

COVID-19 Data Analysis

Analyze by SQL and Tableau

SQL

```
1 -- Total cases vs Total deaths, DeathPercentage in China
2 SELECT location, date, total_cases, total_deaths, (total_deaths/total_cases) * 100 as DeathPercentage
3 FROM covid_deaths
4 WHERE location = 'China'
5 ORDER BY 1, 2
6
7 -- Total cases vs population_density, InfectionRate in China
8 SELECT location, date, total_cases, population, (total_deaths/total_cases) * 100 as InfectionRate
9 FROM covid_deaths
10 WHERE location = 'China'
11 ORDER BY 1, 2
12
13 -- Countries with highest infection rate compared to population
14 SELECT location, population, MAX(total_cases) as HighestInfectionCount, MAX((total_cases/population)) * 100 as
    PercentPopulationInfected
15 FROM covid_deaths
16 GROUP BY location, population
17 ORDER BY PercentPopulationInfected DESC
18
19 -- Countries with highest death count per population
20 SELECT location, population, MAX(total_deaths) as HighestDeathCount, MAX((total_deaths/population)) * 100 as
    PercentPopulationDeath
21 FROM covid_deaths
22 GROUP BY location, population
23 ORDER BY PercentPopulationDeath DESC
24
25 -- break things down by continent, continent with the highest death count
26 SELECT continent, MAX(total_deaths) as HighestDeathCount
27 FROM covid_deaths
28 WHERE continent is not null
29 GROUP BY continent
30 ORDER BY HighestDeathCount DESC
31
32 -- continents with highest death count per population
33 SELECT continent, MAX(total_deaths) as HighestDeathCount, MAX((total_deaths/population)) * 100 as
    PercentPopulationDeath
34 FROM covid_deaths
35 WHERE continent is not null
36 GROUP BY continent
37 ORDER BY PercentPopulationDeath DESC
```

```

39 -- GLOBAL numbers
40 SELECT SUM(new_cases) as total_cases, SUM(new_deaths) as total_deaths, SUM(new_deaths)/SUM(new_cases) * 100 as
    DeathPercentage
41 from covid_deaths
42 WHERE continent is not null
43 -- GROUP BY date
44 ORDER BY 1, 2 DESC
45
46 -- Total population vs new_vaccinations
47 SELECT cd.continent,cd.location, cd.date, cd.population,cv.new_vaccinations,
48     SUM(cv.new_vaccinations) OVER (PARTITION BY cd.location ORDER BY cd.location, cd.date) as RollingPeopleVaccinated
49 -- (RollingPeopleVaccinated/population) * 100
50 FROM covid_deaths cd
51 JOIN covid_vaccinations cv
52     ON cd.location = cv.location and cd.date = cv.date
53 WHERE cd.continent is not NULL
54 ORDER BY 2,3
55
56 --USE cte
57 with PopvsVac (Continent, Location,Date, Population, NewVaccinations,RollingPeopleVaccinated)
58 as
59 (
60 SELECT cd.continent,cd.location, cd.date, cd.population,cv.new_vaccinations,
61     SUM(cv.new_vaccinations) OVER (PARTITION BY cd.location ORDER BY cd.location, cd.date) as RollingPeopleVaccinated
62 -- (RollingPeopleVaccinated/population) * 100
63 FROM covid_deaths cd
64 JOIN covid_vaccinations cv
65     ON cd.location = cv.location and cd.date = cv.date
66 WHERE cd.continent is not NULL
67 ORDER BY 2,3
68 )
69 SELECT *,(RollingPeopleVaccinated/population) * 100
70 FROM PopvsVac
71

```

```

72 -- CREATE table
73 DROP TABLE IF EXISTS PercentPopulationVaccinated;
74 CREATE TABLE PercentPopulationVaccinated
75 (
76 Continent NVARCHAR(225),
77 Location NVARCHAR(225),
78 Date DATETIME,
79 Population NUMERIC,
80 New_vaccinations NUMERIC,
81 RollingPeopleVaccinated BIGINT
82 );
83 INSERT INTO PercentPopulationVaccinated
84 SELECT cd.continent,cd.location, cd.date, cd.population,cv.new_vaccinations,
85        SUM(cv.new_vaccinations) OVER (PARTITION BY cd.location ORDER BY cd.location, cd.date) as RollingPeopleVaccinated
86 -- (RollingPeopleVaccinated/population) * 100
87 FROM covid_deaths cd
88 JOIN covid_vaccinations cv
89     ON cd.location = cv.location and cd.date = cv.date;
90 -- WHERE cd.continent is not NULL
91 -- ORDER BY 2,3
92 SELECT *,(RollingPeopleVaccinated/population) * 100
93 FROM PercentPopulationVaccinated
94
95
96 -- CREATE VIEW TO STORE DATA FOR VISUALIZATION
97 DROP VIEW IF EXISTS PercentPopulationVaccinatedView;
98 CREATE VIEW PercentPopulationVaccinatedView AS
99 SELECT cd.continent,cd.location, cd.date, cd.population,cv.new_vaccinations,
100        SUM(cv.new_vaccinations) OVER (PARTITION BY cd.location ORDER BY cd.location, cd.date) as RollingPeopleVaccinated
101 -- (RollingPeopleVaccinated/population) * 100
102 FROM covid_deaths cd
103 JOIN covid_vaccinations cv
104     ON cd.location = cv.location and cd.date = cv.date;
105 WHERE cd.continent is not NULL
106 -- ORDER BY 2,3
107
108 SELECT *
109 FROM PercentPopulationVaccinatedView
110

```


Tableau- Dashboard

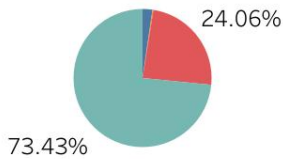
Base data

Average Hourly Arrival Rate	12.6
Average Daily Arrival Rate	297.9
Average Cumulative Hourly Number	47.7

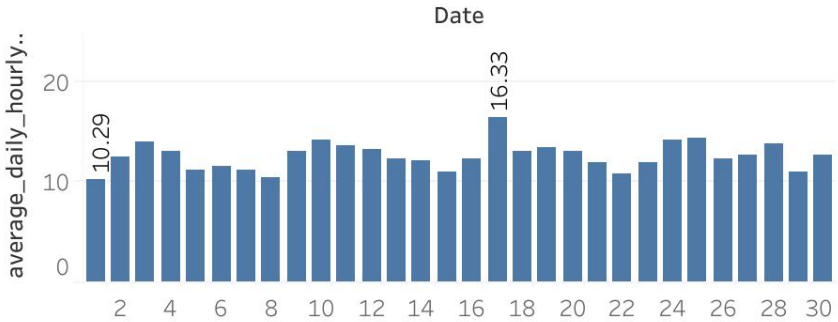
Arrival Mode

Arrival Method	
Walk In	6,562
SCDF Ambulance	2,150
Ambulance (Others)	206
Police Vehicle	19

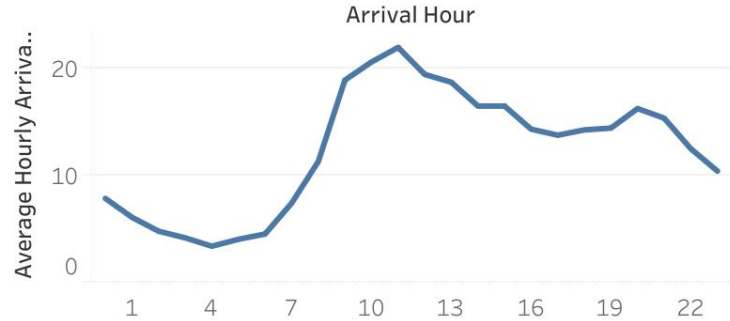
The Proportion of Arrival Mode



Daily Average Hourly Arrival Rates



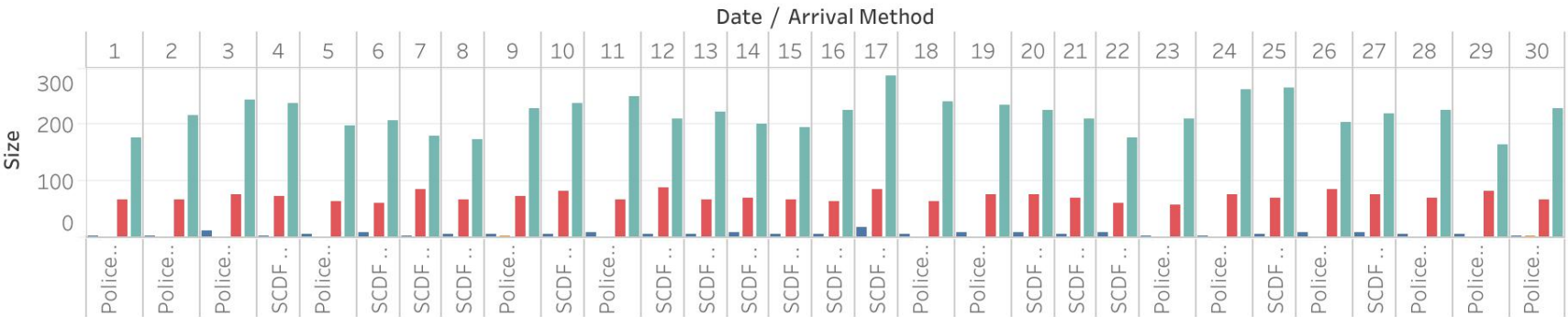
Distribution of Hourly Arrival Rates



Begin 2023 /9/1

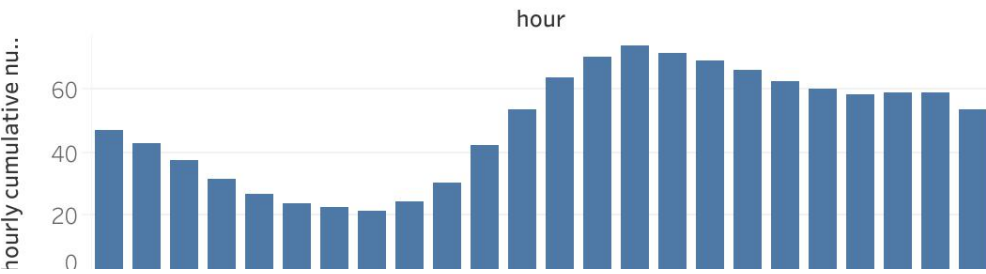
End 2023 9/1 2..

- Arrival Method
- ☒ Ambulance (Other..
 - ☒ Police Vehicle
 - ☒ SCDF Ambulance
 - ☒ Walk In



- Arrival Method
- ☒ Ambulance (Others)
 - ☒ Police Vehicle
 - ☒ SCDF Ambulance
 - ☒ Walk In

Cumulative Hourly Number of Each Hour



Daily Average Cumulative Hourly Number



- Arrival Method
- ☒ Ambulance (Others)
 - ☒ Police Vehicle
 - ☒ SCDF Ambulance
 - ☒ Walk In