

# NE\_inmates\_demographics\_2

March 8, 2021

## 0.1 Updated Dataset

The data being used was originally from inmate databases acquired from the Nebraska Department of Corrections Public Records. [https://dcs-inmatesearch.ne.gov/Corrections/COR\\_download.htm](https://dcs-inmatesearch.ne.gov/Corrections/COR_download.htm)

The Database was modified in NE\_inmates\_demographics\_1.ipynb. It had unnecessary rows deleted, extra rows added for analysis, and was prepared for exploration and visualization.

## 0.2 1. Load the Data

Load the data using Pandas.

Pandas read\_csv will load the data csv file as a Pandas DataFrame object.

```
[1]: import pandas as pd
import numpy as np
```

```
[2]: df = pd.read_csv('inmate_updatedClean_demographics.csv')
```

## 0.3 2. Check the First Few Rows of Data

The DataFrame's first five rows can be viewed using the .head() method

```
[3]: df.head()
```

```
[3]:
```

	ID	NUMBER	DATE OF BIRTH		RACE	DESC	GENDER	FACILITY	\
0	27642	1955-05-18	BLACK				MALE	NaN	
1	27739	1936-04-06	WHITE				MALE	NaN	
2	28160	1953-01-22	WHITE				MALE	NaN	
3	28350	1954-03-27	WHITE				MALE	NaN	
4	28789	1938-05-28	WHITE				MALE	NaN	

	SENTENCE	BEGIN DATE	MIN TERM/YEAR	MAX TERM/YEAR	INST	RELEASE DATE	ACTIVE	\
0		2015-12-01	90.0	95.0		2016-08-25	0	
1		1982-11-15	5.0	11.0		1986-07-23	0	
2		1981-06-09	1.0	5.0		1984-02-19	0	
3		1983-05-03	13.0	20.0		1992-03-22	0	
4		1987-04-03	4.0	15.0		1993-06-24	0	

	SENTENCE	BEGIN DATE	DT	DATE OF BIRTH	DT	SENTENCE	BEGIN AGE	CURRENT AGE	\
--	----------	------------	----	---------------	----	----------	-----------	-------------	---

0	2015-12-01	1955-05-18	60.580822	65.816438
1	1982-11-15	1936-04-06	46.641096	84.942466
2	1981-06-09	1953-01-22	28.397260	68.134247
3	1983-05-03	1954-03-27	29.120548	66.958904
4	1987-04-03	1938-05-28	48.882192	82.800000

	SENTENCE BEGIN YEAR
0	2015
1	1982
2	1981
3	1983
4	1987

### 0.4 3. Description of Data

The DataFrame `info()` method is used to see helpful descriptions of the data, such as the column name and number of rows. The ‘Non-Null Count’ is the number of rows that have a value for that particular column. The ‘Dtype’ is the data type found within each column. An `int64` is an integer, an object type is usually written text, and `datetime64` is a date time value.

```
[4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 68583 entries, 0 to 68582
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            68583 non-null  int64
1   DATE OF BIRTH                        68583 non-null  object
2   RACE DESC                            68583 non-null  object
3   GENDER                              68583 non-null  object
4   FACILITY                             11586 non-null  object
5   SENTENCE BEGIN DATE                  68583 non-null  object
6   MIN TERM/YEAR                        68583 non-null  float64
7   MAX TERM/YEAR                        68583 non-null  float64
8   INST RELEASE DATE                    63663 non-null  object
9   ACTIVE                              68583 non-null  int64
10  SENTENCE BEGIN DATE DT                68583 non-null  object
11  DATE OF BIRTH DT                      68583 non-null  object
12  SENTENCE BEGIN AGE                    68583 non-null  float64
13  CURRENT AGE                          68583 non-null  float64
14  SENTENCE BEGIN YEAR                  68583 non-null  int64
dtypes: float64(4), int64(3), object(8)
memory usage: 7.8+ MB
```

## 0.5 4. Creating An Active Inmate DataFrame

A new DataFrame containing only active inmates can be created by making a new DataFrame from the same info as before but selecting only inmates marked active.

```
[5]: dfActive = df[df.ACTIVE == 1]
```

```
[6]: dfActive.head()
```

```
[6]:
```

	ID NUMBER	DATE OF BIRTH	RACE DESC	GENDER	\
6	29333	1956-07-22	WHITE	MALE	
15	31404	1960-09-14	WHITE	MALE	
21	31928	1954-04-28	WHITE	MALE	
22	31936	1951-11-29	WHITE	MALE	
23	31978	1962-07-08	BLACK	MALE	

	FACILITY	SENTENCE BEGIN DATE	MIN TERM/YEAR	\
6	LINCOLN CORRECTIONAL CENTER	2017-02-03	110.0	
15	TECUMSEH STATE COR INSTITUTION	2017-01-03	83.0	
21	TECUMSEH STATE COR INSTITUTION	2001-11-30	30.0	
22	NEBRASKA STATE PENITENTIARY	2001-11-30	28.0	
23	OMAHA CORRECTIONAL CENTER	2016-03-21	82.0	

	MAX TERM/YEAR	INST RELEASE DATE	ACTIVE	SENTENCE BEGIN DATE	DT	\
6	126.0	NaN	1	2017-02-03		
15	132.0	NaN	1	2017-01-03		
21	30.0	NaN	1	2001-11-30		
22	28.0	NaN	1	2001-11-30		
23	100.0	NaN	1	2016-03-21		

	DATE OF BIRTH	DT	SENTENCE BEGIN AGE	CURRENT AGE	SENTENCE BEGIN YEAR
6	1956-07-22		60.578082	64.635616	2017
15	1960-09-14		56.342466	60.484932	2017
21	1954-04-28		47.624658	66.871233	2001
22	1951-11-29		50.038356	69.284932	2001
23	1962-07-08		53.739726	58.671233	2016

```
[7]: dfActive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 7406 entries, 6 to 67747
```

```
Data columns (total 15 columns):
```

#	Column	Non-Null Count	Dtype
0	ID NUMBER	7406 non-null	int64
1	DATE OF BIRTH	7406 non-null	object
2	RACE DESC	7406 non-null	object
3	GENDER	7406 non-null	object

```

4  FACILITY                7368 non-null  object
5  SENTENCE BEGIN DATE    7406 non-null  object
6  MIN TERM/YEAR          7406 non-null  float64
7  MAX TERM/YEAR          7406 non-null  float64
8  INST RELEASE DATE      2964 non-null  object
9  ACTIVE                 7406 non-null  int64
10 SENTENCE BEGIN DATE DT 7406 non-null  object
11 DATE OF BIRTH DT       7406 non-null  object
12 SENTENCE BEGIN AGE     7406 non-null  float64
13 CURRENT AGE            7406 non-null  float64
14 SENTENCE BEGIN YEAR    7406 non-null  int64
dtypes: float64(4), int64(3), object(8)
memory usage: 925.8+ KB

```

## 0.6 5. Creating a New DataFrame for each Decade

Each decade is separated out into their own DataFrame by selecting only inmates incarcerated within that time span.

```

[8]: #1980s decade
d1980 = (df['SENTENCE BEGIN YEAR'] > 1979) & (df['SENTENCE BEGIN YEAR'] <= 1989)
df1980 = pd.DataFrame(df.loc[d1980])

```

```

[9]: #1990s decade
d1990 = (df['SENTENCE BEGIN YEAR'] > 1989) & (df['SENTENCE BEGIN YEAR'] <= 1999)
df1990 = pd.DataFrame(df.loc[d1990])

```

```

[10]: #2000s decade
d2000 = (df['SENTENCE BEGIN YEAR'] > 1999) & (df['SENTENCE BEGIN YEAR'] <= 2009)
df2000 = pd.DataFrame(df.loc[d2000])

```

```

[11]: #2010s decade
d2010 = (df['SENTENCE BEGIN YEAR'] > 2009) & (df['SENTENCE BEGIN YEAR'] <= 2019)
df2010 = pd.DataFrame(df.loc[d2010])

```

## 0.7 6. Viewing the Data

### 0.7.1 6.1 Broad Viewing of the Full Inmate Numerical Data

A histogram can be plotted for each numerical attribute of the dataset.

A histogram counts how many instances have a given value. The `hist()` method performs this on the whole dataset, and it plots a histogram for each numerical attribute.

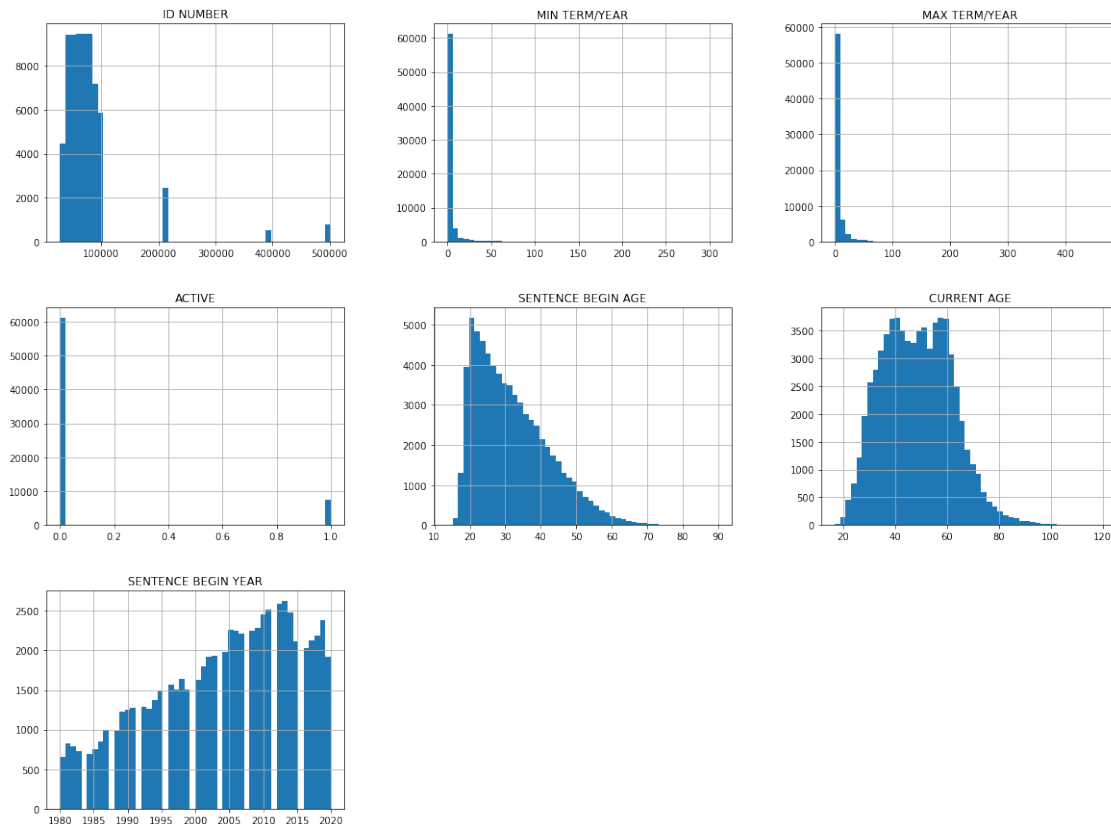
Matplotlib, the python visualization library, can use Jupyter as its backend for analysis. Plots are now shown within the notebook itself.

This gives us a broad idea of how numerical values are being distributed

```
[12]: # Import Matplotlib for plotting
import matplotlib.pyplot as plt
%matplotlib inline

df.hist(bins=50, figsize=(20,15))
```

```
[12]: array([[<AxesSubplot:title={'center':'ID NUMBER'}>,
<AxesSubplot:title={'center':'MIN TERM/YEAR'}>,
<AxesSubplot:title={'center':'MAX TERM/YEAR'}>],
[<AxesSubplot:title={'center':'ACTIVE'}>,
<AxesSubplot:title={'center':'SENTENCE BEGIN AGE'}>,
<AxesSubplot:title={'center':'CURRENT AGE'}>],
[<AxesSubplot:title={'center':'SENTENCE BEGIN YEAR'}>,
<AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```



Findings: 1. Most inmates begin their sentence between 20 and 30 years old, and as they get older it is less likely they will be incarcerated 2. The current age of inmates has higher averages around 40, 50, and 60 years of age. However, most of these individuals are no longer incarcerated. 3. The inmate sentence begin year steadily rises in numbers of prisoners for each decade

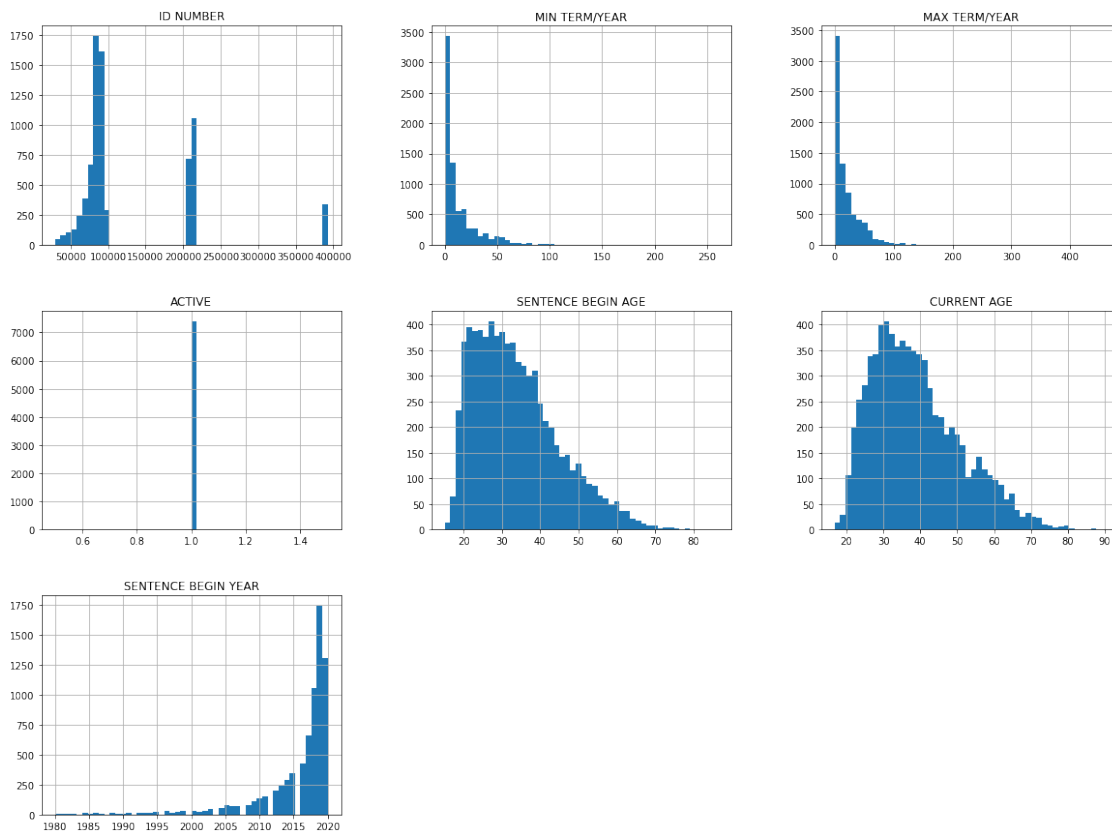
## 0.7.2 6.2 Broad Viewing of the Active Inmate Numerical Data

Repeated steps but with only the active inmates

```
[13]: # Import Matplotlib for plotting
import matplotlib.pyplot as plt
%matplotlib inline

dfActive.hist(bins=50, figsize=(20,15))
```

```
[13]: array([[<AxesSubplot:title={'center':'ID NUMBER'}>,
        <AxesSubplot:title={'center':'MIN TERM/YEAR'}>,
        <AxesSubplot:title={'center':'MAX TERM/YEAR'}>],
        [<AxesSubplot:title={'center':'ACTIVE'}>,
        <AxesSubplot:title={'center':'SENTENCE BEGIN AGE'}>,
        <AxesSubplot:title={'center':'CURRENT AGE'}>],
        [<AxesSubplot:title={'center':'SENTENCE BEGIN YEAR'}>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```



Findings: 1. Most inmates begin their sentence between 20 and 30 years old, and as they get older it is less likely they will be incarcerated. 2. The current age of inmates has higher averages around 30 to 40 years of age. However, there are more higher averages of inmates in their late 40s, 50s,

and 60s compared to the sentence begin age. 3. Most active inmates were arrested from 2015 to 2020.

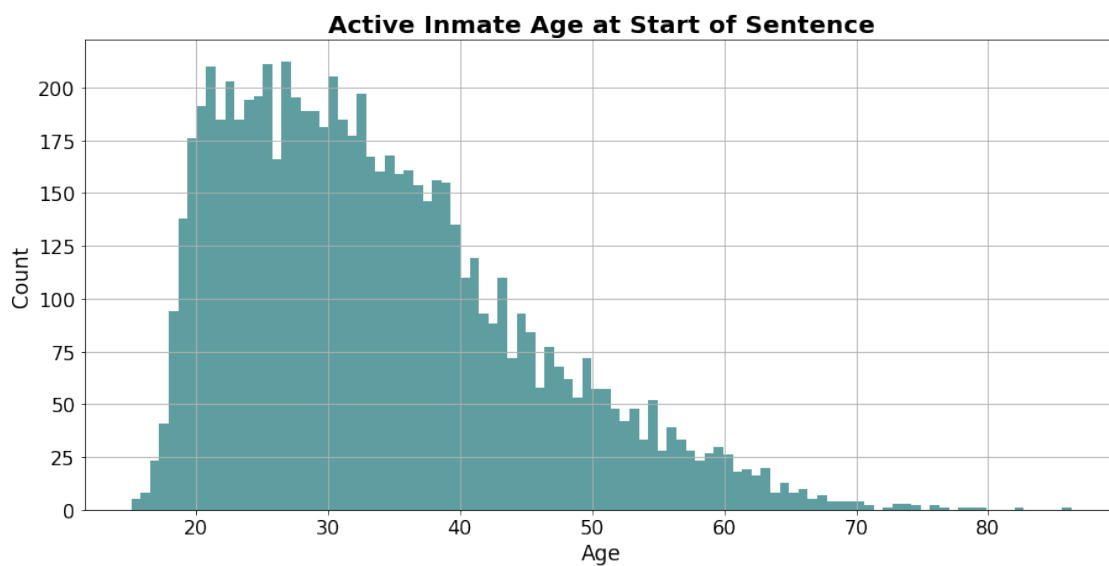
### 0.7.3 6.3 Closer Viewing of the Active Inmate Numerical Data

Let's zoom in on the sentence begin and current age of active prisoners by specifying a specific column in the DataFrame.

```
[14]: plt.xlabel('Age',fontsize=17)
plt.ylabel('Count',fontsize=17)
plt.title('Active Inmate Age at Start of Sentence', fontweight='bold',fontsize=20)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

dfActive['SENTENCE BEGIN AGE'].hist(bins=100, figsize=(15,7),color='cadetblue')
```

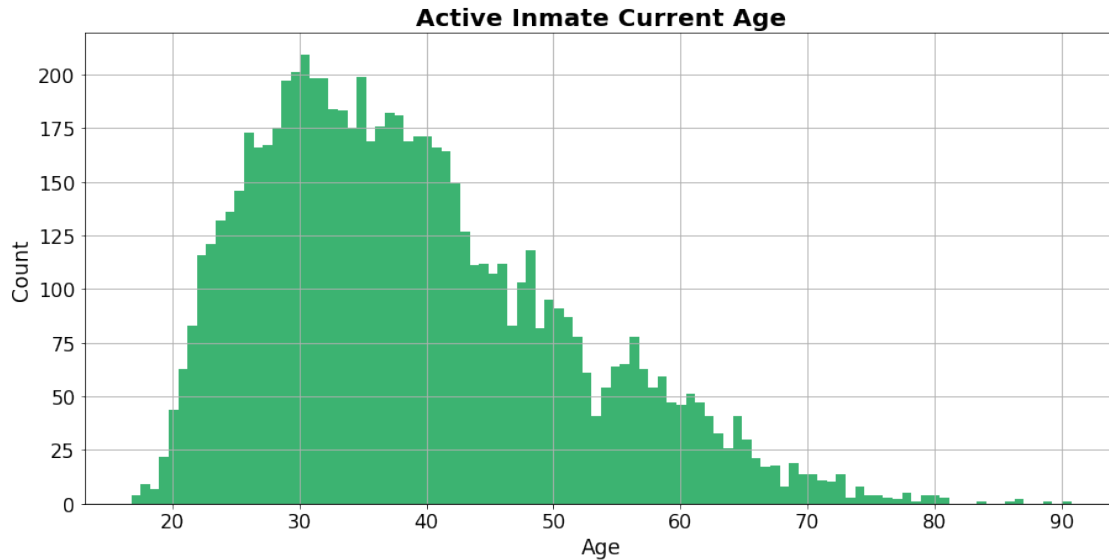
```
[14]: <AxesSubplot:title={'center':'Active Inmate Age at Start of Sentence'},
      xlabel='Age', ylabel='Count'>
```



```
[15]: plt.xlabel('Age',fontsize=17)
plt.ylabel('Count',fontsize=17)
plt.title('Active Inmate Current Age', fontweight='bold', fontsize=20)
plt.xticks(fontsize=16)
plt.yticks(fontsize=16)

dfActive['CURRENT AGE'].hist(bins=100, figsize=(15,7), color='mediumseagreen')
```

```
[15]: <AxesSubplot:title={'center':'Active Inmate Current Age'}, xlabel='Age',
      ylabel='Count'>
```



#### 0.7.4 6.4 Explore the Data

The DataFrame describe method helps us take a broad look at the numerical data. It shows the count, mean, standard deviation, maximum and minimum values

The noteworthy numbers are the mean, min, and max values for MIN / MAX TERM/YEAR, SENTENCE BEGIN AGE, and CURRENT AGE

```
[16]: df.describe()
```

```
[16]:
```

	ID NUMBER	MIN TERM/YEAR	MAX TERM/YEAR	ACTIVE \
count	68583.000000	68583.000000	68583.000000	68583.000000
mean	78225.156438	3.540134	6.155826	0.107986
std	62870.521305	8.013672	10.964725	0.310365
min	27642.000000	0.000000	0.000000	0.000000
25%	49873.500000	1.000000	2.000000	0.000000
50%	67095.000000	1.000000	3.000000	0.000000
75%	84303.500000	3.000000	6.000000	0.000000
max	502235.000000	310.000000	470.000000	1.000000

	SENTENCE BEGIN AGE	CURRENT AGE	SENTENCE BEGIN YEAR
count	68583.000000	68583.000000	68583.000000
mean	31.921698	48.629916	2003.959159
std	10.476343	13.543068	10.706607
min	13.660274	16.789041	1980.000000
25%	23.378082	38.032877	1996.000000



50%	29.873973	48.254795	2006.000000
75%	38.463014	58.468493	2013.000000
max	90.027397	121.213699	2020.000000

```
[17]: dfActive.describe()
```

```
[17]:
```

	ID NUMBER	MIN TERM/YEAR	MAX TERM/YEAR	ACTIVE \
count	7406.000000	7406.000000	7406.000000	7406.0
mean	126547.903592	13.743046	19.598569	1.0
std	80942.449352	18.944491	24.856465	0.0
min	29333.000000	0.000000	0.000000	1.0
25%	81661.250000	2.000000	4.000000	1.0
50%	88078.000000	6.000000	10.000000	1.0
75%	210342.750000	19.000000	27.000000	1.0
max	392481.000000	260.000000	463.000000	1.0

	SENTENCE BEGIN AGE	CURRENT AGE	SENTENCE BEGIN YEAR
count	7406.000000	7406.000000	7406.000000
mean	33.846597	38.914736	2015.589522
std	11.114111	12.024398	6.129352
min	15.095890	16.789041	1980.000000
25%	25.080137	29.682877	2014.000000
50%	31.963014	36.971233	2018.000000
75%	40.194521	46.189041	2019.000000
max	86.271233	90.715068	2020.000000

## 0.8 7. Exploring Value Counts and Value Averages on Full Data

Value counts is a pandas method that returns a list of the counts of every value in the column ordered by most common to least common. This is helpful to distinguish what aspects of inmates were the most common

Let's look at race first

```
[18]: df.value_counts('RACE DESC')
```

```
[18]: RACE DESC
```

WHITE	40208
BLACK	16679
HISPANIC	7760
NATIVE AMERICAN	3161
ASIAN	371
OTHER	331
	52
PACIFIC ISLANDER	20
HAWAIIAN ISLANDER	1

dtype: int64

The `normalize` parameters returns the frequencies of the values

```
[19]: df.value_counts('RACE_DESC', normalize=True)
```

```
[19]: RACE_DESC
      WHITE                0.586268
      BLACK                0.243194
      HISPANIC             0.113148
      NATIVE AMERICAN      0.046090
      ASIAN                0.005410
      OTHER                0.004826
                        0.000758
      PACIFIC ISLANDER     0.000292
      HAWAIIAN ISLANDER    0.000015
      dtype: float64
```

The `mean` function finds the average value from the selected column

```
[20]: df['SENTENCE_BEGIN_AGE'].mean()
```

```
[20]: 31.921698276201568
```

```
[21]: df['CURRENT_AGE'].mean()
```

```
[21]: 48.6299159961965
```

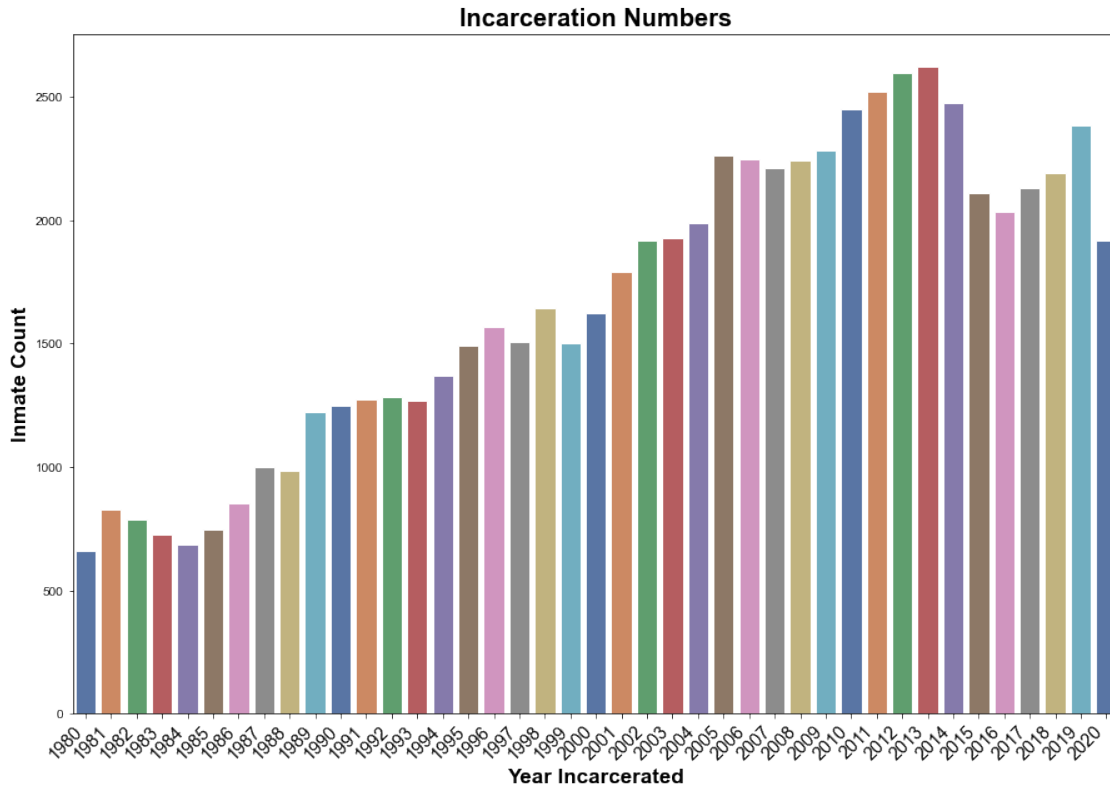
```
[22]: import seaborn as sns
```

```
[23]: fig_dims = (15, 10)
      fig, ax = plt.subplots(figsize=fig_dims)
      sns.set_theme(style="darkgrid")

      ax = sns.countplot(x="SENTENCE_BEGIN_YEAR", ax=ax
                        , data=df, palette=sns.color_palette('deep'))
      plt.xticks(rotation = 45, ha = 'right',fontsize='large')

      plt.xlabel('Year Incarcerated',fontsize=17, fontweight='bold')
      plt.ylabel('Inmate Count',fontsize=17, fontweight='bold')
      plt.title('Incarceration Numbers', fontweight='bold', fontsize=20)
```

```
[23]: Text(0.5, 1.0, 'Incarceration Numbers')
```

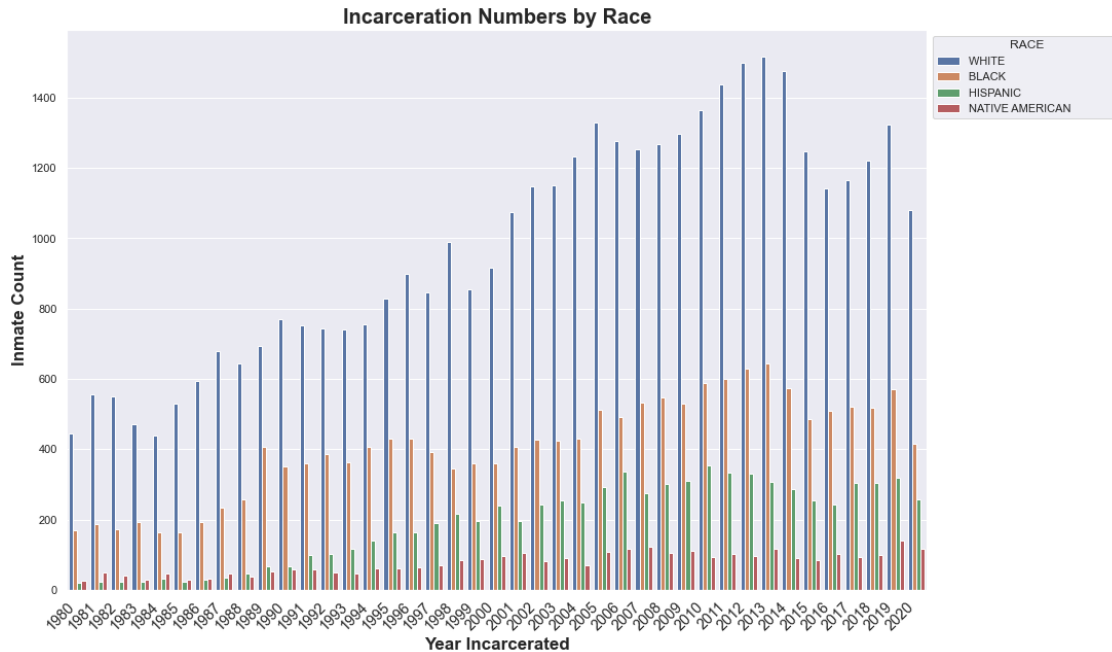


```
[24]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE_DESC", ax=ax
                  , data=df,
                  hue_order=df["RACE_DESC"].value_counts().iloc[:4].index)
plt.xticks(rotation = 45, ha = 'right',fontsize='large')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))

plt.xlabel('Year Incarcerated',fontsize=17, fontweight='bold')
plt.ylabel('Inmate Count',fontsize=17, fontweight='bold')
plt.title('Incarceration Numbers by Race', fontweight='bold', fontsize=20)
```

```
[24]: Text(0.5, 1.0, 'Incarceration Numbers by Race')
```



## 0.9 8. Exploring 1980s Value Counts, Value Averages, and Visualizations

### 0.9.1 8.1 General info

8498 inmates are listed to be incarcerated in 1980

```
[25]: df1980.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8498 entries, 1 to 57899
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                            8498 non-null   int64
1   DATE OF BIRTH                        8498 non-null   object
2   RACE DESC                            8498 non-null   object
3   GENDER                              8498 non-null   object
4   FACILITY                             637 non-null    object
5   SENTENCE BEGIN DATE                  8498 non-null   object
6   MIN TERM/YEAR                        8498 non-null   float64
7   MAX TERM/YEAR                        8498 non-null   float64
8   INST RELEASE DATE                    8427 non-null   object
9   ACTIVE                              8498 non-null   int64
10  SENTENCE BEGIN DATE DT                8498 non-null   object
11  DATE OF BIRTH DT                      8498 non-null   object
12  SENTENCE BEGIN AGE                    8498 non-null   float64
13  CURRENT AGE                          8498 non-null   float64
```

```
14 SENTENCE BEGIN YEAR      8498 non-null   int64
dtypes: float64(4), int64(3), object(8)
memory usage: 1.0+ MB
```

### 0.9.2 8.2 Race

Most inmates are white

```
[26]: df1980.value_counts('RACE DESC', normalize=True)
```

```
[26]: RACE DESC
      WHITE                0.659802
      BLACK                0.251236
      NATIVE AMERICAN      0.045187
      HISPANIC              0.038127
      OTHER                0.005413
      ASIAN                0.000118
      ASIAN                0.000118
```

```
dtype: float64
```

### 0.9.3 8.3 Sentence Begin

```
[27]: df1980['SENTENCE BEGIN AGE'].mean()
```

```
[27]: 28.545750329650527
```

Most inmates were incarcerated in the last 4 years of the 1980s

```
[28]: df1980.value_counts('SENTENCE BEGIN YEAR', normalize=True).head(10)
```

```
[28]: SENTENCE BEGIN YEAR
      1989    0.144034
      1987    0.117557
      1988    0.116027
      1986    0.100612
      1981    0.097435
      1982    0.092610
      1985    0.088021
      1983    0.085197
      1984    0.080960
      1980    0.077548
dtype: float64
```

### 0.9.4 8.4 Sentencing Min/Max Term

```
[29]: df1980['MIN TERM/YEAR'].mean()
```

```
[29]: 2.5947281713344315
```

```
[30]: df1980.value_counts('MIN TERM/YEAR', normalize=True).head(10)
```

```
[30]: MIN TERM/YEAR
1.0    0.463756
0.0    0.200753
2.0    0.097199
3.0    0.077783
4.0    0.033302
5.0    0.032831
6.0    0.023182
10.0   0.016945
8.0    0.008590
7.0    0.008237
dtype: float64
```

```
[31]: df1980['MAX TERM/YEAR'].mean()
```

```
[31]: 5.680630736643916
```

```
[32]: df1980.value_counts('MAX TERM/YEAR', normalize=True).head(10)
```

```
[32]: MAX TERM/YEAR
1.0    0.260061
3.0    0.181572
2.0    0.125441
5.0    0.125441
4.0    0.071782
10.0   0.041892
6.0    0.041422
8.0    0.022711
7.0    0.021770
15.0   0.014474
dtype: float64
```

### 0.9.5 8.5 Graphs

Seaborn is a data visualization library based on matplotlib.

```
[33]: import seaborn as sns
```

A countplot is a histogram for categorical variables. The x axis is the sentence begin year, and the plot counts those variables to create the y axis.

x is the year, y is the inmate count for 1980s

```
[34]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")
```

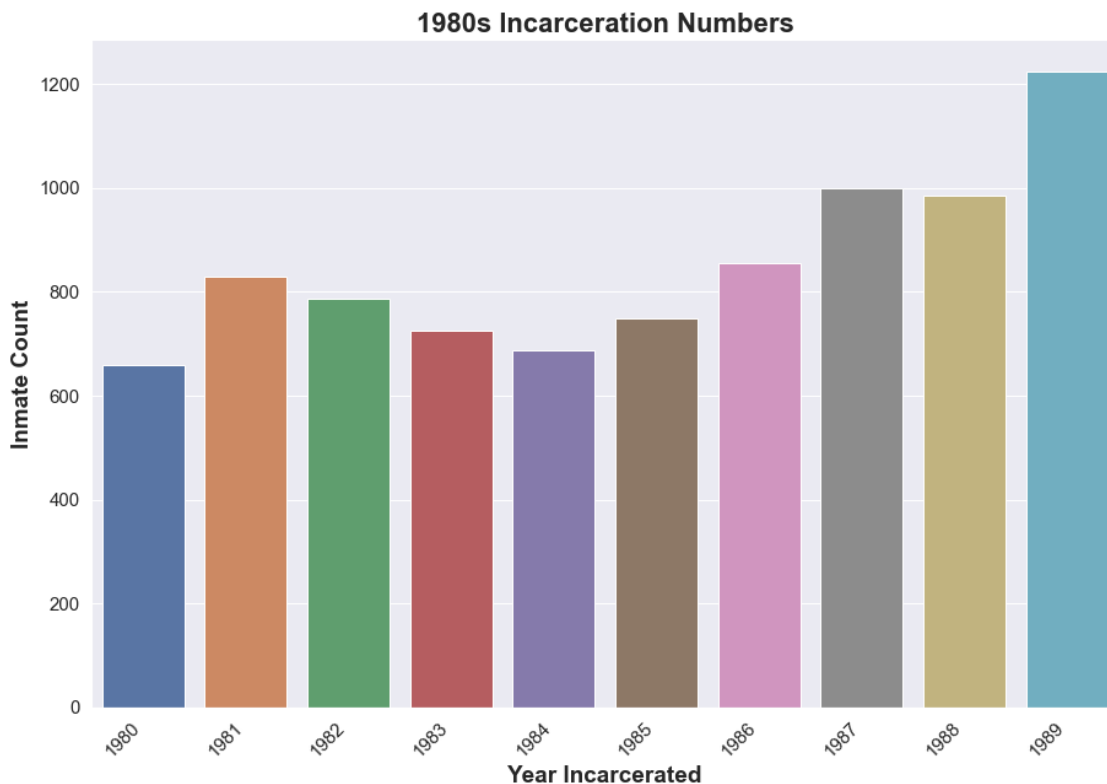
```

ax = sns.countplot(x="SENTENCE BEGIN YEAR", ax=ax
                  , data=df1980)
plt.xticks(rotation = 45, ha = 'right',fontsize=15)
plt.yticks(fontsize=15)

plt.xlabel('Year Incarcerated',fontsize=19, fontweight='bold')
plt.ylabel('Inmate Count',fontsize=19, fontweight='bold')
plt.title('1980s Incarceration Numbers', fontweight='bold', fontsize=22)

```

[34]: Text(0.5, 1.0, '1980s Incarceration Numbers')



The hue is the count broken down by another variable, in this case, race.

You can notice a gradual rise in the number of black prisoners from 1985 to 1989

```

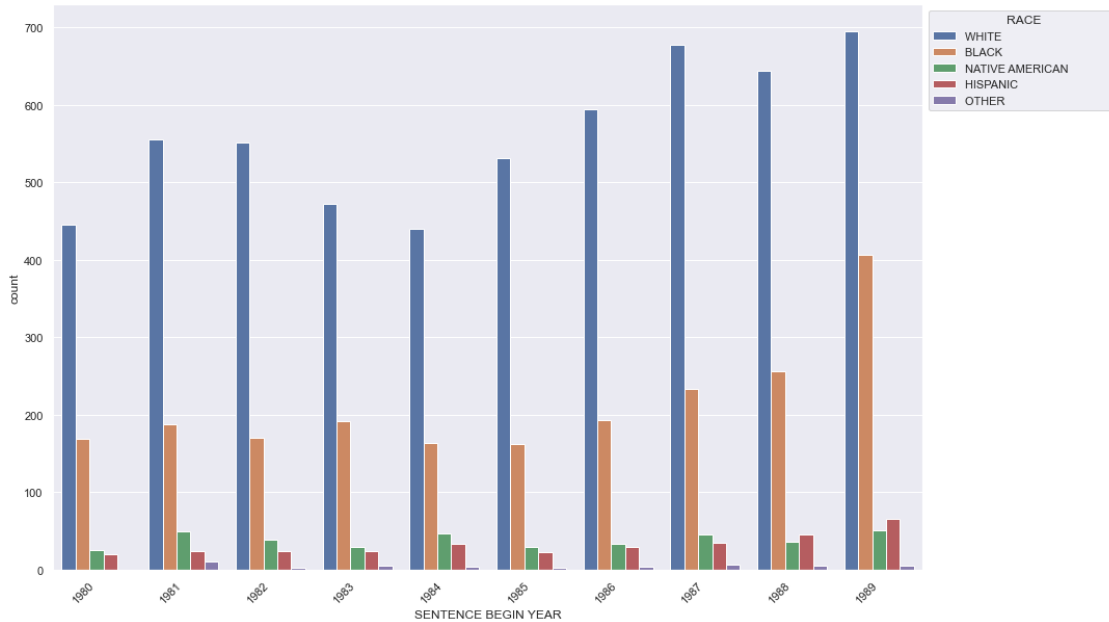
[35]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE DESC", ax=ax
                  , data=df1980,
                  hue_order=df1980["RACE DESC"].value_counts().iloc[:5].index)

```

```
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))
```

[35]: <matplotlib.legend.Legend at 0x7f855fb49fd0>



## 0.10 9. Exploring 1990s Value Counts, Value Averages, and Visualizations

### 0.10.1 9.1 General Info

14,154 inmates started their incarceration in the 1990s

[36]: df1990.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14154 entries, 16 to 59270
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID NUMBER              14154 non-null  int64
1   DATE OF BIRTH          14154 non-null  object
2   RACE DESC              14154 non-null  object
3   GENDER                14154 non-null  object
4   FACILITY               287 non-null   object
5   SENTENCE BEGIN DATE   14154 non-null  object
6   MIN TERM/YEAR         14154 non-null  float64
7   MAX TERM/YEAR         14154 non-null  float64
8   INST RELEASE DATE     14002 non-null  object
9   ACTIVE                14154 non-null  int64
```



```

10 SENTENCE BEGIN DATE DT 14154 non-null object
11 DATE OF BIRTH DT       14154 non-null object
12 SENTENCE BEGIN AGE     14154 non-null float64
13 CURRENT AGE            14154 non-null float64
14 SENTENCE BEGIN YEAR    14154 non-null int64
dtypes: float64(4), int64(3), object(8)
memory usage: 1.7+ MB

```

### 0.10.2 9.2 Race

The largest difference since the 1980s is the 10% of inmates being Hispanic

```
[37]: df1990.value_counts('RACE_DESC', normalize=True)
```

```

[37]: RACE_DESC
WHITE                0.577646
BLACK                0.270242
HISPANIC             0.102515
NATIVE AMERICAN     0.045217
ASIAN                0.003603
OTHER                0.000777
dtype: float64

```

### 0.10.3 9.3 Sentence Begin

The begin average age is slightly older

```
[38]: df1990['SENTENCE BEGIN AGE'].mean()
```

```
[38]: 30.15704491300193
```

```
[39]: df1990.value_counts('SENTENCE BEGIN YEAR', normalize=True).head(10)
```

```

[39]: SENTENCE BEGIN YEAR
1998    0.116292
1996    0.110711
1997    0.106330
1999    0.106118
1995    0.105341
1994    0.096651
1992    0.090716
1991    0.090010
1993    0.089515
1990    0.088314
dtype: float64

```

#### 0.10.4 9.4 Sentencing Min/Max Term

```
[40]: df1990['MIN TERM/YEAR'].mean()
```

```
[40]: 2.926451886392539
```

```
[41]: df1990.value_counts('MIN TERM/YEAR', normalize=True).head(10)
```

```
[41]: MIN TERM/YEAR
1.0      0.408365
2.0      0.149498
0.0      0.146531
3.0      0.098064
4.0      0.054825
5.0      0.032994
6.0      0.025505
10.0     0.016391
8.0      0.010810
7.0      0.010386
dtype: float64
```

```
[42]: df1990['MAX TERM/YEAR'].mean()
```

```
[42]: 5.917337855023315
```

```
[43]: df1990.value_counts('MAX TERM/YEAR', normalize=True).head(10)
```

```
[43]: MAX TERM/YEAR
1.0      0.194645
3.0      0.168574
5.0      0.154727
2.0      0.138123
4.0      0.097075
6.0      0.050586
10.0     0.040271
8.0      0.025646
7.0      0.019853
20.0     0.015119
dtype: float64
```

Facility is still largely missing from the data

```
[44]: df1990.value_counts('FACILITY', normalize=True).head(10)
```

```
[44]: FACILITY
COMMUNITY CORRECTIONS-LINCOLN      0.212544
NEBRASKA STATE PENITENTIARY        0.198606
LINCOLN CORRECTIONAL CENTER        0.160279
```

TECUMSEH STATE COR INSTITUTION	0.139373
OMAHA CORRECTIONAL CENTER	0.114983
HASTINGS CORRECTIONAL CENTER	0.069686
COMMUNITY CORRECTIONS-OMAHA	0.041812
DIAGNOSTIC & EVALUATION CENTER	0.038328
NEBRASKA CORR CENTER FOR WOMEN	0.017422
WORK ETHIC CAMP	0.006969

dtype: float64

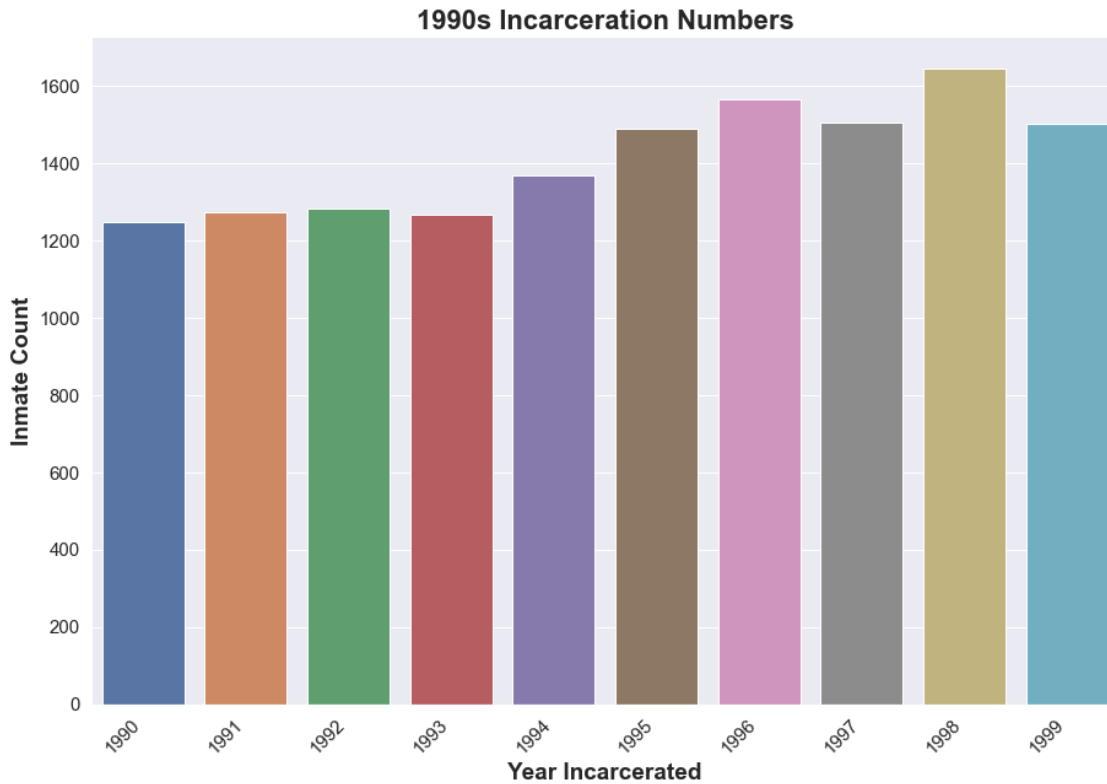
### 0.10.5 9.5 Graphs

```
[45]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", ax=ax
                  , data=df1990)
plt.xticks(rotation = 45, ha = 'right',fontsize=15)
plt.yticks(fontsize=15)

plt.xlabel('Year Incarcerated',fontsize=19, fontweight='bold')
plt.ylabel('Inmate Count',fontsize=19, fontweight='bold')
plt.title('1990s Incarceration Numbers', fontweight='bold', fontsize=22)

[45]: Text(0.5, 1.0, '1990s Incarceration Numbers')
```

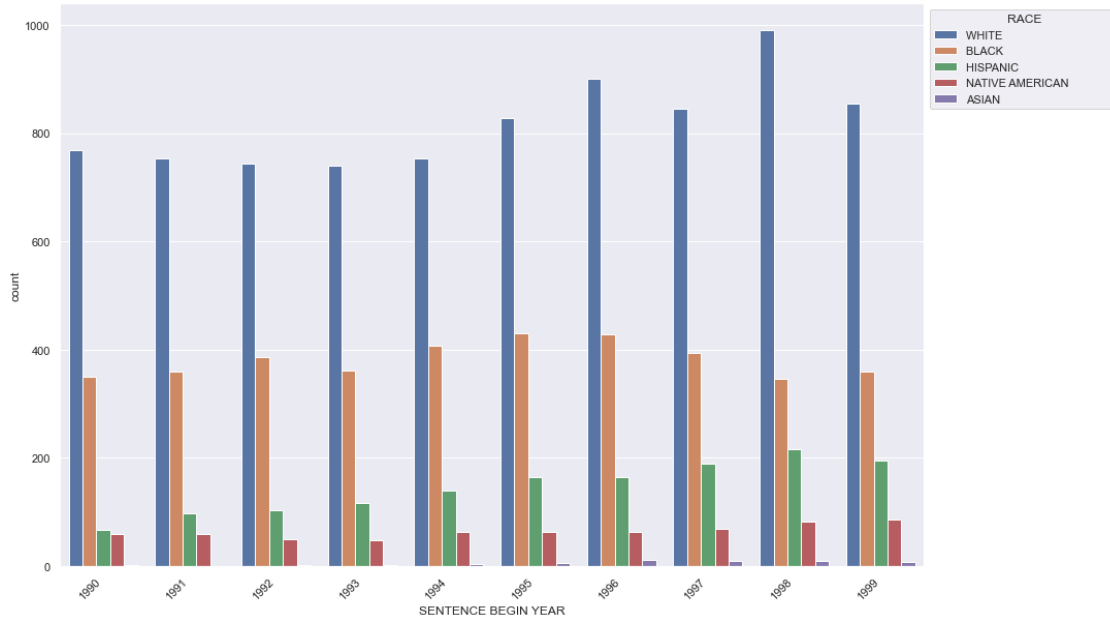


The amount of hispanics incarcerated each year increases, while black inmates remains largely around the same

```
[46]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE_DESC", ax=ax
                  , data=df1990,
                  hue_order=df1990["RACE_DESC"].value_counts().iloc[:5].index)
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))
```

```
[46]: <matplotlib.legend.Legend at 0x7f855db669a0>
```



## 0.11 10. Exploring 2000s Value Counts, Value Averages, and Visualizations

### 0.11.1 10.1 General info

20501 inmates are listed to be incarcerated in 2000s

```
[47]: df2000.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20501 entries, 21 to 68504
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   ID NUMBER                             20501 non-null  int64
1   DATE OF BIRTH                         20501 non-null  object
2   RACE DESC                             20501 non-null  object
3   GENDER                                20501 non-null  object
4   FACILITY                               1217 non-null   object
5   SENTENCE BEGIN DATE                   20501 non-null  object
6   MIN TERM/YEAR                         20501 non-null  float64
7   MAX TERM/YEAR                         20501 non-null  float64
8   INST RELEASE DATE                     20151 non-null  object
9   ACTIVE                                20501 non-null  int64
10  SENTENCE BEGIN DATE DT                 20501 non-null  object
11  DATE OF BIRTH DT                       20501 non-null  object
12  SENTENCE BEGIN AGE                     20501 non-null  float64
13  CURRENT AGE                           20501 non-null  float64
14  SENTENCE BEGIN YEAR                    20501 non-null  int64
```

```
dtypes: float64(4), int64(3), object(8)
memory usage: 2.5+ MB
```

### 0.11.2 10.2 Race

The hispanic makeup of inmates rises again

```
[48]: df2000.value_counts('RACE_DESC', normalize=True)
```

```
[48]: RACE_DESC
      WHITE                0.582801
      BLACK                0.227355
      HISPANIC             0.131555
      NATIVE AMERICAN      0.049071
      ASIAN                0.006780
      OTHER                0.002195
                        0.000146
      PACIFIC ISLANDER     0.000098
dtype: float64
```

### 0.11.3 10.3 Sentence Begin

the sentence begin age rises a few years again

```
[49]: df2000['SENTENCE_BEGIN_AGE'].mean()
```

```
[49]: 32.253207294264755
```

```
[50]: df2000.value_counts('SENTENCE_BEGIN_YEAR', normalize=True).head(10)
```

```
[50]: SENTENCE_BEGIN_YEAR
      2009    0.111312
      2005    0.110336
      2006    0.109604
      2008    0.109507
      2007    0.107946
      2004    0.096922
      2003    0.094142
      2002    0.093605
      2001    0.087459
      2000    0.079167
dtype: float64
```

### 0.11.4 10.4 Sentence Min/Max Term

```
[51]: df2000['MIN_TERM/YEAR'].mean()
```

```
[51]: 3.3803229110775086
```

```
[52]: df2000.value_counts('MIN TERM/YEAR', normalize=True).head(10)
```

```
[52]: MIN TERM/YEAR
1.0    0.455197
2.0    0.124872
0.0    0.115409
3.0    0.092142
4.0    0.053997
5.0    0.035706
6.0    0.022096
10.0   0.017755
8.0    0.013804
7.0    0.008244
dtype: float64
```

```
[53]: df2000['MAX TERM/YEAR'].mean()
```

```
[53]: 5.8596653821764795
```

```
[54]: df2000.value_counts('MAX TERM/YEAR', normalize=True).head(10)
```

```
[54]: MAX TERM/YEAR
1.0    0.198332
3.0    0.154968
2.0    0.143700
5.0    0.131018
4.0    0.104580
6.0    0.047510
10.0   0.036096
8.0    0.028974
0.0    0.020877
7.0    0.019804
dtype: float64
```

### 0.11.5 10.5 Graphs

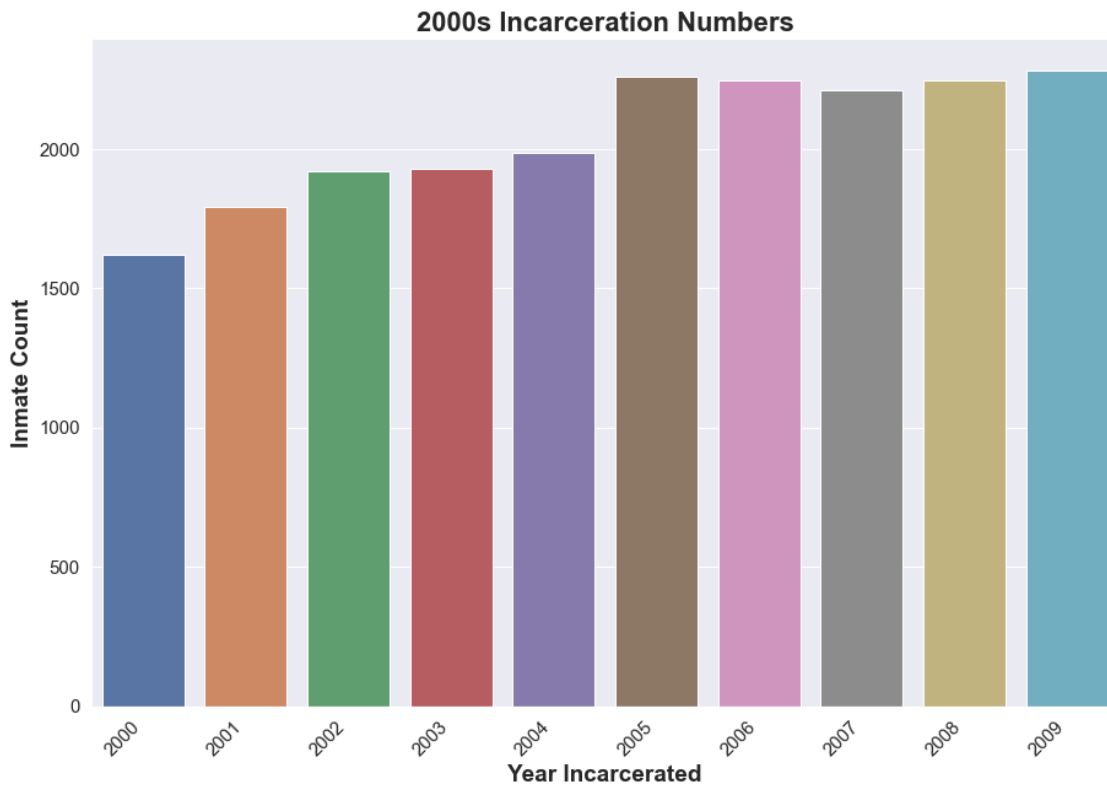
```
[55]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", ax=ax
                  , data=df2000)
plt.xticks(rotation = 45, ha = 'right', fontsize=15)
plt.yticks(fontsize=15)

plt.xlabel('Year Incarcerated', fontsize=19, fontweight='bold')
plt.ylabel('Inmate Count', fontsize=19, fontweight='bold')
```

```
plt.title('2000s Incarceration Numbers', fontweight='bold', fontsize=22)
```

```
[55]: Text(0.5, 1.0, '2000s Incarceration Numbers')
```



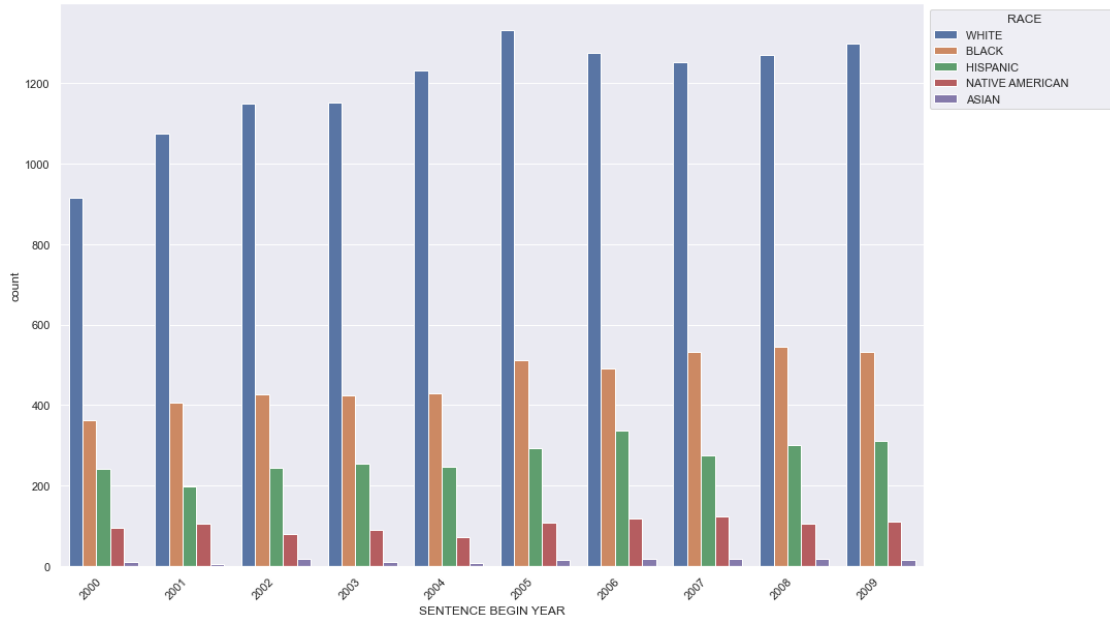
The percentage of black inmates steadily rises over the 2000s. Hispanic inmate numbers stay about the same.

```
[56]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE DESC", ax=ax
                  , data=df2000,
                  hue_order=df2000["RACE DESC"].value_counts().iloc[:5].index)
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))
```

```
[56]: <matplotlib.legend.Legend at 0x7f855dbb29d0>
```





## 0.12 11. Exploring 2010s Value Counts, Value Averages, and Visualizations

### 0.12.1 11.1 General info

23514 inmates are listed to be incarcerated in 2010s

### 0.12.2 2010s

```
[57]: df2010.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23514 entries, 0 to 68582
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID NUMBER              23514 non-null  int64
1   DATE OF BIRTH          23514 non-null  object
2   RACE DESC              23514 non-null  object
3   GENDER                 23514 non-null  object
4   FACILITY               7546 non-null   object
5   SENTENCE BEGIN DATE    23514 non-null  object
6   MIN TERM/YEAR          23514 non-null  float64
7   MAX TERM/YEAR          23514 non-null  float64
8   INST RELEASE DATE      20546 non-null  object
9   ACTIVE                 23514 non-null  int64
10  SENTENCE BEGIN DATE DT  23514 non-null  object
11  DATE OF BIRTH DT       23514 non-null  object
12  SENTENCE BEGIN AGE     23514 non-null  float64
```

```

13  CURRENT AGE                23514 non-null float64
14  SENTENCE BEGIN YEAR        23514 non-null int64
dtypes: float64(4), int64(3), object(8)
memory usage: 2.9+ MB

```

### 0.12.3 11.2 Race

Black and Hispanic inmates percentages continue to rise.

```
[58]: df2010.value_counts('RACE_DESC', normalize=True)
```

```

[58]: RACE_DESC
WHITE                0.569746
BLACK                0.239942
HISPANIC             0.128859
NATIVE AMERICAN     0.043166
OTHER                0.008591
ASIAN                0.006975
                    0.001956
PACIFIC ISLANDER    0.000723
HAWAIIAN ISLANDER   0.000043
dtype: float64

```

### 0.12.4 11.3 Sentence Begin

Average inmate begin age rises again

```
[59]: df2010['SENTENCE BEGIN AGE'].mean()
```

```
[59]: 33.66685052682106
```

```
[60]: df2010.value_counts('SENTENCE BEGIN YEAR', normalize=True).head(10)
```

```

[60]: SENTENCE BEGIN YEAR
2013    0.111636
2012    0.110360
2011    0.107170
2014    0.105214
2010    0.104193
2019    0.101344
2018    0.093264
2017    0.090584
2015    0.089776
2016    0.086459
dtype: float64

```

### 0.12.5 11.4 Sentence Min/Max Term

```
[61]: df2010['MIN TERM/YEAR'].mean()
```

```
[61]: 4.360678744577698
```

```
[62]: df2010.value_counts('MIN TERM/YEAR', normalize=True).head(10)
```

```
[62]: MIN TERM/YEAR
1.0      0.300034
0.0      0.174109
2.0      0.136642
3.0      0.109892
4.0      0.061240
5.0      0.046568
6.0      0.032066
10.0     0.025176
8.0      0.018457
7.0      0.011568
dtype: float64
```

```
[63]: df2010['MAX TERM/YEAR'].mean()
```

```
[63]: 6.705962405375521
```

```
[64]: df2010.value_counts('MAX TERM/YEAR', normalize=True).head(10)
```

```
[64]: MAX TERM/YEAR
1.0      0.156758
3.0      0.123118
2.0      0.119971
0.0      0.111508
5.0      0.106107
4.0      0.084460
6.0      0.055711
10.0     0.038743
8.0      0.035511
7.0      0.020924
dtype: float64
```

### 0.12.6 11.5 Graphs

```
[65]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", ax=ax)
```

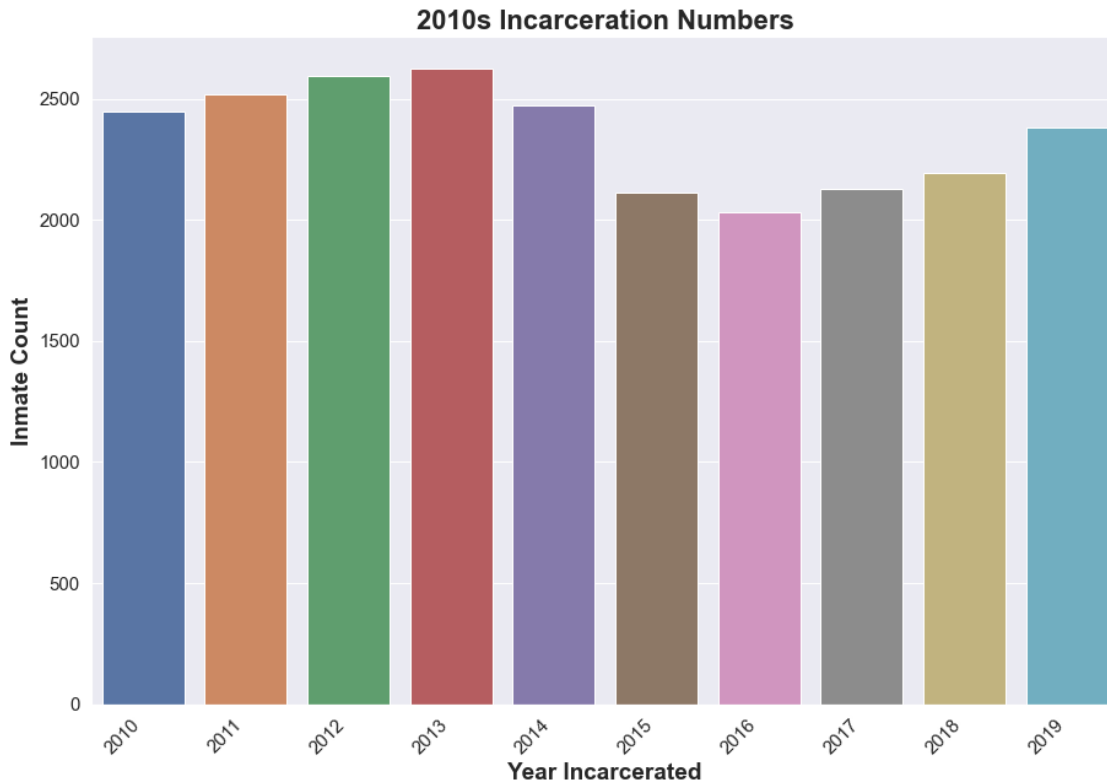
```

        , data=df2010)
plt.xticks(rotation = 45, ha = 'right',fontsize=15)
plt.yticks(fontsize=15)

plt.xlabel('Year Incarcerated',fontsize=19, fontweight='bold')
plt.ylabel('Inmate Count',fontsize=19, fontweight='bold')
plt.title('2010s Incarceration Numbers', fontweight='bold', fontsize=22)

```

[65]: Text(0.5, 1.0, '2010s Incarceration Numbers')



The percentage of black inmates steadily rises over the 2010s. Hispanic inmate numbers stay about the same.

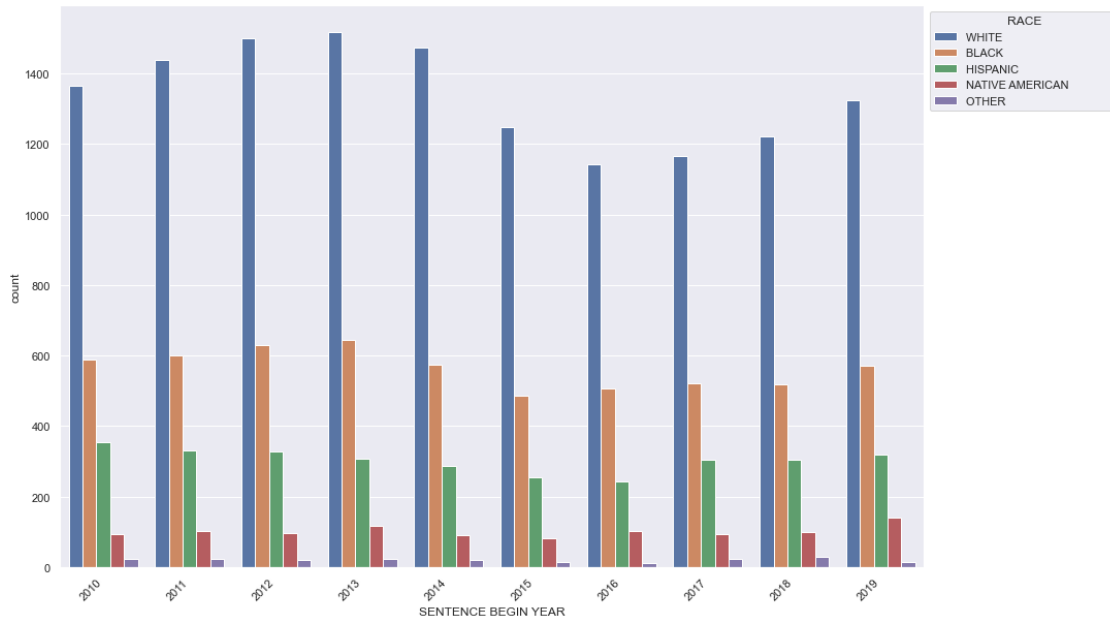
```

[66]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE DESC", ax=ax
                  , data=df2010,
                  hue_order=df2010["RACE DESC"].value_counts().iloc[:5].index)
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))

```

[66]: <matplotlib.legend.Legend at 0x7f8560113670>



## 0.13 12. Exploring Active Inmate Value Counts, Value Averages, and Visualizations

### 0.13.1 12.1 General info

7406 inmates are listed to be incarcerated in Actives

### 0.13.2 Actives

```
[67]: dfActive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7406 entries, 6 to 67747
Data columns (total 15 columns):
#   Column              Non-Null Count  Dtype
---  -
0   ID NUMBER           7406 non-null   int64
1   DATE OF BIRTH       7406 non-null   object
2   RACE DESC           7406 non-null   object
3   GENDER              7406 non-null   object
4   FACILITY            7368 non-null   object
5   SENTENCE BEGIN DATE 7406 non-null   object
6   MIN TERM/YEAR       7406 non-null   float64
7   MAX TERM/YEAR       7406 non-null   float64
8   INST RELEASE DATE   2964 non-null   object
9   ACTIVE              7406 non-null   int64
```

```

10 SENTENCE BEGIN DATE DT 7406 non-null object
11 DATE OF BIRTH DT       7406 non-null object
12 SENTENCE BEGIN AGE     7406 non-null float64
13 CURRENT AGE            7406 non-null float64
14 SENTENCE BEGIN YEAR    7406 non-null int64
dtypes: float64(4), int64(3), object(8)
memory usage: 925.8+ KB

```

### 0.13.3 11.2 Race

Black and Hispanic inmates percentages are high compared to population

```
[68]: dfActive.value_counts('RACE_DESC', normalize=True)
```

```

[68]: RACE_DESC
WHITE                0.534837
BLACK                0.261140
HISPANIC             0.140697
NATIVE AMERICAN     0.045234
ASIAN                0.008777
OTHER                0.008237
PACIFIC ISLANDER    0.000540
dtype: float64

```

### 0.13.4 11.3 Sentence Begin

Average inmate begin age rises again

```
[69]: dfActive['SENTENCE BEGIN AGE'].mean()
```

```
[69]: 33.84659679859728
```

most were incarcerated in the 2010s

```
[70]: dfActive.value_counts('SENTENCE BEGIN YEAR', normalize=True).head(10)
```

```

[70]: SENTENCE BEGIN YEAR
2019    0.235485
2020    0.176344
2018    0.143127
2017    0.088847
2016    0.057656
2015    0.046179
2014    0.038752
2013    0.032541
2012    0.027410
2011    0.020254
dtype: float64

```

### 0.13.5 11.4 Current Age

```
[71]: dfActive['CURRENT AGE'].mean()
```

```
[71]: 38.914736293046474
```

### 0.13.6 11.5 Sentence Min/Max Term

```
[72]: dfActive['MIN TERM/YEAR'].mean()
```

```
[72]: 13.743046178773968
```

```
[73]: dfActive.value_counts('MIN TERM/YEAR', normalize=True).head(10)
```

```
[73]: MIN TERM/YEAR
0.0      0.161626
3.0      0.088847
4.0      0.064677
2.0      0.063732
10.0     0.061572
5.0      0.059411
6.0      0.051175
20.0     0.043748
8.0      0.036862
15.0     0.028220
dtype: float64
```

```
[74]: dfActive['MAX TERM/YEAR'].mean()
```

```
[74]: 19.598568728058332
```

```
[75]: dfActive.value_counts('MAX TERM/YEAR', normalize=True).head(10)
```

```
[75]: MAX TERM/YEAR
0.0      0.098974
2.0      0.059006
3.0      0.056441
20.0     0.055631
10.0     0.053335
6.0      0.049689
8.0      0.047394
5.0      0.044558
4.0      0.035377
12.0     0.034161
dtype: float64
```

### 0.13.7 11.6 Facility

```
[76]: dfActive.value_counts('FACILITY', normalize=True)
```

```
[76]: FACILITY
NEBRASKA STATE PENITENTIARY      0.208605
COMMUNITY CORRECTIONS-LINCOLN    0.203176
OMAHA CORRECTIONAL CENTER        0.150651
TECUMSEH STATE COR INSTITUTION   0.148751
LINCOLN CORRECTIONAL CENTER      0.076140
DIAGNOSTIC & EVALUATION CENTER   0.065961
COMMUNITY CORRECTIONS-OMAHA      0.054425
NEBRASKA CORR CENTER FOR WOMEN   0.047503
WORK ETHIC CAMP                 0.034338
NEBRASKA CORR YOUTH FACILTY      0.010451
dtype: float64
```

### 0.13.8 11.5 Graphs

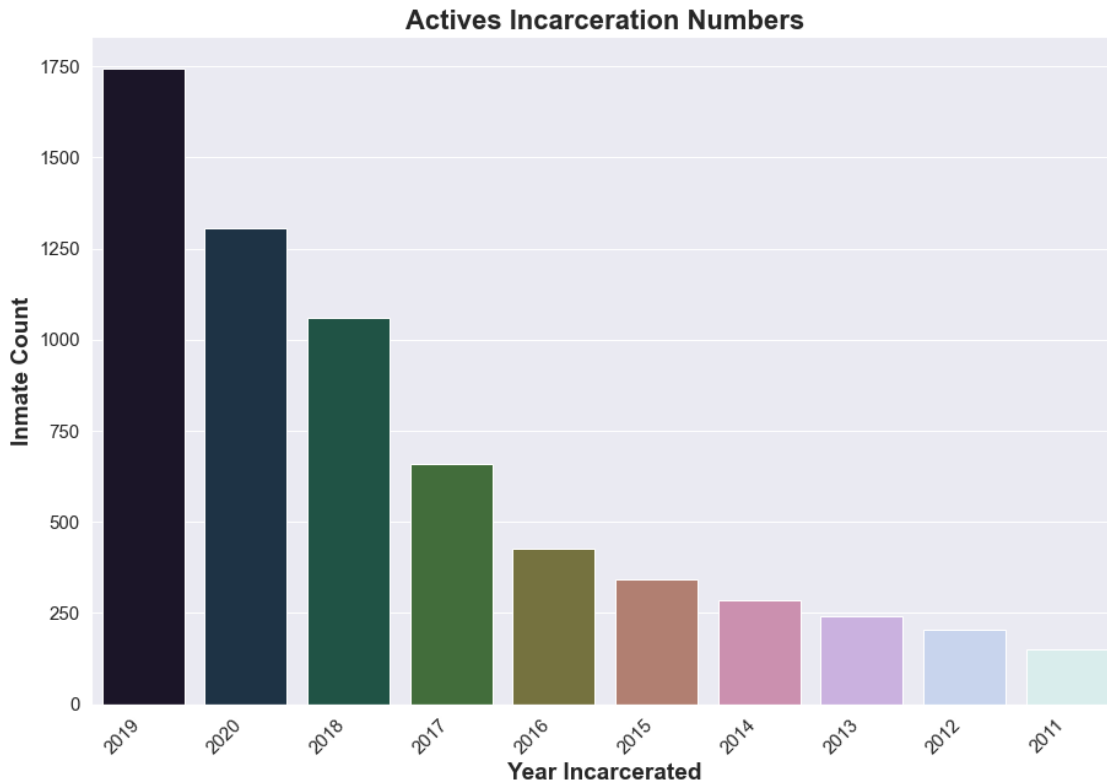
```
[77]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", ax=ax,
                   order=dfActive["SENTENCE BEGIN YEAR"].value_counts().iloc[:
→10].index
                   , data=dfActive, palette='cubehelix')
plt.xticks(rotation = 45, ha = 'right', fontsize=15)
plt.yticks(fontsize=15)

plt.xlabel('Year Incarcerated', fontsize=19, fontweight='bold')
plt.ylabel('Inmate Count', fontsize=19, fontweight='bold')
plt.title('Actives Incarceration Numbers', fontweight='bold', fontsize=22)
```

```
[77]: Text(0.5, 1.0, 'Actives Incarceration Numbers')
```



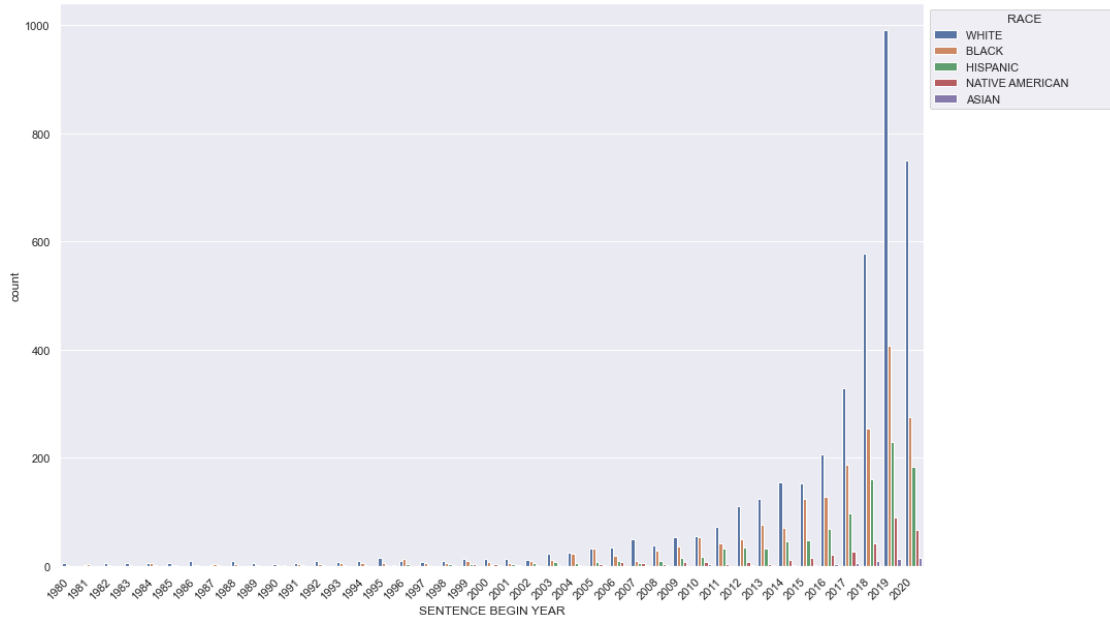


The percentage of black inmates steadily rises over the Actives. Hispanic inmate numbers stay about the same.

```
[78]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="SENTENCE BEGIN YEAR", hue="RACE DESC", ax=ax
                  , data=dfActive,
                  hue_order=dfActive["RACE DESC"].value_counts().iloc[:5].index)
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))
```

```
[78]: <matplotlib.legend.Legend at 0x7f85601f5280>
```

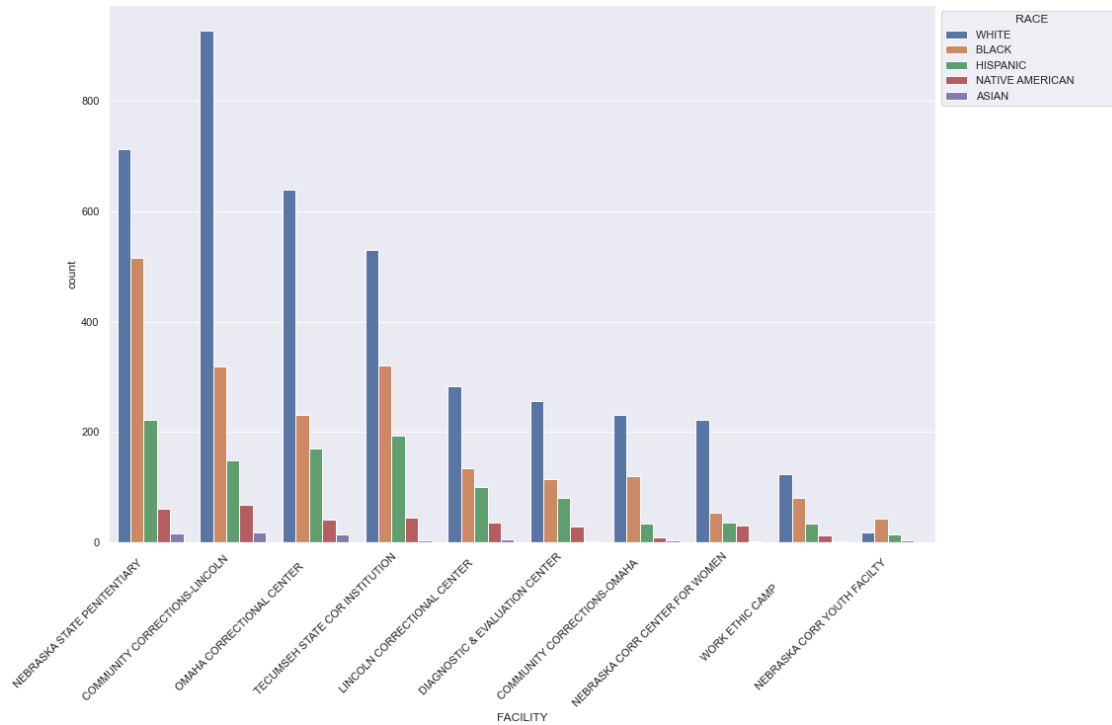


Since most of the active prisoners are labelled with what facility they are in, another graph was made to show the inmate count of facilities by race.

```
[79]: fig_dims = (15, 10)
fig, ax = plt.subplots(figsize=fig_dims)
sns.set_theme(style="darkgrid")

ax = sns.countplot(x="FACILITY", hue="RACE_DESC", ax=ax
                  , data=dfActive,
                  order=dfActive["FACILITY"].value_counts().iloc[:15].index,
                  hue_order=dfActive["RACE_DESC"].value_counts().iloc[:5].index)
plt.xticks(rotation = 45, ha = 'right')
plt.legend(title = 'RACE', bbox_to_anchor = (1, 1))
```

```
[79]: <matplotlib.legend.Legend at 0x7f8561968c10>
```



## 0.14 13.0 All Values Compared

### 0.14.1 13.1 Average Sentence Begin Age

```
[80]: print('1980s:')
      print(df1980['SENTENCE BEGIN AGE'].mean())

      print('1990s:')
      print(df1990['SENTENCE BEGIN AGE'].mean())

      print('2000s:')
      print(df2000['SENTENCE BEGIN AGE'].mean())

      print('2010s:')
      print(df2010['SENTENCE BEGIN AGE'].mean())

      print('Active:')
      print(dfActive['SENTENCE BEGIN AGE'].mean())
```

```
1980s:
28.545750329650527
1990s:
30.15704491300193
2000s:
```

32.253207294264755  
2010s:  
33.66685052682106  
Active:  
33.84659679859728

#### 0.14.2 13.2 Average Min/Max Term or Year by Decade

```
[81]: print('1980s:')  
      print(df1980['MIN TERM/YEAR'].mean())  
      print(df1980['MAX TERM/YEAR'].mean())  
  
      print('1990s:')  
      print(df1990['MIN TERM/YEAR'].mean())  
      print(df1990['MAX TERM/YEAR'].mean())  
  
      print('2000s:')  
      print(df2000['MIN TERM/YEAR'].mean())  
      print(df2000['MAX TERM/YEAR'].mean())  
  
      print('2010s:')  
      print(df2010['MIN TERM/YEAR'].mean())  
      print(df2010['MAX TERM/YEAR'].mean())  
  
      print('Active:')  
      print(dfActive['MIN TERM/YEAR'].mean())  
      print(dfActive['MAX TERM/YEAR'].mean())
```

1980s:  
2.5947281713344315  
5.680630736643916  
1990s:  
2.926451886392539  
5.917337855023315  
2000s:  
3.3803229110775086  
5.8596653821764795  
2010s:  
4.360678744577698  
6.705962405375521  
Active:  
13.743046178773968  
19.598568728058332

### 0.14.3 13.3 Racial Breakdown of Inmates by Decade

```
[82]: print('1980s:')
      print(df1980.value_counts('RACE DESC', normalize=True))
      print()
      print('1990s:')
      print(df1990.value_counts('RACE DESC', normalize=True))
      print()
      print('2000s:')
      print(df2000.value_counts('RACE DESC', normalize=True))
      print()
      print('2010s:')
      print(df2010.value_counts('RACE DESC', normalize=True))
      print()
      print('Active:')
      print(dfActive.value_counts('RACE DESC', normalize=True))
```

```
1980s:
RACE DESC
WHITE                0.659802
BLACK                0.251236
NATIVE AMERICAN     0.045187
HISPANIC             0.038127
OTHER                0.005413
ASIAN                0.000118
                   0.000118
```

dtype: float64

```
1990s:
RACE DESC
WHITE                0.577646
BLACK                0.270242
HISPANIC             0.102515
NATIVE AMERICAN     0.045217
ASIAN                0.003603
OTHER                0.000777
```

dtype: float64

```
2000s:
RACE DESC
WHITE                0.582801
BLACK                0.227355
HISPANIC             0.131555
NATIVE AMERICAN     0.049071
ASIAN                0.006780
OTHER                0.002195
                   0.000146
```

```
PACIFIC ISLANDER    0.000098
```

```
dtype: float64

2010s:
RACE DESC
WHITE                0.569746
BLACK                0.239942
HISPANIC             0.128859
NATIVE AMERICAN     0.043166
OTHER                0.008591
ASIAN                0.006975
                    0.001956
PACIFIC ISLANDER    0.000723
HAWAIIAN ISLANDER   0.000043
dtype: float64

Active:
RACE DESC
WHITE                0.534837
BLACK                0.261140
HISPANIC             0.140697
NATIVE AMERICAN     0.045234
ASIAN                0.008777
OTHER                0.008237
PACIFIC ISLANDER    0.000540
                    0.000540
dtype: float64
```

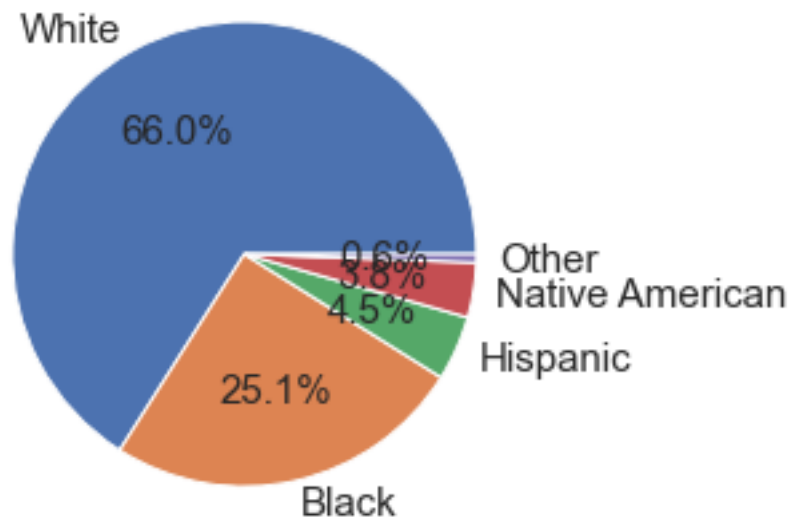
## 0.15 14. Extra Figures for Race Breakdown in each decade

```
[83]: import matplotlib.pyplot as plt
import numpy as np

y = np.array([65.98, 25.12, 4.51, 3.81, .58])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Inmates Incarcerated within the 1980s', bbox={'facecolor':'0.8',
↳ 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
↳ textprops={'fontsize': 15})
plt.show()
```

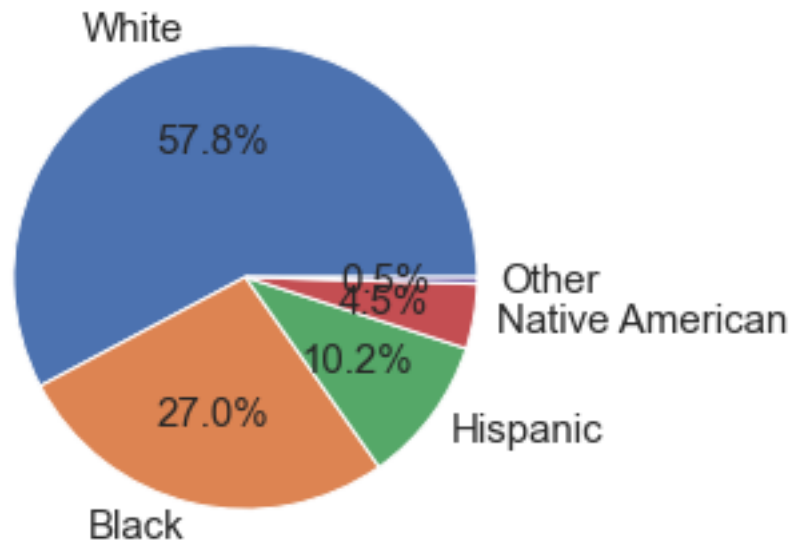
Inmates Incarcerated within the 1980s



```
[84]: y = np.array([57.76, 27.02, 10.25, 4.5, .47])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Inmates Incarcerated within the 1990s', bbox={'facecolor':'0.8',
↳ 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
↳ textprops={'fontsize': 15})
plt.show()
```

Inmates Incarcerated within the 1990s

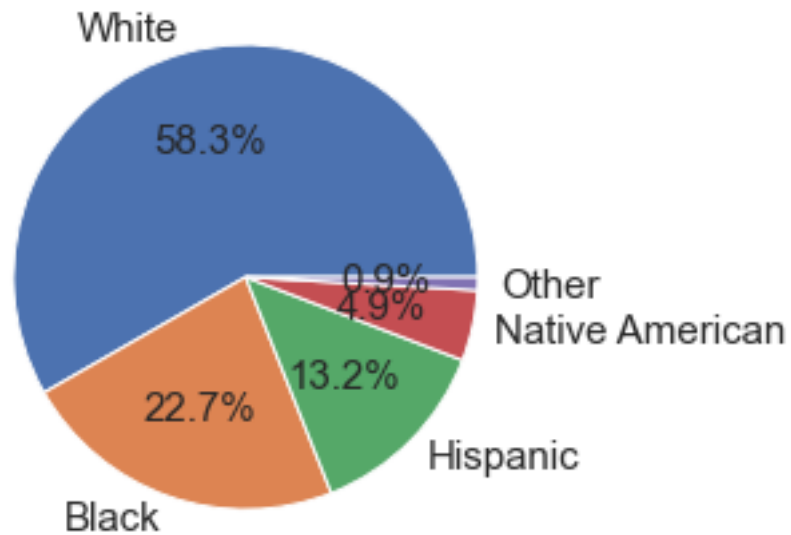


```
[85]: y = np.array([58.28, 22.74, 13.16, 4.91, .91])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Inmates Incarcerated within the 2000s', bbox={'facecolor':'0.8',
↳ 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
↳ textprops={'fontsize': 15})
plt.show()
```



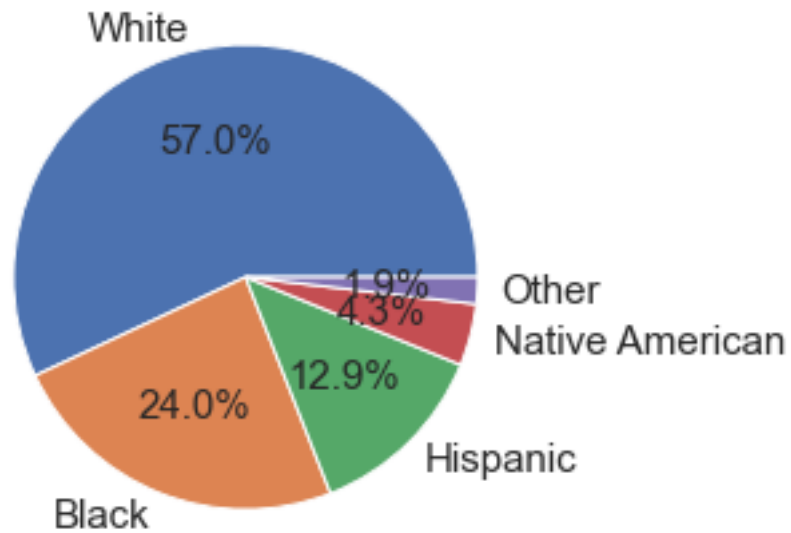
Inmates Incarcerated within the 2000s



```
[86]: y = np.array([56.97, 23.99, 12.88, 4.31, 1.85])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Inmates Incarcerated within the 2010s', bbox={'facecolor':'0.8',
↳ 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
↳ textprops={'fontsize': 15})
plt.show()
```

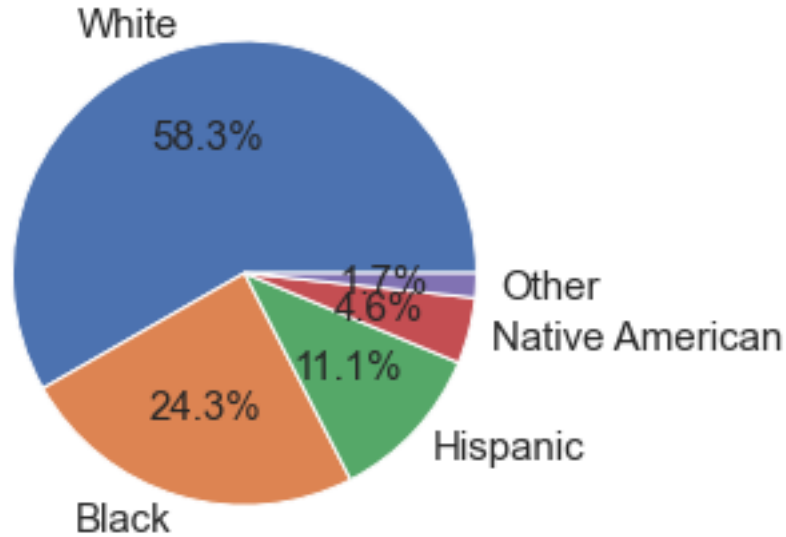
Inmates Incarcerated within the 2010s



```
[87]: y = np.array([58.26, 24.31, 11.13, 4.60, 1.7])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Inmates Incarcerated between 1980 and 2020', bbox={'facecolor':'0.
↪8', 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
↪textprops={'fontsize': 15})
plt.show()
```

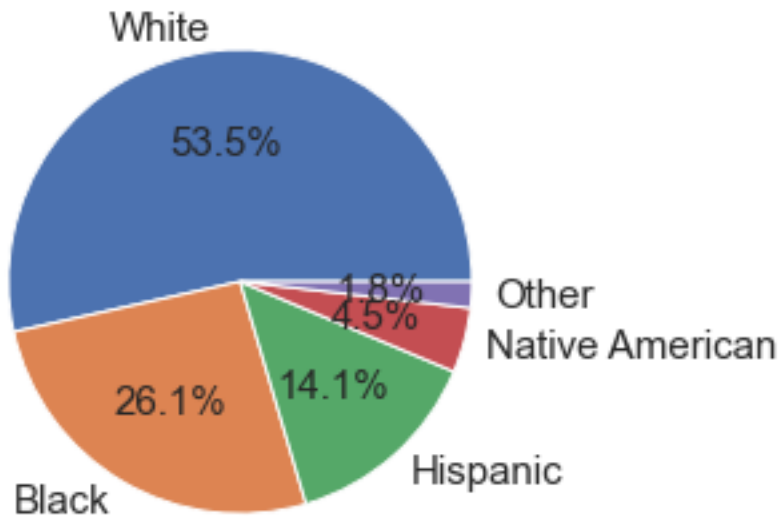
Inmates Incarcerated between 1980 and 2020



```
[88]: y = np.array([53.48, 26.11, 14.07, 4.52, 1.82,])
mylabels = ["White", "Black", "Hispanic", "Native American", "Other"]

plt.title('Active Inmates', bbox={'facecolor':'0.8', 'pad':5})
plt.pie(y, labels = mylabels, autopct='%1.1f%%', radius=1.0,
        textprops={'fontsize': 15})
plt.show()
```

Active Inmates



[ ]: