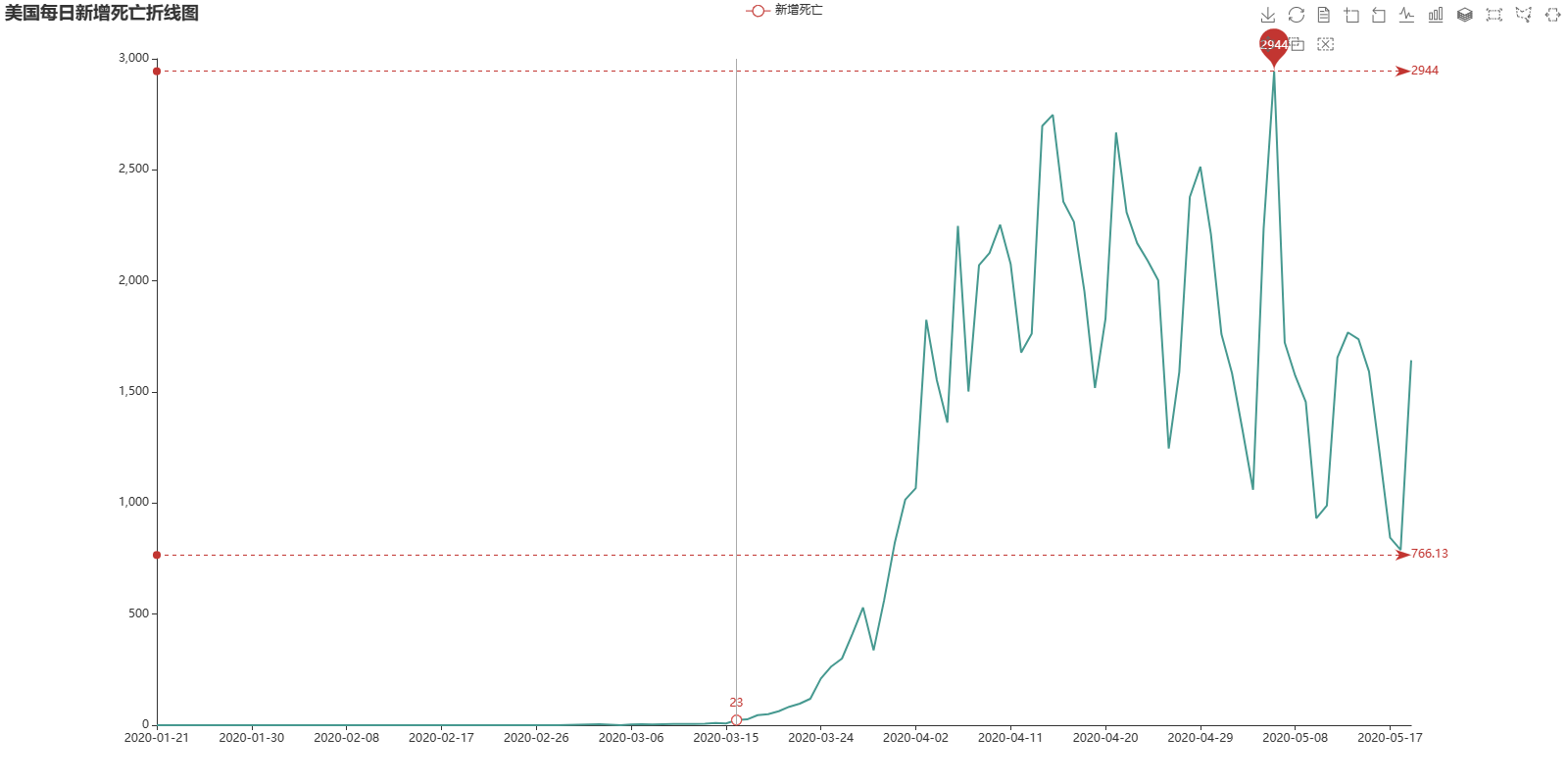
查看每一天每个州的确诊和死亡人数



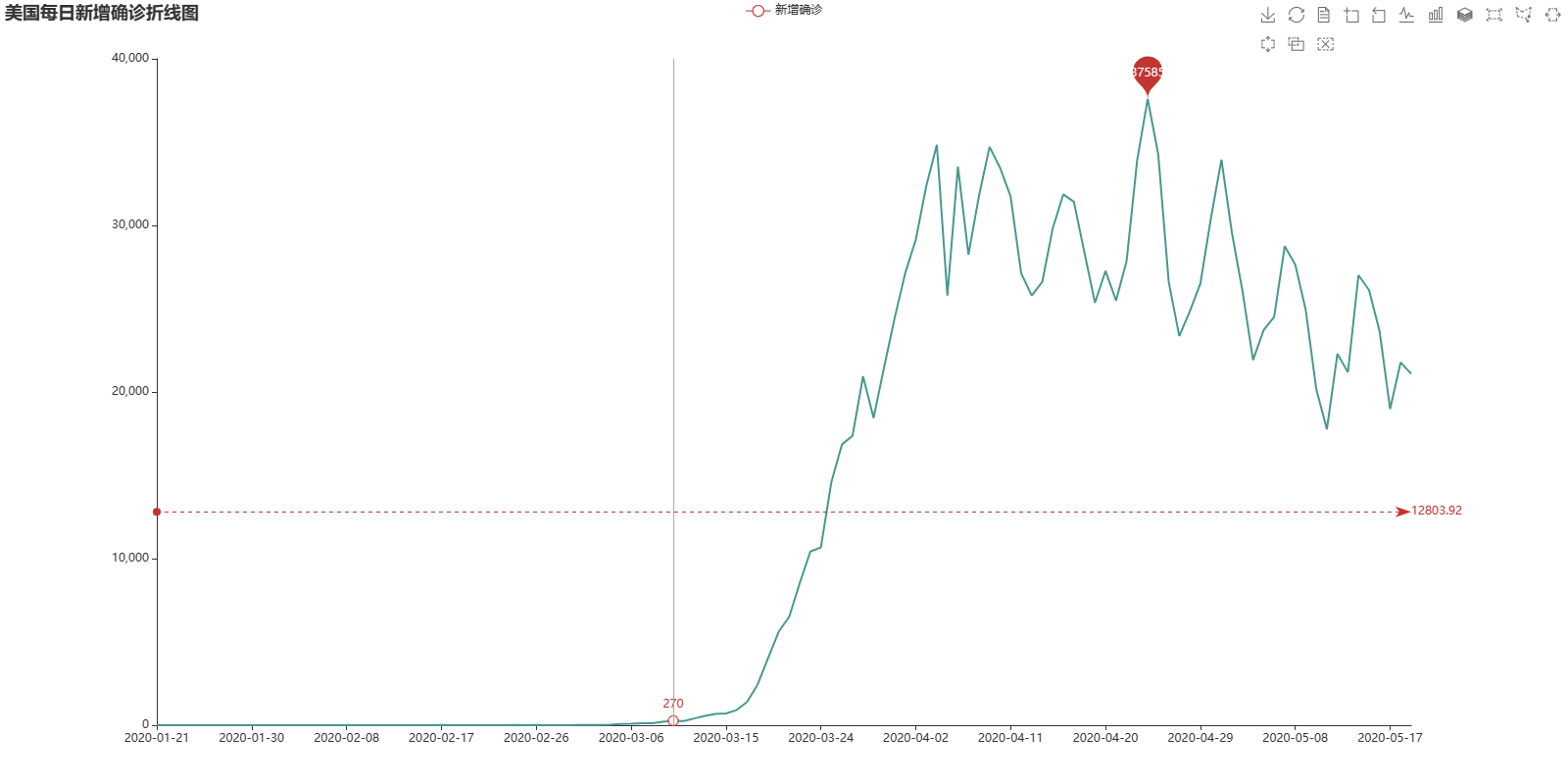
1. **数据可视化展示**

MySQL中的数据进行可视化分析。柱状图可视化分析、散点图可视化分析、地图可视化分析

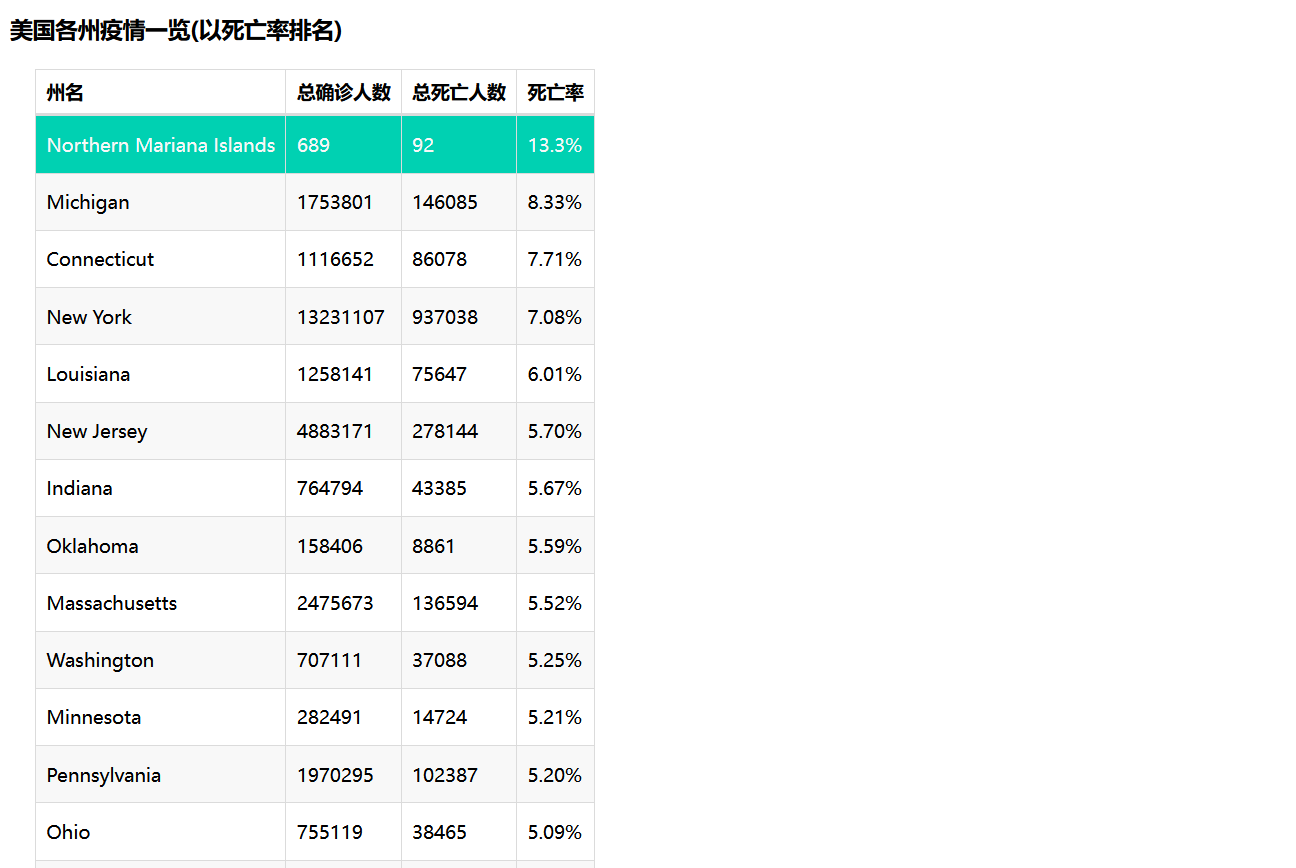
1. 每日新增死亡人数折线图



1. 每日新增确诊折线图



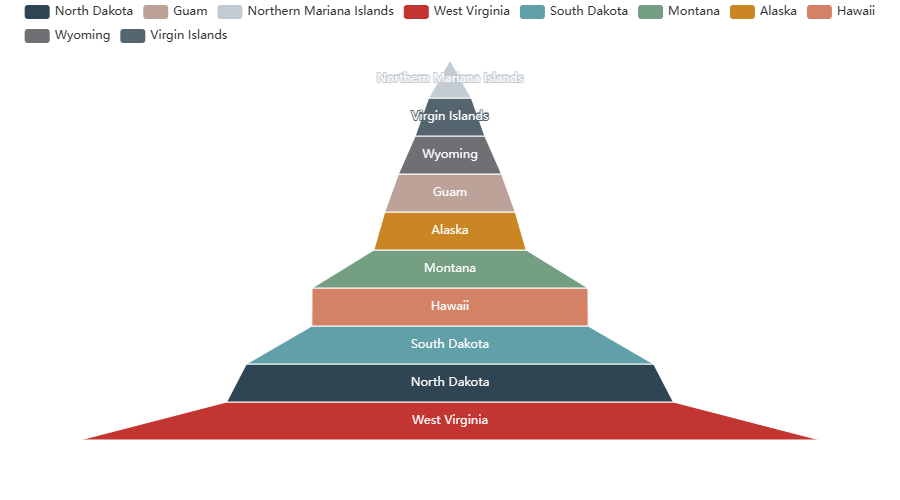
1. 死亡率表格



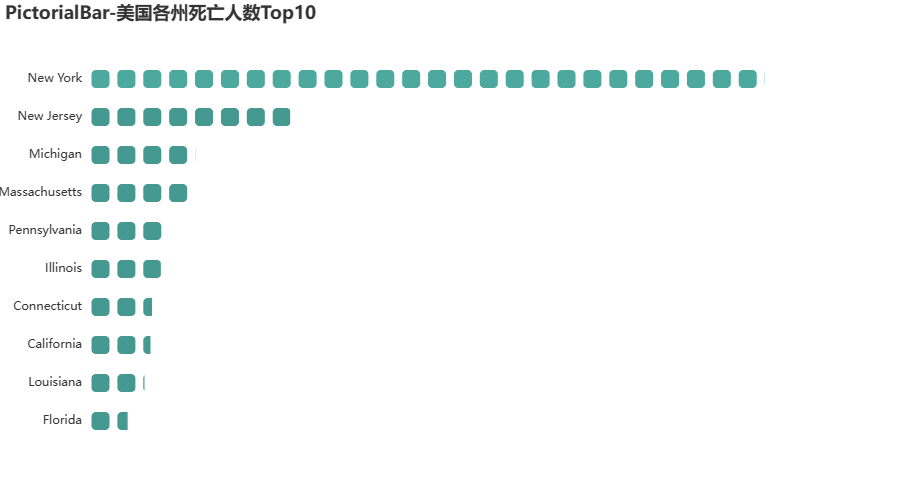
1. 确诊最多词云图



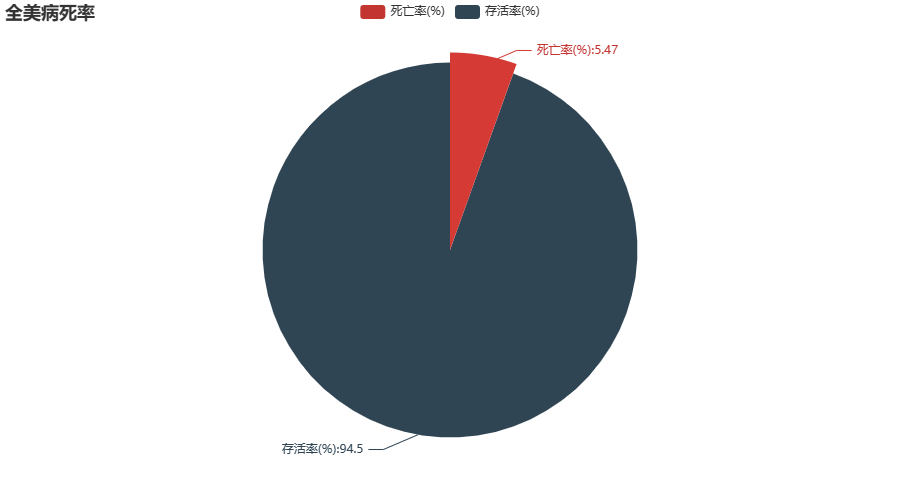
1. 美国死亡最少的10个州--漏斗图



1. 美国死亡最多的10个州横向柱状图



1. 死亡率饼状图



**四、个人总结**

本次实验采用完全分布式进行，部署三台服务器，使用xshell连接大部分实验内容在终端上完成，使用了pycharm、idea等工具完成对数据的处理、分析

、可视化，总体来说收获很大。

由于之前做过网站用户行为分析实验，所以本次的实验并不算太过困难，在安装软件以及搭建hadoop环境的过程中当然也遇到了很多问题，但是通过自己在网上查找各种资料以及凭借个人的知识经验积累，成功的将这些问题都一一克服，或是解决当前问题，或是使用另一种方法迂回的解决出现的问题。在将数据导入到hbase中时，原本打算仿照网站用户行为分析实验使用sqoop将数据从mysql中导入到hbase，但出现很多问题，在网上寻找了很多解决方法，依旧无法解决问题，于是决定使用idea创建maven工程，编写java代码，使用org.apache.hadoop下的各种包连接虚拟机中的hbase数据库，然后读取.csv文件中的数据并通过java代码将其导入到hbase数据库中。在对数据进行处理并进行可视化时，借鉴了网上通过spark对美国新冠疫情做可视化分析的python代码，然后自己在pycharm中安装了pymysql、pyecharts等各种库，连接mysql数据库，通过sql查询语句得到想要的数据，然后使用python代码将其进一步处理为pyecharts中各种函数需要的参数类型，生成html文件将其保存在html目录下。本次实验让我更为深刻的理解了hadoop家族中各服务的作用及使用方法，对大数据这一概念也有了更好的认知。