Muertes por Sobredosis de Drogas en los Estados Unidos de Norteamérica

Sánchez Esteban, Padilla Jefferson, Varela Gabriel, Bustos Dixon.

[esteban.sanchez.sanchez@est.una.ac.cr](mailto:esteban.sanchez.sanchez@est.una.ac.cr), [jefferson.padilla.gonzalez@est.una.ac.cr](mailto:jefferson.padilla.gonzalez@est.una.ac.cr), [gabriel.varela.chacon@est.una.ac.cr](mailto:gabriel.varela.chacon@est.una.ac.cr), [dixon.bustos.medina@est.una.ac.cr](mailto:dixon.bustos.medina@est.una.ac.cr).

Universidad Nacional de Costa Rica.

***Abstract*- En las últimas décadas, ha surgido una preocupante epidemia de sobredosis de drogas en América del Norte, generando un complejo desafío de salud pública. Para abordar esta problemática, el lenguaje de programación Python ha emergido como una herramienta esencial en el análisis de datos relacionados con las sobredosis. Python facilita la recolección, procesamiento y visualización de datos, permitiendo la identificación de patrones y tendencias relevantes en un contexto epidemiológico. Además, su flexibilidad y poder computacional lo convierten en una herramienta versátil en diversas disciplinas de la Tecnología de la Información.**

I. INTRODUCCIÓN

En las últimas décadas, se ha presenciado una sombría epidemia en América del Norte: las muertes por sobredosis de drogas. Este devastador fenómeno ha dejado una profunda huella en la región, afectando a personas de todas las edades, géneros y orígenes étnicos. Más allá de las frías estadísticas, se esconden desgarradoras historias de lucha, adicción y pérdida que han conmovido a comunidades enteras.

Para comprender este grave problema de salud pública y abordarlo de manera efectiva, se ha considerado crucial desentrañar las causas subyacentes de la sobredosis y comprender las complejas ramificaciones que afectan a las personas y a sus seres queridos. Desde el aumento en la disponibilidad de opioides recetados hasta la creciente influencia de drogas ilícitas cada vez más potentes, el panorama de las sobredosis en América del Norte es sumamente complejo y requiere una respuesta integral.

En este análisis exhaustivo, no solo se examinará las tendencias y estadísticas más recientes relacionadas con las muertes por sobredosis, sino que también se explorará cómo la tecnología, en particular el lenguaje de programación Python, ha desempeñado un papel fundamental en el análisis de datos relacionados con esta crisis. Python, con su potencia y flexibilidad, se ha convertido en una herramienta esencial para investigadores y profesionales de la salud, permitiéndoles analizar y comprender a fondo las dimensiones de esta epidemia.

Desde la recopilación y visualización de datos hasta la identificación de patrones y tendencias, Python ha sido un aliado invaluable en la búsqueda de soluciones efectivas. Además, mientras se explora y analiza esta preocupante crisis de salud pública, se reflexionará sobre las implicaciones que tiene para la sociedad en general. A medida que se profundice en este tema, se hará evidente que la lucha contra las sobredosis de drogas es un desafío que demanda la colaboración de todos los sectores de la sociedad, desde profesionales de la salud y legisladores, hasta comunidades locales y personas directamente afectadas por esta dolorosa realidad.

Por otro lado, en un ámbito diferente pero estrechamente relacionado, es esencial destacar cómo Python, el versátil lenguaje de programación, ha emergido como una herramienta absolutamente indispensable en el análisis de datos en el contexto de esta crisis, así como en una multitud de otros campos dentro de la Tecnología de la Información (TI). Su capacidad para procesar, visualizar y extraer conocimientos a partir de grandes conjuntos de datos lo ha convertido en una herramienta insustituible para comprender la problemática de la sobredosis de drogas.

En particular, Python permite implementar gráficos interactivos y funciones especializadas para visualizar de manera más efectiva las muertes por sobredosis de droga en América del Norte. Estos gráficos dinámicos ofrecen una visión más clara y comprensible de la extensión de la crisis, permitiendo así identificar patrones geográficos, temporales y demográficos clave. Además, se han desarrollado funciones personalizadas que permiten realizar un análisis más profundo y detallado de los datos, desglosando información relevante y proporcionando información crítica para la toma de decisiones.

A través de ejemplos concretos y casos de estudio, se podrá ilustrar de manera vívida cómo Python se convierte, una y otra vez, en la opción preferida para enfrentar una amplia gama de desafíos tecnológicos en nuestra sociedad cada vez más interconectada. Ya sea en la mejora de la atención médica, la gestión de recursos en la nube, la seguridad cibernética o la automatización de tareas complejas, Python es el motor que impulsa la innovación en la TI y permite a las organizaciones abordar problemas complejos de manera eficiente y efectiva. Su comunidad activa y el continuo desarrollo de bibliotecas y herramientas especializadas lo convierten en un aliado confiable para resolver desafíos tecnológicos en constante evolución. A medida que se profundiza en este tema, se hará evidente que Python no solo es una herramienta esencial, sino también un habilitador de soluciones innovadoras que tienen un impacto significativo en nuestra sociedad en constante transformación.

# II. aplicaciones del lenguaje Python

A continuación, se presentan investigaciones que se han realizado en el mundo haciendo uso del lenguaje de programación Python.

1. *Cómo Python se lanza en el mundo-TI.*

Python se destaca en diversos campos de la Tecnología de la Información, como el desarrollo web, la ciencia de datos y la automatización. A pesar de los desafíos para el año 2019, su amplia gama de bibliotecas y comunidad activa impulsan su crecimiento en este dinámico ámbito del estudio. [1]

1. *Computación paralela de Python en la identificación en línea de Thevenin*

El artículo se enfoca en emplear la computación paralela de Python para identificar en tiempo real los parámetros equivalentes de Thevenin en circuitos eléctricos. Python se eligió por su habilidad en computación paralela, su amplia aplicación en ciencia de datos e inteligencia artificial, y la conveniencia de aprovechar bibliotecas como Lcapy para facilitar la implementación. [2]

1. *Python basada en Big Data: ¿la RSE trae beneficios o penalizaciones económicas?*

El texto aborda el empleo de Python, particularmente el módulo Selenium, para evaluar la responsabilidad social corporativa; posteriormente, describe la utilización de la librería Python pdfplumber para transformar la información en archivos PDF a formato de texto, facilitando así su análisis. [3]

1. *Rendimiento computacional de MATLAB y Python para aplicaciones electromagnéticas.*

El contenido se centra en llevar a cabo pruebas comparativas entre las herramientas de MATLAB y Python al abordar la creación de prototipos para problemas en el ámbito electromagnético. Tanto MATLAB como Python demuestran ser eficientes al permitir que los usuarios resuelvan problemas matemáticamente intensivos con una cantidad mínima de código. Estas evaluaciones experimentales se realizarán en configuraciones de CPU y GPU para medir y exponer los tiempos de ejecución de manera precisa. [4]

1. *Detección de intrusiones de datos con codificación básica de Python y prevención mediante el uso de aplicaciones de intrusión*

El material habla sobre un modelo en desarrollo construido en Python el cual es capaz de detectar entradas (break in/ una forma de entrada a la fuerza) en el sistema a través de verificar el historial(log) del framework en búsqueda de patrones inusuales en la utilización de este a fin de poder detectar estas entradas no autorizadas. [5]

1. *Diseño de algoritmo de mejora de imagen médica basado en Python*

El artículo seleccionado utiliza diversas técnicas de mejora de imágenes médicas en el dominio espacial y aplica el software Python para mejorar los bordes de las imágenes que contienen mucha información. Los resultados experimentales muestran que el algoritmo no solo retiene efectivamente la textura y los detalles en la imagen médica, sino que también mejora el contorno y los bordes de la imagen médica, lo que permite observar algunos detalles que no se pueden observar a simple vista. El uso de Python hace que todo el algoritmo sea más rápido y propicio para que el personal médico observe el enfoque de la imagen médica. [6]

1. *Diseño del sistema automático de operación y mantenimiento VMware vSphere en Python*

El articulo destaca el cambio hacia una era inteligente impulsada por la nube, macrodatos e inteligencia artificial. Las principales empresas chinas de Internet están integrando recursos de computación en la nube, big data e IA para impulsar la actualización industrial. La base es la tecnología de virtualización, con VMware y Microsoft Hyper-V siendo líderes en el mercado chino. El texto también menciona la importancia de la automatización en la gestión de recursos de computación en la nube, destacando el uso de Python y módulos como Pysphere y pyVmomi para adquirir datos y realizar operaciones en hosts ESXi y máquinas virtuales en VMware vSphere, mejorando la eficiencia del desarrollo. La extensión MySQLdb se utiliza para operaciones Python-MySQL. [7]

1. *Efectividad del diagrama de flujo como herramienta de andamiaje para aprender Python*

Este artículo explora la efectividad de los diagramas de flujo como una herramienta de apoyo para aprender a programar, especialmente en un contexto educativo que atiende a estudiantes de minorías subrepresentadas. Compara la eficacia de los diagramas de flujo con Python para desarrollar habilidades de resolución de problemas informáticos y explora la relación entre la capacidad para leer y escribir código en Python. Aunque algunos estudios respaldan los beneficios de los diagramas de flujo en el proceso de programación, otros encuentran resultados mixtos, lo que sugiere que la efectividad de las herramientas visuales puede depender del contexto y las características individuales de los estudiantes. [8]

1. *Predicción salarial con Python: Modelos de aprendizaje automático para el conjunto de datos.*

El artículo explora el uso de modelos de aprendizaje automático para predecir salarios, resaltando su importancia en la motivación laboral. Tres modelos supervisados son presentados: regresión lineal, bosques aleatorios y redes neuronales, aplicados a un conjunto de más de 20,000 salarios en EE. UU. La red neuronal destaca por su precisión, aunque es más lenta en entrenamiento, mientras que la regresión lineal es rápida pero menos precisa. La evaluación se enfoca en precisión y tiempo. El artículo sugiere la utilidad de estos modelos, ofrece un resumen de su implementación en Python, menciona otros modelos y enfatiza la necesidad de datos representativos para precisiones altas. [9]

1. *Redes neuronales en nuevas plataformas habilitadas para Python.*

El artículo se centra en el uso innovador de redes neuronales en tecnologías reconfigurables. Explora diferentes implementaciones, como flujos de datos directos y procesadores "suaves" usando bloques DSP. Se discuten los compromisos entre el costo de implementación y el consumo de energía en varias implementaciones con diferentes niveles de precisión de bits. Además, se menciona el uso de abstracciones basadas en Python, con soporte para una interfaz de Jupyter Notebook, para facilitar estas implementaciones novedosas. Se presentan ejemplos en plataformas integradas y en la nube, como Pynq y AWS F1. También se mencionan las oportunidades en nuevas plataformas Xilinx de 7nm con hardware especializado para redes neuronales. [10]

III. DESCRIPCIÓN DE PÁGINA

La información de los datos utilizados en este informe se obtuvo de Data.gov, el sitio web de datos abiertos del gobierno de los Estados Unidos. Data.gov proporciona acceso a conjuntos de datos publicados por diversas agencias gubernamentales federales. El propósito de Data.gov es brindar acceso público a datos gubernamentales abiertos, cumplir con las misiones de las agencias, fomentar la innovación, impulsar la actividad económica y promover los ideales de un gobierno abierto y transparente.

El conjunto de datos específico utilizado en este informe se titula "Drug overdose deaths - Accidental drug-related deaths during 2012-2018". Este conjunto de datos proporciona información valiosa sobre muertes por sobredosis de drogas en los Estados Unidos de Norteamérica durante el período de 2012 a 2018.

Siendo un recurso de datos gubernamentales, Data.gov se ha convertido en un elemento clave para investigaciones y análisis relacionados con cuestiones de salud pública y seguridad. Los datos aquí recopilados y publicados son esenciales para comprender las tendencias, patrones y evolución de la epidemia de sobredosis de drogas en el país, lo que contribuye a la toma de decisiones informadas y a la formulación de políticas más efectivas para abordar esta problemática de salud.

IV. DESCRIPCIÓN DE PROCESAMIENTO DE LA INFORMACIÓN

Para la extracción de los datos del CSV se usó la librería Pandas para la lectura de los datos y generar el dataframe de los datos encontrados en drug\_deaths.csv. También se hizo uso de matplotlib.pyplot y seaborn para la generación de los gráficos.

1. *rename\_columns.*

Con el fin de un mayor entendimiento de los resultados que se den , se cambian los nombres de las columnas.

def rename\_columns(filename):

df = pd.read\_csv(filename)

new\_column\_names = {

'ID': 'id',

'Date': 'fecha',

'DateType': 'tipo\_fecha',

'Age': 'edad',

'Sex': 'genero',

'Race': 'raza',

'ResidenceCity': 'ciudad\_residencia',

'ResidenceCounty': 'condado\_residencia',

'ResidenceState': 'estado\_residencia',

'DeathCity': 'ciudad\_muerte',

'DeathCounty': 'condado\_muerte',

'Location': 'ubicacion',

'LocationifOther': 'ubicacion\_otra',

'DescriptionofInjury': 'descripcion\_herida',

'InjuryPlace': 'lugar\_herida',

'InjuryCity': 'ciudad\_herida',

'InjuryCounty': 'condado\_herida',

'InjuryState': 'estado\_herida',

'COD': 'causa\_muerte',

'OtherSignifican': 'otras\_causas',

'Heroin': 'heroina',

'Cocaine': 'cocaina',

'Fentanyl': 'fentanilo',

'Fentanyl\_Analogue': 'analogos\_fentanilo',

'Oxycodone': 'oxycodona',

'Oxymorphone': 'oximorfona',

'Ethanol': 'etanol',

'Hydrocodone': 'hidrocodona',

'Benzodiazepine': 'benzodiacepinas',

'Methadone': 'metadona',

'Amphet': 'anfetaminas',

'Tramad': 'tramadol',

'Morphine\_NotHeroin': 'morfina\_no\_heroina',

'Hydromorphone': 'hidromorfona',

'Other': 'otras\_drogas',

'OpiateNOS': 'oxido\_nitroso',

'AnyOpioid': 'cualquier\_opioide',

'MannerofDeath': 'manera\_muerte',

'DeathCityGeo': 'geolocalizacion\_ciudad\_muerte',

'ResidenceCityGeo': 'geolocalizacion\_ciudad\_residencia',

'InjuryCityGeo': 'geolocalizacion\_ciudad\_herida'

}

df = df.rename(columns=new\_column\_names)

return df

df = rename\_columns('drug\_deaths.csv')

print(df.head())

Resultado:

A computer screen shot of a computer

Description automatically generated

Fig.1 Salida del método rename\_columns

1. *Cantidad de Personas en el Estudio por Genero:*

A la hora de iniciar un estudio sobre las muertes por sobredosis se planteo el contabilizar las muertes en grupos(en es el género) para poder hallar patrones en estas muertes.

def Cantidad\_de\_Personas\_en\_el\_Estudio\_por\_Genero(df):

resultMale=df['genero'].value\_counts()['Male']

resultFemale=df['genero'].value\_counts()['Female']

resultUnknown=df['genero'].value\_counts()['Unknown']

total=str(resultMale)+"+"+str(resultFemale)+"+"+str(resultUnknown)

resultTotal=eval(total)

print("Hay un total de muertes de",resultTotal,", de los cuales",resultMale,"son hombres ,", resultFemale,"son mujeres y",resultUnknown,"es desconocido.")

Cantidad\_de\_Personas\_en\_el\_Estudio\_por\_Genero(df)

Resultado:

Hay un total de muertes de 5099, de los cuales 3773 son hombres, 1325 son mujeres y 1 es desconocido.

1. *Cantidad de personas muertas por sobredosis a causa de cualquier Opioide:*

Al iniciar con el análisis de datos se encontró que el grupo de opioides tenían una gran parte de las muertes por lo que se usa el código en Python para dar con la cifra exacta.

def Muertes\_por\_Cualquier\_Opioide\_totales(df):

resultAnyOpioid=df['cualquier\_opioide'].value\_counts()[1]

print("La cantidad de personas muertas a causa de cualquier opiode es de",str(resultAnyOpioid))

Muertes\_por\_Cualquier\_Opioide\_totales(df)

Resultado:

La cantidad de personas muertas a causa de cualquier opiode es de 2460.

1. *Muertes por Cualquier Opioide por género:*

De la mano con el resultado anterior ya teniendo la cifra total se opto por estratificar por genero estas muertes a fin de hallar algún patrón o afinidad en los grupos.

def Muertes\_por\_Cualquier\_Opioide\_por\_genero(df):

AnyOpioidbySex=df[['cualquier\_opioide','genero']]

AnyOpioidbySex=AnyOpioidbySex.loc[(AnyOpioidbySex['cualquier\_opioide']=="1")]

resultFemaleOpioid=AnyOpioidbySex['genero'].value\_counts()['Female']

resultMaleOpioid=AnyOpioidbySex['genero'].value\_counts()['Male']

resultUnknownOpioid=AnyOpioidbySex['genero'].value\_counts()['Unknown']

print("La cantidad de hombres que fallecieron por sobredosis de cualquier opioide es de",abs(resultMaleOpioid),"y la cantidad de mujeres fue de",abs(resultFemaleOpioid),

"y la cantidad de desconocidos fue de",abs(resultUnknownOpioid))

Muertes\_por\_Cualquier\_Opioide\_por\_genero(df)

Resultado:

La cantidad de hombres que fallecieron por sobredosis de cualquier opioide es de 1862 y la cantidad de mujeres fue de 594 y la cantidad de desconocidos fue de 1.

1. *Muertes\_por\_Cualquier\_Opioide\_por\_genero\_en\_rangos\_de\_edad:*

Una vez encontradas las cantidades de muertes en cada genero se opto por analizarlas en base a la edad a fin de poder hallar una correlación entre la edad y el genero en las muertes por sobredosis.

def Muertes\_por\_Cualquier\_Opioide\_por\_genero\_en\_rangos\_de\_edad(df):

AnyOpioidbySex=df[['cualquier\_opioide','genero','edad']]

AnyOpioidbySex=AnyOpioidbySex.loc[(AnyOpioidbySex['cualquier\_opioide']=="1")]

resultFemaleOpioid=AnyOpioidbySex[(df['genero']=='Female')]

resultFemaleOpioid=resultFemaleOpioid['edad'].value\_counts()

resultMaleOpioid=AnyOpioidbySex[(df['genero']=='Male')]

resultMaleOpioid=resultMaleOpioid['edad'].value\_counts()

resultUnknownOpioid=AnyOpioidbySex[(df['genero']=='Unknown')]

resultUnknownOpioid=resultUnknownOpioid['edad'].value\_counts()

new\_column\_names = {

'edad': 'cantidad'

}

print("La menor edad en las muertes por opioides fue de",int(AnyOpioidbySex.edad.min()),"mientras que la mayor edad registrada fue de",int(AnyOpioidbySex.edad.max()),

"finalmente la edad promedio en las muertes fue de",int(AnyOpioidbySex.edad.mean()),"\n","con una desviacion estandar",

int(AnyOpioidbySex.edad.std()),"por lo que podemos concluir que hay una alta varianza en las edades")

resultFemaleOpioid=pd.DataFrame(resultFemaleOpioid)

resultFemaleOpioid = resultFemaleOpioid.rename(columns=new\_column\_names)

print("\nTabla de Muertes femeninas por cualquier opioide en rangos de edad\n",resultFemaleOpioid.sort\_index(axis = 0))

resultMaleOpioid=pd.DataFrame(resultMaleOpioid)

resultMaleOpioid = resultMaleOpioid.rename(columns=new\_column\_names)

print("\nTabla de Muertes masculinas por cualquier opioide en rangos de edad\n",resultMaleOpioid.sort\_index(axis = 0))

resultUnknownOpioid=pd.DataFrame(resultUnknownOpioid)

resultUnknownOpioid = resultUnknownOpioid.rename(columns=new\_column\_names)

print("\nTabla de Muertes de genero desconocido por cualquier opioide en rangos de edad\n",resultUnknownOpioid.sort\_index(axis = 0))

Muertes\_por\_Cualquier\_Opioide\_por\_genero\_en\_rangos\_de\_edad(df)

Resultados:

Nota: El resultado puede quedar un poco extenso, pero esto se debe a la cantidad de datos del estudio. La menor edad en las muertes por opioides fue de 15 mientras que la mayor edad registrada fue de 84 finalmente la edad promedio en las muertes fue de 41 con una desviacion estandar 12 por lo que podemos concluir que hay una alta varianza en las edades.

Tabla de Muertes femeninas por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | cantidad |
| 17.0 | 5 |
| 18.0 | 1 |
| 19.0 | 6 |
| 20.0 | 2 |
| 21.0 | 10 |
| 22.0 | 9 |
| 23.0 | 7 |
| 24.0 | 10 |
| 25.0 | 16 |
| 26.0 | 12 |
| 27.0 | 15 |
| 28.0 | 18 |
| 29.0 | 14 |
| 30.0 | 16 |
| 31.0 | 10 |
| 32.0 | 14 |
| 33.0 | 18 |
| 34.0 | 16 |
| 35.0 | 9 |
| 36.0 | 11 |
| 37.0 | 15 |
| 38.0 | 13 |
| 39.0 | 13 |
| 40.0 | 15 |
| 41.0 | 13 |
| 42.0 | 9 |
| 43.0 | 9 |
| 44.0 | 13 |
| 45.0 | 14 |
| 46.0 | 8 |
| 47.0 | 15 |
| 48.0 | 22 |
| 49.0 | 15 |
| 50.0 | 19 |
| 51.0 | 18 |
| 52.0 | 12 |
| 53.0 | 12 |
| 54.0 | 23 |
| 55.0 | 17 |
| 56.0 | 13 |
| 57.0 | 23 |
| 58.0 | 11 |
| 59.0 | 11 |
| 60.0 | 10 |
| 61.0 | 5 |
| 62.0 | 7 |
| 63.0 | 4 |
| 64.0 | 2 |
| 65.0 | 3 |
| 66.0 | 5 |
| 71.0 | 1 |
| 72.0 | 5 |

Tabla de Muertes masculinas por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | Cantidad |
| 15.0 | 1 |
| 17.0 | 1 |
| 18.0 | 5 |
| 19.0 | 8 |
| 20.0 | 13 |
| ... | ... |
| 72.0 | 2 |
| 73.0 | 2 |
| 74.0 | 1 |
| 75.0 | 1 |
| 84.0 | 1 |

[61 rows x 1 columns]

Tabla de Muertes de genero desconocido por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | Cantidad |
| 66.0 | 1 |

<ipython-input-11-9716ad0381fe>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultFemaleOpioid=AnyOpioidbySex[(df['genero']=='Female')]

<ipython-input-11-9716ad0381fe>:6: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultMaleOpioid=AnyOpioidbySex[(df['genero']=='Male')]

<ipython-input-11-9716ad0381fe>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultUnknownOpioid=AnyOpioidbySex[(df['genero']=='Unknown')]

1. *Muertes\_por\_Hidromorfona\_totales:*

Al analizar las cifras de los opioides se llega a ver una clara disparidad entre las muertes en hombres y mujeres, por lo que se optó por realizar el mismo análisis en una droga que posee un grupo mucho menor de muertes atribuidas.

def Muertes\_por\_Hidromorfona\_totales(df):

resultHydromorphone=df['hidromorfona'].value\_counts()[1]

print("La cantidad de personas muertas a causa de hidromorfona es de",str(resultHydromorphone))

Muertes\_por\_Hidromorfona\_totales(df)

Resultado:

La cantidad de personas muertas a causa de hidromorfona es de 25.

1. *Muertes\_por\_Hidromorfona\_por\_genero:*

Siguiendo con el método pasado ahora se usa el código para sacar de las muertes por hidromorfona las muertes de cada género.

def Muertes\_por\_Hidromorfona\_por\_genero(df):

HydromorphonebySex=df[['hidromorfona','genero']]

HydromorphonebySex=HydromorphonebySex.loc[(HydromorphonebySex['hidromorfona']==1)]

resultFemaleHydromorphone=HydromorphonebySex['genero'].value\_counts()['Female']

resultMaleHydromorphone=HydromorphonebySex['genero'].value\_counts()['Male']

resultUnknownHydromorphone=HydromorphonebySex['genero'].value\_counts()['Unknown']

print("La cantidad de hombres que fallecieron por sobredosis de hidromorfona es de",abs(resultMaleHydromorphone),"y la cantidad de mujeres fue de",abs(resultFemaleHydromorphone),

"y la cantidad de desconocido fue de",abs(resultUnknownHydromorphone))

Muertes\_por\_Hidromorfona\_por\_genero(df)

Resultado:

La cantidad de hombres que fallecieron por sobredosis de hidromorfona es de 15 y la cantidad de mujeres fue de 9 y la cantidad de desconocido fue de 1.

1. *Muertes\_por\_Hidromorfona\_por\_genero\_Muertes\_por\_Hidromorfona\_por\_genero\_en\_rangos\_de\_edad:*

Finalmente, para que los resultados entre ambas drogas sean comparables se agrupan en rangos de edad las muertes por hidromorfona.

def Muertes\_por\_Hidromorfona\_por\_genero\_en\_rangos\_de\_edad(df):

HydromorphonebySex=df[['hidromorfona','genero','edad']]

HydromorphonebySex=HydromorphonebySex.loc[(HydromorphonebySex['hidromorfona']==1)]

resultFemaleHydromorphone=HydromorphonebySex[(df['genero']=='Female')]

resultFemaleHydromorphone=resultFemaleHydromorphone['edad'].value\_counts()

resultMaleHydromorphone=HydromorphonebySex[(df['genero']=='Male')]

resultMaleHydromorphone=resultMaleHydromorphone['edad'].value\_counts()

resultUnknownHydromorphone=HydromorphonebySex[(df['genero']=='Unknown')]

resultUnknownHydromorphone=resultUnknownHydromorphone['edad'].value\_counts()

new\_column\_names = {

'edad': 'cantidad'

}

print("La menor edad en las muertes por opioides fue de",int(HydromorphonebySex.edad.min()),"mientras que la mayor edad registrada fue de",int(HydromorphonebySex.edad.max()),

"finalmente la edad promedio en las muertes fue de",int(HydromorphonebySex.edad.mean()),"\n","con una desviacion estandar",

int(HydromorphonebySex.edad.std()),"por lo que podemos concluir que hay una alta varianza en las edades")

resultFemaleHydromorphone=pd.DataFrame(resultFemaleHydromorphone)

resultFemaleHydromorphone = resultFemaleHydromorphone.rename(columns=new\_column\_names)

print("\nTabla de Muertes femeninas por cualquier opioide en rangos de edad\n",resultFemaleHydromorphone.sort\_index(axis = 0))

resultMaleHydromorphone=pd.DataFrame(resultMaleHydromorphone)

resultMaleHydromorphone = resultMaleHydromorphone.rename(columns=new\_column\_names)

print("\nTabla de Muertes masculinas por cualquier opioide en rangos de edad\n",resultMaleHydromorphone.sort\_index(axis = 0))

resultUnknownHydromorphone=pd.DataFrame(resultUnknownHydromorphone)

resultUnknownHydromorphone = resultUnknownHydromorphone.rename(columns=new\_column\_names)

print("\nTabla de Muertes de genero desconocido por cualquier opioide en rangos de edad\n",resultUnknownHydromorphone.sort\_index(axis = 0))

Muertes\_por\_Hidromorfona\_por\_genero\_en\_rangos\_de\_edad(df)

Resultado:

La menor edad en las muertes por opioides fue de 23 mientras que la mayor edad registrada fue de 66 finalmente la edad promedio en las muertes fue de 49 con una desviación estándar 12 por lo que podemos concluir que hay una alta varianza en las edades.

Tabla de Muertes femeninas por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | Cantidad |
| 36.0 | 1 |
| 41.0 | 1 |
| 48.0 | 1 |
| 50.0 | 1 |
| 55.0 | 1 |
| 56.0 | 1 |
| 61.0 | 1 |
| 62.0 | 1 |
| 66.0 | 1 |

Tabla de Muertes masculinas por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | Cantidad |
| 23.0 | 1 |
| 32.0 | 2 |
| 34.0 | 1 |
| 37.0 | 2 |
| 47.0 | 1 |
| 52.0 | 2 |
| 53.0 | 1 |
| 57.0 | 1 |
| 59.0 | 1 |
| 60.0 | 2 |
| 63.0 | 1 |

Tabla de Muertes de genero desconocido por cualquier opioide en rangos de edad

|  |  |
| --- | --- |
| Edad | Cantidad |
| 66.0 | 1 |

<ipython-input-14-d733fd344761>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultFemaleHydromorphone=HydromorphonebySex[(df['genero']=='Female')]

<ipython-input-14-d733fd344761>:6: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultMaleHydromorphone=HydromorphonebySex[(df['genero']=='Male')]

<ipython-input-14-d733fd344761>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.

resultUnknownHydromorphone=HydromorphonebySex[(df['genero']=='Unknown')]

1. *Causas\_de\_muerte:*

Al realizar el análisis de los datos se encontró con que las muertes pueden tener 3 posibles causas de muerte Accidental, Natural o pendiente por lo que se optó por cuantificar estas cifras.

def Causas\_de\_muerte(df):

resultAccident=df['manera\_muerte'].value\_counts()['Accident']

resultAccident=resultAccident+df['manera\_muerte'].value\_counts()['ACCIDENT']

resultAccident=resultAccident+df['manera\_muerte'].value\_counts()['accident']

resultNatural=df['manera\_muerte'].value\_counts()['Natural']

resultPending=df['manera\_muerte'].value\_counts()['Pending']

MannerofDeathArray=[resultAccident,resultNatural,resultPending]

MannerofDeath=sum(MannerofDeathArray)

print("El número total de muertes fue de",MannerofDeath,", de las cuales",resultAccident,"fueron accidentales,",

resultNatural,"fueron naturales y",resultPending,"siguen pendientes a determinar al momento del estudio.")

Causas\_de\_muerte(df)

Resultado:

El número total de muertes fue de 5095, de las cuales 5080 fueron accidentales, 1 fueron naturales y 14 siguen pendientes a determinar al momento del estudio.

1. *Muertes segun la edad de los ciudadanos en el estado de Connecticut y New York en el año 2018:*

En la misma línea de hallar correlaciones se platea si la edad demuestra tener una alta correlación con la ciudad de residencia, por lo que se optó por realizar una lista de edades y su cantidad de muertes en las ciudades de Connecticut y Nueva York en el año 2018.

def muertes\_por\_edad\_2018\_NY(df):

df\_2018\_NY = df[(df['fecha'].str.contains('2018')) & (df['estado\_residencia'] == 'NY')]

muertes\_por\_edad = df\_2018\_NY['edad'].value\_counts()

print( f"El número de muertes por drogas en el estado de NY en el año 2018, por edad, es:\n{muertes\_por\_edad}" )

muertes\_por\_edad\_2018\_NY(df)

def muertes\_por\_edad\_2018\_CT(df):

df\_2018\_CT = df[(df['fecha'].str.contains('2018')) & (df['estado\_residencia'] == 'CT')]

muertes\_por\_edad = df\_2018\_CT['edad'].value\_counts()

print( f"El número de muertes por drogas en el estado de Connecticut en el año 2018, por edad, es:\n{muertes\_por\_edad}" )

muertes\_por\_edad\_2018\_CT(df)

Resultado:

El número de muertes por drogas en el estado de NY en el año 2018, por edad, es:

|  |  |
| --- | --- |
| Edad | Cantidad |
| 29.0 | 3 |
| 27.0 | 1 |
| 58.0 | 1 |
| 41.0 | 1 |
| 22.0 | 1 |

Name: edad, dtype: int64

El número de muertes por drogas en el estado de Connecticut en el año 2018, por edad, es:

|  |  |
| --- | --- |
| Edad | Cantidad |
| 36.0 | 34 |
| 34.0 | 30 |
| 54.0 | 28 |
| 53.0 | 27 |
| 35.0 | 26 |
| 44.0 | 26 |
| 41.0 | 26 |
| 29.0 | 26 |
| 48.0 | 26 |
| 30.0 | 26 |
| 37.0 | 25 |
| 52.0 | 25 |
| 51.0 | 25 |
| 57.0 | 24 |
| 43.0 | 24 |
| 50.0 | 24 |
| 33.0 | 23 |
| 46.0 | 23 |
| 47.0 | 22 |
| 42.0 | 22 |
| 32.0 | 21 |
| 39.0 | 21 |
| 28.0 | 21 |
| 40.0 | 21 |
| 56.0 | 20 |
| 49.0 | 20 |
| 38.0 | 20 |
| 27.0 | 19 |
| 59.0 | 19 |
| 24.0 | 19 |
| 45.0 | 17 |
| 23.0 | 17 |
| 55.0 | 17 |
| 60.0 | 16 |
| 64.0 | 14 |
| 31.0 | 14 |
| 61.0 | 14 |
| 25.0 | 14 |
| 58.0 | 12 |
| 63.0 | 12 |
| 22.0 | 11 |
| 65.0 | 11 |
| 26.0 | 11 |
| 66.0 | 9 |
| 62.0 | 9 |
| 21.0 | 7 |
| 20.0 | 5 |
| 67.0 | 4 |
| 69.0 | 4 |
| 19.0 | 3 |
| 17.0 | 3 |
| 68.0 | 2 |
| 71.0 | 1 |
| 15.0 | 1 |
| 74.0 | 1 |
| 84.0 | 1 |
| 72.0 | 1 |
| 75.0 | 1 |

Name: edad, dtype: int64

1. *Tasa de Mortalidad:*

De muertes según la edad de los ciudadanos en el estado de CT y NY en el año 2018. De la mano con el método anterior se busca hallar la tasa de mortalidad según la edad en las ciudades de Connecticut y Nueva York en el año 2018.

# Población de Nueva York y Connecticut en 2018

poblacion\_NY = 19530000

poblacion\_CT = 3573000

# Número de muertes por drogas en Nueva York y Connecticut en 2018, por edad

muertes\_por\_edad\_NY = df[(df['fecha'].str.contains('2018')) & (df['estado\_residencia'] == 'NY')]['edad'].value\_counts()

muertes\_por\_edad\_CT = df[(df['fecha'].str.contains('2018')) & (df['estado\_residencia'] == 'CT')]['edad'].value\_counts()

# Tasa de mortalidad por drogas en Nueva York y Connecticut en 2018, por edad

tasa\_mortalidad\_NY = (muertes\_por\_edad\_NY / poblacion\_NY) \* 100000

tasa\_mortalidad\_CT = (muertes\_por\_edad\_CT / poblacion\_CT) \* 100000

print("Tasa de mortalidad por drogas en Nueva York en 2018, por edad:")

print(tasa\_mortalidad\_NY)

print("\nTasa de mortalidad por drogas en Connecticut en 2018, por edad:")

print(tasa\_mortalidad\_CT)

max\_tasa\_mortalidad\_NY = tasa\_mortalidad\_NY.max()

max\_tasa\_mortalidad\_CT = tasa\_mortalidad\_CT.max()

if max\_tasa\_mortalidad\_NY > max\_tasa\_mortalidad\_CT:

print("La tasa de mortalidad más alta por drogas en 2018, por edad, fue en Nueva York.")

elif max\_tasa\_mortalidad\_NY < max\_tasa\_mortalidad\_CT:

print("La tasa de mortalidad más alta por drogas en 2018, por edad, fue en Connecticut.")

else:

print("La tasa de mortalidad por drogas en 2018, por edad, fue igual en ambos estados.")

Resultado:

Tasa de mortalidad por drogas en Nueva York en 2018, por edad:

|  |  |
| --- | --- |
| Edad | Cantidad |
| 29.0 | 0.015361 |
| 27.0 | 0.005120 |
| 58.0 | 0.005120 |
| 41.0 | 0.005120 |
| 22.0 | 0.005120 |

Name: edad, dtype: float64

Tasa de mortalidad por drogas en Connecticut en 2018, por edad:

|  |  |
| --- | --- |
| Edad | Cantidad |
| 36.0 | 0.951581 |
| 34.0 | 0.839631 |
| 54.0 | 0.783655 |
| 53.0 | 0.755668 |
| 35.0 | 0.727680 |
| 44.0 | 0.727680 |
| 41.0 | 0.727680 |
| 29.0 | 0.727680 |
| 48.0 | 0.727680 |
| 30.0 | 0.727680 |
| 37.0 | 0.699692 |
| 52.0 | 0.699692 |
| 51.0 | 0.699692 |
| 57.0 | 0.671704 |
| 43.0 | 0.671704 |
| 50.0 | 0.671704 |
| 33.0 | 0.643717 |
| 46.0 | 0.643717 |
| 47.0 | 0.615729 |
| 42.0 | 0.615729 |
| 32.0 | 0.587741 |
| 39.0 | 0.587741 |
| 28.0 | 0.587741 |
| 40.0 | 0.587741 |
| 56.0 | 0.559754 |
| 49.0 | 0.559754 |
| 38.0 | 0.559754 |
| 27.0 | 0.531766 |
| 59.0 | 0.531766 |
| 24.0 | 0.531766 |
| 45.0 | 0.475791 |
| 23.0 | 0.475791 |
| 55.0 | 0.475791 |
| 60.0 | 0.447803 |
| 64.0 | 0.391828 |
| 31.0 | 0.391828 |
| 61.0 | 0.391828 |
| 25.0 | 0.391828 |
| 58.0 | 0.335852 |
| 63.0 | 0.335852 |
| 22.0 | 0.307865 |
| 65.0 | 0.307865 |
| 26.0 | 0.307865 |
| 66.0 | 0.251889 |
| 62.0 | 0.251889 |
| 21.0 | 0.195914 |
| 20.0 | 0.139938 |
| 67.0 | 0.111951 |
| 69.0 | 0.111951 |
| 19.0 | 0.083963 |
| 17.0 | 0.083963 |
| 68.0 | 0.055975 |
| 71.0 | 0.027988 |
| 15.0 | 0.027988 |
| 74.0 | 0.027988 |
| 84.0 | 0.027988 |
| 72.0 | 0.027988 |
| 75.0 | 0.027988 |

Name: edad, dtype: float64

La tasa de mortalidad más alta por drogas en 2018, por edad, fue en Connecticut.

1. *Cantidades de muertes por sobredosis en la región madison:*

Al momento de realizar el estudio y analizar los datos no se hallan datos de la ciudad de Madison por lo que a fin de corroborar esto ultimo se desarrolla un método que verifique si la cifra de muertes en Madison fue de 0.

def muerteSobredosisMadison(df):

dfMadison = df[df['estado\_residencia'] == 'MADISON']

numero\_muertes = dfMadison.shape[0] # Calcula el número de filas en dfMadison

print(f"El número de muertes por sobredosis de droga en la región de Madison es: {numero\_muertes}")

# Llama a la función pasando tu DataFrame como argumento

muerteSobredosisMadison(df)

Resultado:

El número de muertes por sobredosis de droga en la región de Madison es: 0

1. *Números de muertes en personas mayores a 25 años con sobredosis de morfina:*

Como se puede apreciar en las muertes de Connecticut en las distintas edades la mayoría de las edades se ubican arriba de los 25 años por lo que se realiza un análisis sobre las muertes por sobredosis de morfina, pero solo en personas mayores a 25 años

def contar\_personas\_sobredosis\_morfina(df):

# Filtrar las filas donde la edad es mayor de 25 y hay sobredosis de morfina

personas\_sobredosis\_morfina = df[(df['edad'] > 25) & (df['morfina\_no\_heroina'] == 1)]

# Contar la cantidad de personas en el DataFrame resultante

cantidad\_personas = len(personas\_sobredosis\_morfina)

# Imprimir la cantidad de personas

print(f"Cantidad de personas mayores de 25 años con sobredosis de morfina: {cantidad\_personas}")

# Llamar a la función con tu DataFrame

df = pd.DataFrame({'edad': [30, 40, 20, 28, 35],

'morfina\_no\_heroina': [1, 0, 1, 1, 0]})

contar\_personas\_sobredosis\_morfina(df)

Resultado:

Cantidad de personas mayores de 25 años con sobredosis de morfina: 2

1. *Muertes\_por\_Tipo\_de\_Droga:*

Ya habiendo realizado diversos análisis se opta por encontrar el total de muertes de todas las drogas abarcadas por el estudio.

def Muertes\_por\_Tipo\_de\_Droga(df):

drogas = ['heroina', 'cocaina', 'fentanilo', 'analogos\_fentanilo', 'oxycodona', 'oximorfona', 'etanol',

'hidrocodona', 'benzodiacepinas', 'metadona', 'anfetaminas', 'tramadol', 'morfina\_no\_heroina',

'oxido\_nitroso', 'cualquier\_opioide', 'otras\_drogas']

for droga in drogas:

df[droga] = pd.to\_numeric(df[droga], errors='coerce')

muertes\_por\_droga = {}

for droga in drogas:

muertes\_por\_droga[droga] = df[droga].sum()

nombres\_drogas = list(muertes\_por\_droga.keys())

muertes = list(muertes\_por\_droga.values())

dfpresent = pd.DataFrame({'droga' : list(muertes\_por\_droga.keys()), 'cantidad' : list(muertes\_por\_droga.values())})

print("Lista de drogas con sus muertes totales asociadas")

print(dfpresent)

Muertes\_por\_Tipo\_de\_Droga(df)

Resultado:

Lista de drogas con sus muertes totales asociadas

|  |  |  |
| --- | --- | --- |
|  | droga | cantidad |
| 0 | heroína | 2529.0 |
| 1 | cocaína | 1521.0 |
| 2 | fentanilo | 2228.0 |
| 3 | analogos\_fentanilo | 389.0 |
| 4 | oxycodona | 607.0 |
| 5 | oximorfona | 108.0 |
| 6 | etanol | 1247.0 |
| 7 | hidrocodona | 118.0 |
| 8 | benzodiacepinas | 1343.0 |
| 9 | metadona | 474.0 |
| 10 | anfetaminas | 159.0 |
| 11 | tramadol | 130.0 |
| 12 | morfina\_no\_heroina | 38.0 |
| 13 | oxido\_nitroso | 88.0 |
| 14 | cualquier\_opioide | 2460.0 |
| 15 | otras\_drogas | 0.0 |

1. *Data\_Evolucion\_de\_Muertes\_por\_Sobredosis\_por\_Año:*

Finalmente se encuentra la tendencia en las muertes año por año.

def Data\_Evolucion\_de\_Muertes\_por\_Sobredosis\_por\_Año(df):

df['fecha'] = pd.to\_datetime(df['fecha'])

df['año'] = df['fecha'].dt.year

muertes\_por\_año = df.groupby('año').size() #Agrupa y cuenta las muertes por año

muertes\_por\_año.columns =['Año', 'Cantidad']

print("Lista de años con sus respectivas muertes por sobredosis")

print(muertes\_por\_año)

Data\_Evolucion\_de\_Muertes\_por\_Sobredosis\_por\_Año(df)

Resultado:

Lista de años con sus respectivas muertes por sobredosis

|  |  |
| --- | --- |
| Año |  |
| 2012.0 | 355 |
| 2013.0 | 490 |
| 2014.0 | 558 |
| 2015.0 | 727 |
| 2016.0 | 917 |
| 2017.0 | 1038 |
| 2018.0 | 1018 |

dtype: int64

V. ANÁLISIS DE LOS DATOS

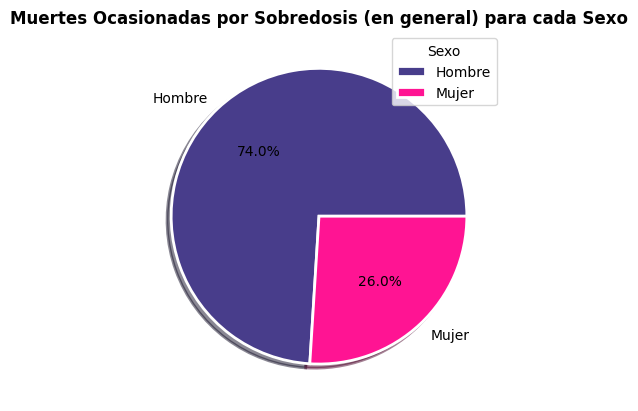
Para llevar a cabo un análisis efectivo de los datos, se emplearon diversas herramientas de programación en Python. Las librerías Matplotlib y Seaborn se utilizaron para crear gráficos visuales y mapas de calor que facilitaron la interpretación de la información y la visualización de la correlación entre variables. Además, se aprovechó la librería Pandas para la manipulación y análisis de datos, incluyendo la carga de conjuntos de datos y la realización de cálculos estadísticos. Cada una de estas librerías desempeñó un papel crucial en la visualización y el análisis de los datos recopilados.

1. *Gráfico 1: Muertes Ocasionadas por Sobredosis (en general) para cada Sexo:*

# Mapear los valores originales a los valores en español

sex\_map = {'Male': 'Hombre', 'Female': 'Mujer'}  
df['genero'] = df['genero'].map(sex\_map)  
# Agrupar los datos por la columna 'Sex' y contar el número de ocurrencias  
sex\_counts = df.groupby('genero').size()  
colors = ['darkslateblue','deeppink', 'darkslategreys']  
# Crear el gráfico de pastel  
plt.pie(sex\_counts, labels=sex\_counts.index, autopct='%1.1f%%', colors=colors, shadow=True,  
wedgeprops={'linewidth': 2, 'edgecolor': 'white'})  
plt.legend(title='Sexo', loc='best', labels=sex\_counts.index)  
plt.title('Muertes Ocasionadas por Sobredosis (en general) para cada Sexo', fontweight='bold')  
plt.figure(figsize=(8, 6))  
plt.show()

Resultado:



*Gráfico 1 Muertes Ocasionadas por Sobredosis (en general) para cada Sexo*

El gráfico 1, representa la distribución de muertes por sobredosis en función del género en América del Norte. Hombres (Hombre) representan el 74% de las muertes por sobredosis en general, mientras que las mujeres (Mujer) representan el 26%. La predominancia de muertes en hombres sugiere una mayor vulnerabilidad o exposición a las causas de las sobredosis de drogas en comparación con las mujeres.

1. *Gráfico 2: Histograma del Número de Muertes por Sobredosis (en general) según el Rango de Edad:*

# Cargar los datos en un DataFrame de Pandas

df = pd.read\_csv('drug\_deaths.csv')

# Crear el histograma con los datos de edad

a = df.Age.plot.hist(color='skyblue', edgecolor='white', bins=20)

a.set\_title(r'Número de Muertes por Sobredosis (en general) según el Rango de Edad', fontweight='bold')

a.set\_xlabel('Rango de Edad')

a.set\_ylabel('Muertes por Sobredosis')

# Agregar una cuadrícula

a.grid(axis='y', alpha=0.75)

plt.show()

Resultado:

A graph of blue bars

Description automatically generated

*Gráfico 2 Histograma del Número de Muertes por Sobredosis (en general) según el Rango de Edad*

El gráfico 2, muestra una representación visual del número de muertes por sobredosis en función del rango de edad en América del Norte. Se observa que la mayoría de las muertes, con más de 500 casos, se concentran en dos grupos de edad: personas cercanas a los 30 años y aquellos que están cerca de los 50 años. Además, se nota una marcada disminución en el número de muertes en los extremos de edad, es decir, entre los menores de 20 años y los mayores de 70 años.

Los grupos de edad cercanos a los 30 años y a los 50 años parecen ser los más vulnerables a las muertes por sobredosis. Esto podría estar relacionado con factores como el acceso a las drogas, la prevalencia de ciertas adicciones y las condiciones de salud en esas etapas de la vida. La disminución en el número de muertes en los extremos de edad (menores de 20 y mayores de 70 años) sugiere que estas poblaciones pueden estar menos expuestas o propensas a las causas de las sobredosis de drogas. Esto puede deberse a diferencias en la disponibilidad y el uso de sustancias.

1. *Gráfico 3: Evolución de las Muertes por Sobredosis (en general) a lo Largo de los Años (2012-2018)*

def Evolucion\_de\_Muertes\_por\_Sobredosis\_por\_Ano(df):

    df['fecha'] = pd.to\_datetime(df['fecha'])

    df['año'] = df['fecha'].dt.year

    muertes\_por\_año = df.groupby('año').size() #Agrupa y cuenta las muertes por año

    plt.figure(figsize=(10, 6))

    plt.plot(muertes\_por\_año.index, muertes\_por\_año.values, marker='o', linestyle='-', color='b')

    plt.xlabel('Año')

    plt.ylabel('Número de Muertes')

    plt.title('Evolución de Muertes por Sobredosis (en general) en los Estados Unidos (2012-2018)')

    plt.grid(True)

    plt.show()

Evolucion\_de\_Muertes\_por\_Sobredosis\_por\_Ano(df)

Resultado:

A graph with a line going up

Description automatically generated

*Gráfico 3 Evolución de las Muertes por Sobredosis (en general) a lo Largo de los Años (2012-2018)*

Este gráfico 3, representa la evolución temporal del número de muertes por sobredosis en los Estados Unidos durante el período comprendido entre 2012 y 2018. Inicialmente, en el año 2012, se registra un número de muertes por debajo de 400. A partir de ese punto, se observa un aumento continuo en el número de muertes cada año, alcanzando su punto máximo en 2017, con más de 1000 muertes. Sin embargo, en 2018 se aprecia una disminución en el número de muertes con respecto al año anterior, aunque sigue siendo superior a 1000 casos. El mayor incremento en el número de muertes parece haber ocurrido entre los años 2015 y 2016. En 2015, se registraron cerca de 700 muertes, y al año siguiente, la cifra superó los 900 casos.

Durante la mayor parte del período, se observa un crecimiento constante en el número de muertes por sobredosis. Esto sugiere la necesidad de medidas y políticas efectivas para abordar esta crisis de salud pública. El año 2017 marca el punto más alto de muertes por sobredosis en este período, superando la cifra de 1000. Este podría ser un punto de referencia crítico para comprender la gravedad de la crisis.

1. *Gráfico 4: Cantidad de Muertes por Sobredosis Conforme a el Tipo de Droga (según el documento):*

def Grafico\_Muertes\_por\_Tipo\_de\_Droga(df):

    drogas = ['heroina', 'cocaina', 'fentanilo', 'analogos\_fentanilo', 'oxycodona', 'oximorfona', 'etanol',

             'hidrocodona', 'benzodiacepinas', 'metadona', 'anfetaminas', 'tramadol', 'morfina\_no\_heroina',

             'oxido\_nitroso', 'cualquier\_opioide', 'otras\_drogas']

    for droga in drogas:

        df[droga] = pd.to\_numeric(df[droga], errors='coerce')

    muertes\_por\_droga = {}

    for droga in drogas:

        muertes\_por\_droga[droga] = df[droga].sum()

    nombres\_drogas = list(muertes\_por\_droga.keys())

    muertes = list(muertes\_por\_droga.values())

    plt.figure(figsize=(12, 6))

    plt.barh(nombres\_drogas, muertes, color='skyblue')

    plt.xlabel('Número de Muertes')

    plt.ylabel('Tipo de Droga')

    plt.title('Número de Muertes por Tipo de Droga (según el Documento)')

    plt.gca().invert\_yaxis()

    plt.show()

Grafico\_Muertes\_por\_Tipo\_de\_Droga(df)

Resultado:

A graph with blue lines

Description automatically generated

*Gráfico 4 Cantidad de Muertes por Sobredosis Conforme a el Tipo de Droga (según el documento)*

Este gráfico 4, ofrece una visión fundamental de las drogas más relacionadas con las muertes por sobredosis en el conjunto de datos. Las sustancias más destacadas son la heroína, el fentanilo, la cocaína, las benzodiacepinas y el etanol (alcohol), que presentan una cantidad significativa de muertes registradas. Se observa que la heroína es la droga con la mayor cantidad de muertes registradas, superando las 2500 muertes. Esto destaca la gravedad de la crisis de la heroína y del fentanilo, así como la importancia de abordar el uso indebido de medicamentos recetados y los riesgos del consumo excesivo de alcohol. Por otro lado, algunas drogas como la oximorfona y el óxido nitroso, aunque en menor cantidad, siguen siendo preocupantes y requieren atención.

1. *Gráfico 5: Mapa de Calor sobre la Correlación entre la Heroína y la Edad (según una muestra aleatoria):*

from random import sample

contingency\_table = pd.crosstab(df['heroina'].head(15),sample(sorted(df['edad']),15))

plt.figure(figsize=(10,7))

sns.heatmap(contingency\_table,annot=True,cmap="coolwarm")

#Mostrar el grafico

plt.xlabel('Edad')

plt.ylabel('Consumo de Heroina(0 no consume,1 si consume)')

plt.figtext(0.1, 0.01, "Grafico basado en una muestra aleatoria", ha="center", fontsize=10, bbox={"facecolor":"lightblue", "alpha":0.5, "pad":5})

plt.title('Mapa de Calor de Heroina vs Edad')

plt.show()

Resultado:

A graph with numbers and a red and blue line

Description automatically generated

*Gráfico 5 Mapa de Calor sobre la Correlación entre la Heroína y la Edad (según una muestra aleatoria)*

El gráfico 5, genera un mapa de calor utilizando la función pd.crosstab() de Pandas y la función sns.heatmap() de Seaborn. Este mapa representa la frecuencia de combinaciones de edad y consumo de heroína en un conjunto de datos. Se utiliza un DataFrame llamado 'df', tomando las primeras 15 filas de la columna 'heroina' y una muestra aleatoria de 15 valores de la columna 'edad'. El eje x muestra la edad, el eje y muestra el consumo de heroína (0 para no consumo, 1 para consumo), y los colores reflejan la frecuencia de cada combinación.

La interpretación del mapa de calor varía según el objetivo del análisis. Puede revelar patrones de consumo en diferentes grupos de edad o identificar áreas con una alta incidencia de consumo de heroína. En general, es una herramienta valiosa para visualizar la distribución de datos y descubrir patrones y tendencias.

1. *Gráfico 6: Top 10 Ciudades de Residencia con Mayor Registro de Muertes por Sobredosis (en general) en el Estado de Connecticut de los Estados Unidos (2012-2018):*

def Distribucion\_Geografica\_Muertes(df):

    # Filtra las filas con información de geolocalización no nula

    df\_geolocalizado = df.dropna(subset=['geolocalizacion\_ciudad\_residencia'])

    muertes\_por\_ciudad\_residencia = df\_geolocalizado['ciudad\_residencia'].value\_counts()

    # Visualiza las 10 principales ciudades de residencia y muerte

    top\_10\_residencia = muertes\_por\_ciudad\_residencia.head(10)

    fig, axs = plt.subplots(figsize=(12, 6))  # Solo un gráfico en lugar de 2x1

    #fig, axs = plt.subplots(2, 1, figsize=(12, 10)) # En caso de que quisiera hacer el grafico 2x1 (planeaba hacer una comparacion abajo con el top 10 de ciudades de muerte)

    axs.bar(top\_10\_residencia.index, top\_10\_residencia.values, color='darkslateblue')

    axs.set\_ylabel('Número de Muertes')

    axs.set\_title('Top 10 Ciudades de Residencia con Más Muertes por Sobredosis (en general) en el Estado de Connecticut')

    axs.tick\_params(axis='x', rotation=45)

    plt.tight\_layout()

    plt.show()

# Llama a la función para analizar la distribución geográfica

Distribucion\_Geografica\_Muertes(df)

Resultado:

A graph of a number of people

Description automatically generated with medium confidence

*Gráfico 6 Top 10 Ciudades de Residencia con Mayor Registro de Muertes por Sobredosis (en general) en el Estado de Connecticut de los Estados Unidos (2012-2018)*

Este gráfico 6, muestra las 10 principales ciudades de residencia con el mayor número de muertes por sobredosis en el estado de Connecticut durante el período de 2012 a 2018. Hartford se destaca significativamente como la ciudad con el mayor número de muertes, aproximadamente 300, lo que indica una alta incidencia de este trágico problema en la capital del estado.

Le siguen ciudades como Waterbury, Bridgeport y New Haven, todas con un número considerable de muertes, lo que subraya la amplitud de esta crisis en áreas urbanas clave. Aunque la mayoría de las ciudades en el top 10 superan las 100 muertes, es importante destacar que Torrington, la décima ciudad en la lista, aún registra una cantidad significativa de muertes por sobredosis.

# VI. CONCLUSIONES

En conclusión, se ha presenciado a la epidemia de fallecimientos a causa de sobredosis de drogas en América del Norte como un problema de salud pública que ha dejado una marca profunda en esta sociedad. Más allá de las cifras frías, existen historias desgarradoras de lucha, adicción y pérdida que han conmovido a comunidades enteras. Comprender este desafío es crucial para abordarlo de manera efectiva, y esto implica desentrañar las causas subyacentes y comprender las complejas ramificaciones que afectan a las personas y sus seres queridos.

El análisis minucioso de las tendencias y estadísticas más recientes relacionadas con las muertes por sobredosis se ha encontrado que es esencial para abordar esta crisis. Además, la tecnología, en particular el lenguaje de programación Python, ha desempeñado un papel fundamental en este análisis al permitir la recopilación, visualización y extracción de conocimientos a partir de grandes conjuntos de datos. Python ha demostrado ser una herramienta en el estudio sobre las muertes de la sobredosis de drogas, facilitando la identificación de patrones clave y así poder ofrecer soluciones efectivas.

Mientras se reflexiona sobre esta preocupante crisis, queda claro que enfrentarla es un desafío que requiere la colaboración de todos los sectores de la sociedad, desde profesionales de la salud y legisladores hasta comunidades locales y personas directamente afectadas. La lucha contra las sobredosis de drogas es un llamado a la acción colectiva.

Además, en un ámbito diferente pero estrechamente relacionado, Python ha emergido como un elemento esencial en una amplia variedad de campos dentro de la Tecnología de la Información (TI). Su capacidad para procesar datos, visualizar información y extraer conocimientos lo ha convertido en una herramienta versátil y valiosa. A través de ejemplos y casos de estudio, se ha ilustrado cómo Python se convierte en una solución preferida para enfrentar desafíos tecnológicos en una sociedad interconectada.

VII. referencias

1. A. Kumar and Supriya. P. Panda, “A Survey: How Python Pitches in IT-World,” in 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India: IEEE, 2019, pp. 248–251. doi: 10.1109/COMITCon.2019.8862251.
2. M. Xinyuan, W. Tao, and M. Kaigang, “Application of Python Parallel Computing in Online Identification of Thevenin Equivalent Parameters,” in 2020 IEEE 3rd Student Conference on Electrical Machines and Systems (SCEMS), Jinan, China: IEEE, Dec. 2020, pp. 20–23. doi: 10.1109/SCEMS48876.2020.9352354.
3. Y. Huang, S. Ao, and Y. Chen, “Application of Python Technology Based on Big Data: Does CSR Bring Economic Benefits or Penalties?” in 2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), Chengdu, China: IEEE, Apr. 2023, pp. 1–5. doi: 10.1109/ICCCBDA56900.2023.10154734.
4. A. Weiss and A. Elsherbeni, “Computational Performance of MATLAB and Python for Electromagnetic Applications,” in 2020 International Applied Computational Electromagnetics Society Symposium (ACES), Monterey, CA, USA: IEEE, 2020, pp. 1–2. doi: 10.23919/ACES49320.2020.9196078.
5. N. Chattopadhyay, S. Bhattacharya, R. Ghosh, and A. Paal, “Data Intrusion Detection with basic Python coding and prevention of other intrusive manifestation by the use of intrusion application,” in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC: IEEE, 2018, pp. 1094–1100. doi: 10.1109/IEMCON.2018.8614842.
6. H. Zhou and S. Wu, “Design of medical image enhancement algorithm based on Python,” in 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China: IEEE, Jan. 2021, pp. 482–485. doi: 10.1109/ICPECA51329.2021.9362581.
7. F. Liu and Z. Yang, “Design of VMware vSphere Automatic Operation and Maintenance System Based on Python,” in 2018 International Conference on Advanced Mechatronic Systems (ICAMechS), Zhengzhou: IEEE, 2018, pp. 283–286. doi: 10.1109/ICAMechS.2018.8506789.
8. C. Cabo, “Effectiveness of Flowcharting as a Scaffolding Tool to Learn Python,” in 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA: IEEE, 2018, pp. 1–7. doi: 10.1109/FIE.2018.8658891.
9. R. Kablaoui and A. Salman, “Machine Learning Models for Salary Prediction Dataset using Python,” in 2022 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates: IEEE, Nov. 2022, pp. 143–147. doi: 10.1109/ICECTA57148.2022.9990316.
10. K. Vissers, “Novel Neural Network Applications on New Python Enabled Platforms,” in 2018 International Conference on Field-Programmable Technology (FPT), Naha, Japan: IEEE, 2018, pp. 23–23. doi: 10.1109/FPT.2018.00011.
11. C. Califf RM, Woodcock J, Ostroff S. A proactive response to prescription. opioid abuse. N Engl J Med. 2016;374(15).
12. T. Informe Mundial de Drogas 2018: Crisis de opioides, abuso de medicamentos y niveles récord de opio y cocaína. Nueva York/Vienna/Ginebra/Nairobi.
13. C. Centers for Disease Control and Prevention. Vital signs: overdoses of prescription opioid pain relievers. United States, 1999–2008. MMWR Morbidity and mortality weekly report. 2011;60(43):1487– 92. PubMed PMID: 22048730.
14. S. Substance Abuse and Mental Health Services Administration. Results from the 2012 National Survey on Drug Use and Health: Summary of National Findings. Rockville, MD: Substance Abuse and Mental Health Services Administration, 2013.