BANA.780 Midterm

By Dana DiVincenzo

Introduction

In this assignment, I was given data from a Rochester NY based company which described if their staff members were vaccinated, and gave other information about the employees. I was tasked with exploring and preparing the data to possibly be used in a predictive model later on. The goal of the predictive model would be to predict which employees are likely to get vaccinated, and to identify which features are most important in this prediction.

Step 1: Inspecting the data

In this section, an initial inspection was done on the dataset. I looked at each of the columns individually to see if they were necessary, or if they needed cleaning or manipulation for easier understanding and visualization. Further inspection than what is shown below may have been done, but redundant inspections will be removed/ collapsed.

In [1]:
```
import matplotlib.pyplot as plt
import matplotlib.ticker as tkr
import numpy as np
import pandas as pd
import seaborn as sns
from scipy.stats import ttest_ind
```

In [2]:
```
# read in files
file = '/Users/dana/Downloads/VaccinationAnalysis.csv'
vacdf = pd.read_csv(file)
```

In [3]:
```
# 622 rows
vacdf.head()
```

Out[3]:

	Marital Status	Ethnicity	Gender	Age	Shift	Years of Service	Zip Code	Skill	Vaccinated	Salaried / Hourly	Age Range
0	Single	Two or more races	Male	28	1st Shift (7a-3p)	5	13039	A	Yes	Hourly	25-34
1	Single	White	Male	56	1st Shift (7a-3p)	8	14001	A	Yes	Salaried	55-64
2	Married	White	Male	52	1st Shift (7a-3p)	8	14005	B	Yes	Hourly	45-54

	Marital Status	Ethnicity	Gender	Age	Shift	Years of Service	Zip Code	Skill	Vaccinated	Salaried / Hourly	Age Range
3	Married	White	Male	51	1st Shift (7a-3p)	10	14005	B	No	Hourly	45-54
4	Married	White	Male	48	1st Shift (7a-3p)	21	14020	A	Yes	Salaried	45-54

In [4]:
```
print(vacdf["Ethnicity"].unique())
# I will remove the "Not specified" group

print(vacdf["Marital Status"].unique())
# I will remove the "Unknown" group

print(vacdf["Gender"].unique())

print(vacdf["Age"].unique())

print(vacdf["Shift"].unique())
# I will remove the "Other" group

print(vacdf["Years of Service"].unique())

print(vacdf["Zip Code"].unique())
# I will change 145341450 to 14534
# there is only 1 count of 145341450 and 12 counts of 14534, and all other zip c
# im going to remove the 13039 because that is a completely different area and t
```

```
['Two or more races' 'White' 'Hispanic or Latino'
 'Black or African American' 'Native Hawaiian or Other Pacific Islander'
 'Not specified' 'Asian' 'American Indian/Alaskan Native']
['Single' 'Married' 'Unknown']
['Male' 'Female']
[28 56 52 51 48 42 62 24 60 43 50 32 46 54 30 45 44 49 73 29 27 55 35 65
 39 61 64 25 41 22 36 40 21 70 57 31 59 17 58 34 74 16 71 33 47 53 38 37
 66 19 20 26 63 67 68 69 23 72 75 76 80]
['1st Shift (7a-3p)' '3rd Shift (11p-7a)' '2nd Shift (3p-11p)' 'Other']
[ 5  8 10 21 20  1  2  7  3  6 16 28 14 18 15  4 40 29  0 19 17 32 27  9
 11 12 13 39 22 31 24 30 25 23 41 43 42 36 34 26 33]
[    13039     14001     14005     14020     14054     14414     14416
     14420     14423     14424     14425     14428     14435     14437
     14445     14450     14454     14464     14467     14468     14469
     14470     14472     14480     14481     14482     14485     14487
     14502     14505     14510     14511     14514     14517     14519
     14522     14525     14526     14532     14534     14543     14546
     14548     14551     14555     14559     14560     14564     14568
     14572     14580     14586     14589     14605     14606     14607
     14608     14609     14610     14611     14612     14613     14615
     14616     14617     14618     14619     14620     14621     14622
     14623     14624     14625     14626     14692     14822     14836
 145341450]
```

After looking at the data more closely, I was able to decide which rows should be eliminated. Rows that containe information that is not helpful, such as "Marital Status" = "Other," should be removed. A few values in the zip code factor should be edited to maintain consistency throughout the column. One other thing that stood out to me is that there are two columns which tell us about the employee's age: "Age" and "Age Range." We do not need two columns that tell us the same thing, and they would be too highly correlated anyways, so one of them can be removed. I think "Age" should be removed, as vaccinations are being rolled out to certain age groups at once, and that factor may be more meaningful for predicting vaccinations.

Step 2: Data Preparation

In this section, I prepare the data for visualizations by implementing the changes discussed in the previous section.

In [5]:
```
# dropping rows with bad data
vacdf = vacdf[vacdf["Ethnicity"] != "Not specified"]
vacdf = vacdf[vacdf["Marital Status"] != "Unknown"]
vacdf = vacdf[vacdf["Shift"] != "Other"]
vacdf = vacdf[vacdf["Zip Code"] != 13039]
vacdf = vacdf.drop(columns=['Age'])
```

In [6]:
```
# replacing the badly formatted zip code with one in the correct format
vacdf['Zip Code'].replace(145341450, 14534, inplace=True)
```

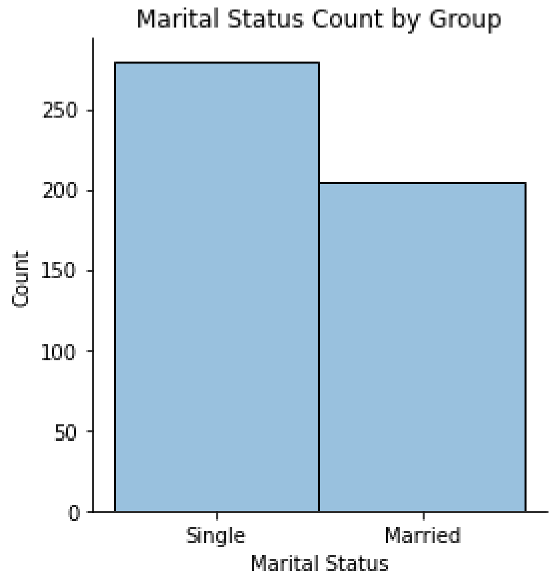
Step 3: Inspecting Feature Variability

In this section, I inspect each of the variables in the dataset to see which have low variability. Variables which are identified as having too low of variability should be dropped as they will not meaningfully contribute to the model’s ability to predict vaccinations.

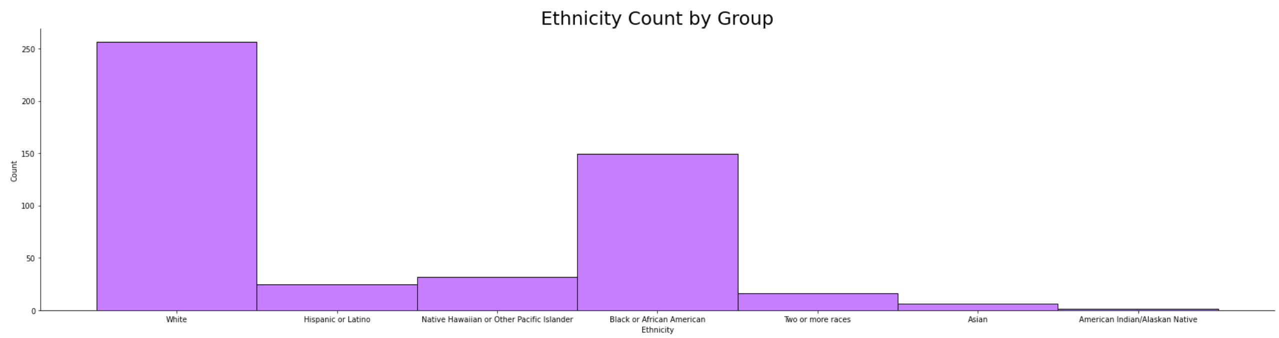
In [7]:
```
vacdf.var()
# ignore zip code
```

Out[7]:
```
Years of Service      74.919784
Zip Code            8052.432751
dtype: float64
```

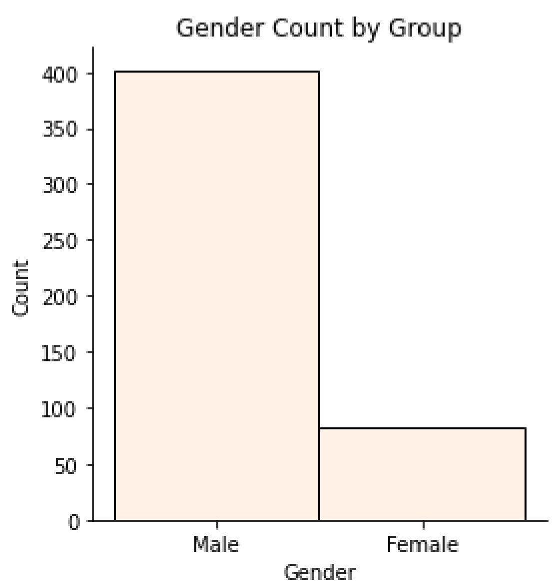
In [8]:
```
maritalhist = sns.displot(vacdf["Marital Status"], color = "#99c1de", height = 4
plt.title("Marital Status Count by Group", fontsize = 12);
```



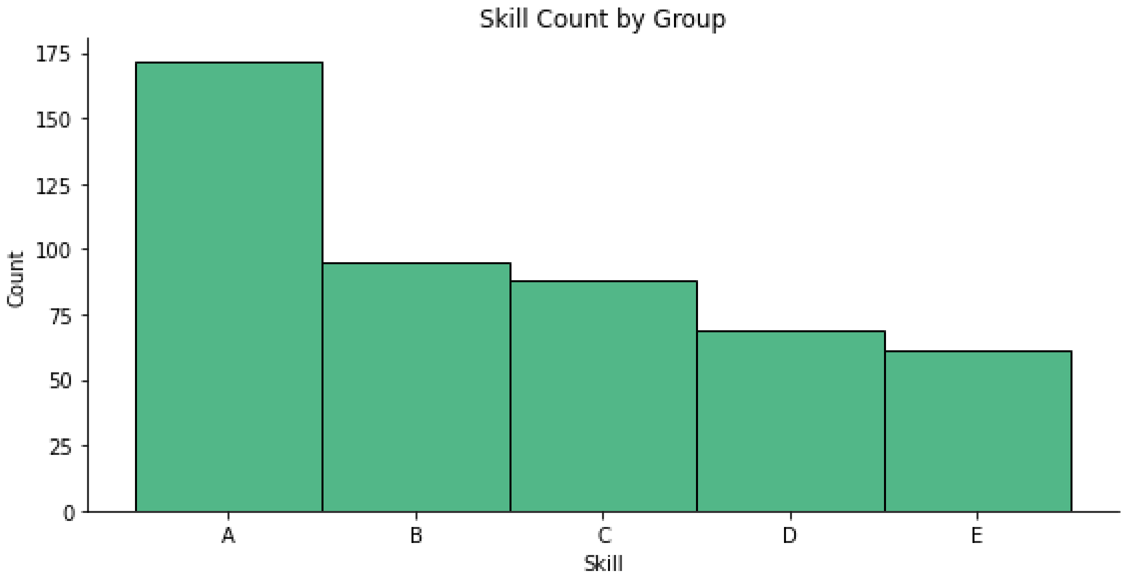
In [9]:
```
ethhist = sns.displot(vacdf["Ethnicity"], color = "#c77dff", height = 6, aspect
plt.title("Ethnicity Count by Group", fontsize = 25);
# this one should be dropped
```



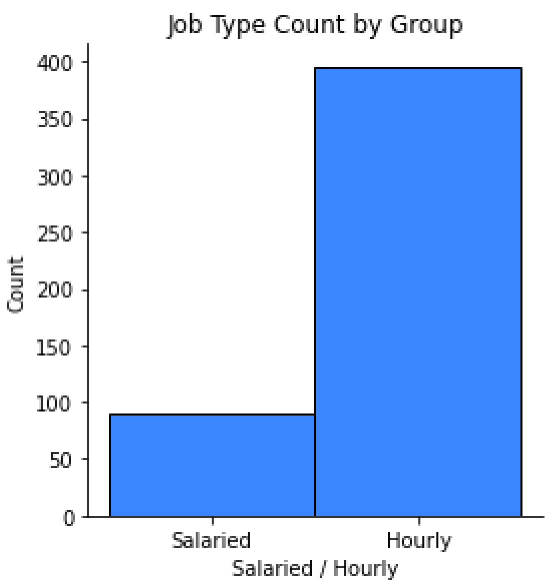
In [10]:
```
genhist = sns.displot(vacdf["Gender"], color = "#fff1e6", height = 4, aspect = 1
plt.title("Gender Count by Group", fontsize = 12);
# this one should be dropped
```



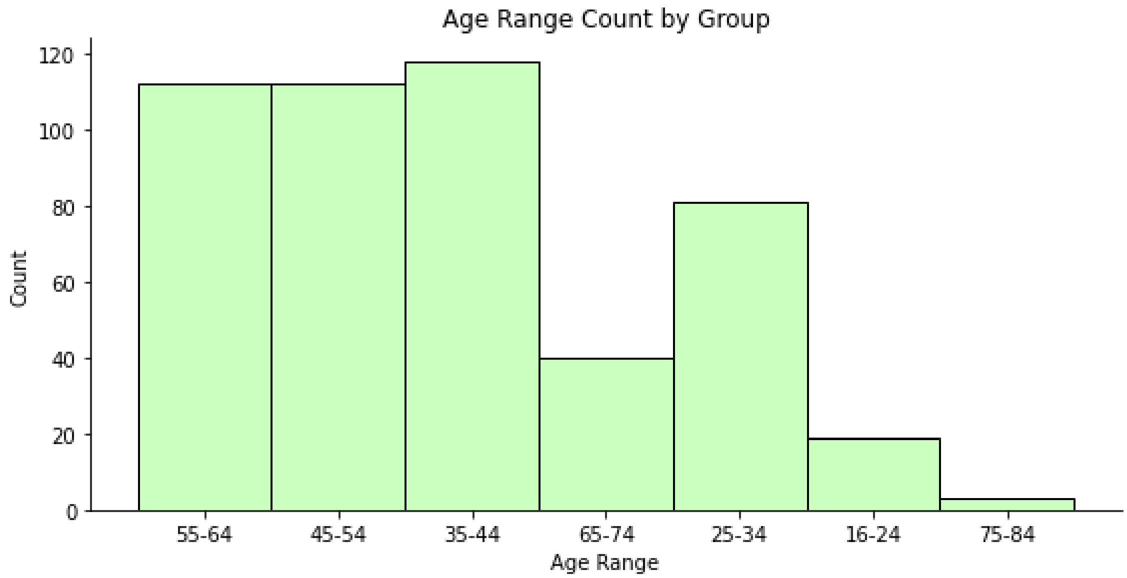
In [42]:
```
skillhist = sns.displot(vacdf["Skill"], color = "#52b788", height = 4, aspect =
plt.title("Skill Count by Group", fontsize = 12);
```



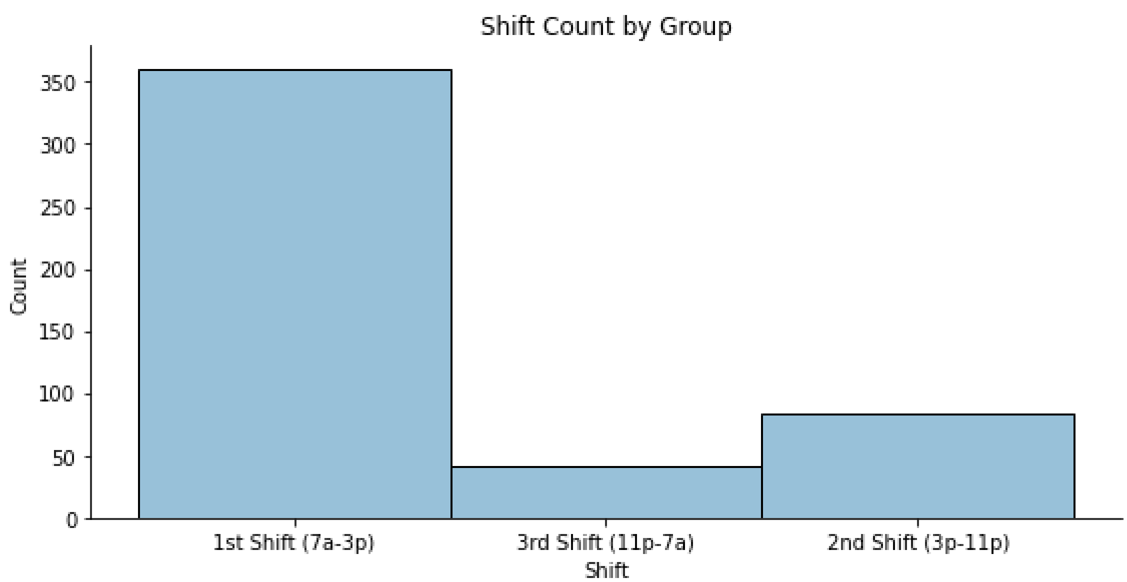
In [12]:
```
salhist = sns.displot(vacdf["Salaried / Hourly"], color = "#3a86ff", height = 4,
plt.title("Job Type Count by Group", fontsize = 12);
# this one should be dropped
```



In [13]:
```
agehist = sns.displot(vacdf["Age Range"], color = "#caffbf", height = 4, aspect
plt.title("Age Range Count by Group", fontsize = 12);
```



In [14]:
```
salhist = sns.displot(vacdf["Shift"], color = "#98c1d9", height = 4, aspect = 2,
plt.title("Shift Count by Group", fontsize = 12);
# drop
```



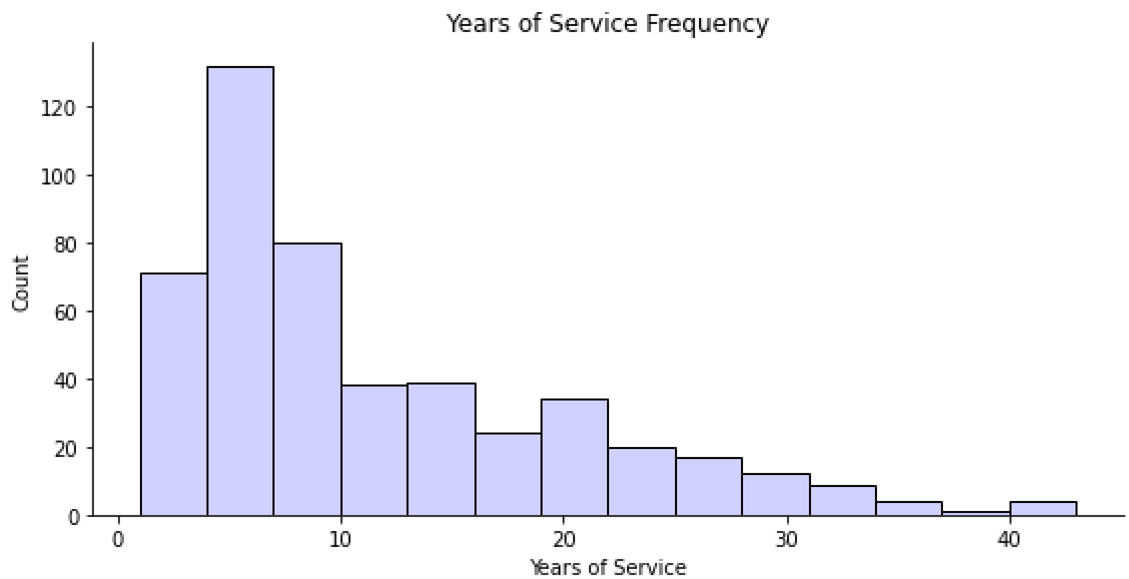
In [15]:
```
# dropping categorical variables with low variability
vacdf = vacdf.drop(columns=['Ethnicity', 'Gender', 'Salaried / Hourly', "Shift"]
```

I calculated the variance of each of the numerical factors. I then created frequency plots to visualize the count of datapoints in each level of every categorical variable. The goal of these plots was to quicky see which variables had datapoints that were not well distributed across the levels, and therefore had low variabilty. There was no set threshold (such as 90%) which determined which factors had "too low" of variability, rather, those that seemed very poorly distributed to the naked eye were dropped. Based on this examination, I decided to remove "Ethnicity," "Gender," "Salaried / Hourly," and "Shift" from the dataset.

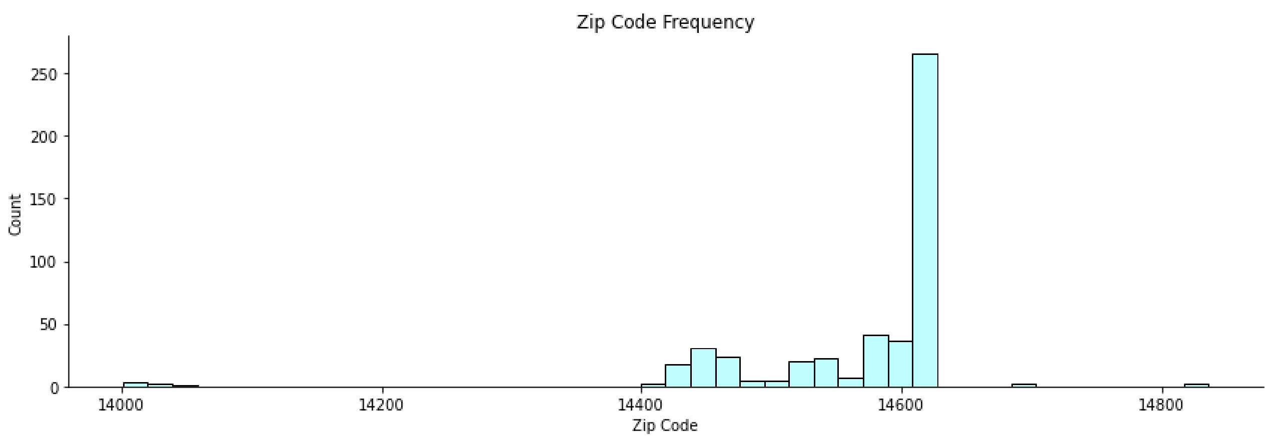
Step 4: Showing the Frequency of the Remaining Variables

In the previous step, I created histograms to visualize the frequency of the categorical independant variables in the dataset. Here, I create histograms for the numerical variables as well as our dependant variable.

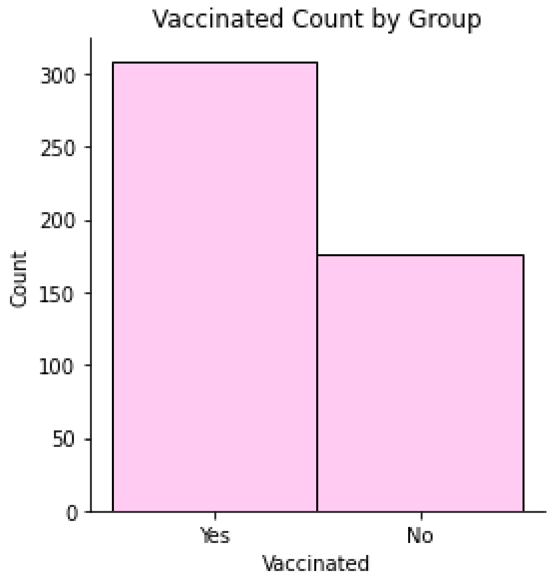
In [16]:
```
yoshist = sns.displot(vacdf["Years of Service"], color = "#d0d1ff", height = 4,
plt.title("Years of Service Frequency", fontsize = 12);
```



In [17]:
```
# lets do a different visualization for this one...
ziphist = sns.displot(vacdf["Zip Code"], color = "#c0fdff", height = 4, aspect =
plt.title("Zip Code Frequency", fontsize = 12);
```



In [18]:
```
agehist = sns.displot(vacdf["Vaccinated"], color = "#ffcbf2", height = 4, aspect
plt.title("Vaccinated Count by Group", fontsize = 12);
```

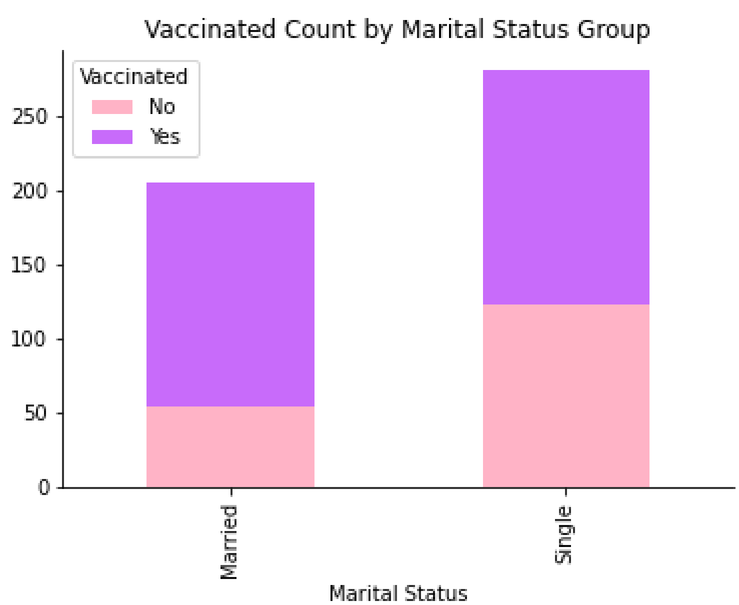


Based on these histograms, we can see that "Years of Service" peaks around 5 years, and then quickly decreases. We can also see that most of the zip codes fall between 14400 and 14630, with the vast majority falling in the 14620's. The visualization of the dependant variables shows us that more employees in the dataset have been vaccinated than have not been vaccinated.

Step 5: Visualizing Factors with the Dependant Variable

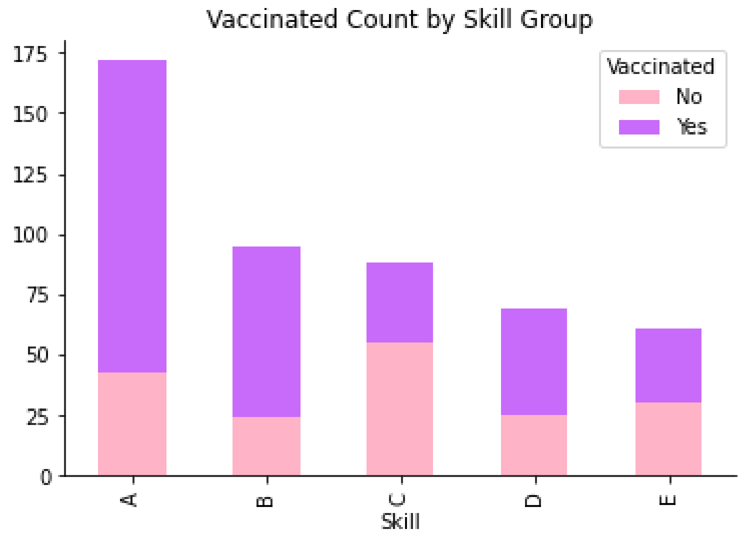
In this section, I plot each factor agains the dependant variable. Categorical variables are shown with stacked bar chart, and numerical variables are shown with a violin plot. These plots allow us to see how the different features may be influencing vaccinations. Individual analysis will be written for each graph.

In [34]:
```
g = pd.crosstab(vacdf['Marital Status'], vacdf['Vaccinated']).plot(kind='bar', s
plt.title("Vaccinated Count by Marital Status Group", fontsize = 12)
g.spines['top'].set_visible(False)
g.spines['right'].set_visible(False);
```



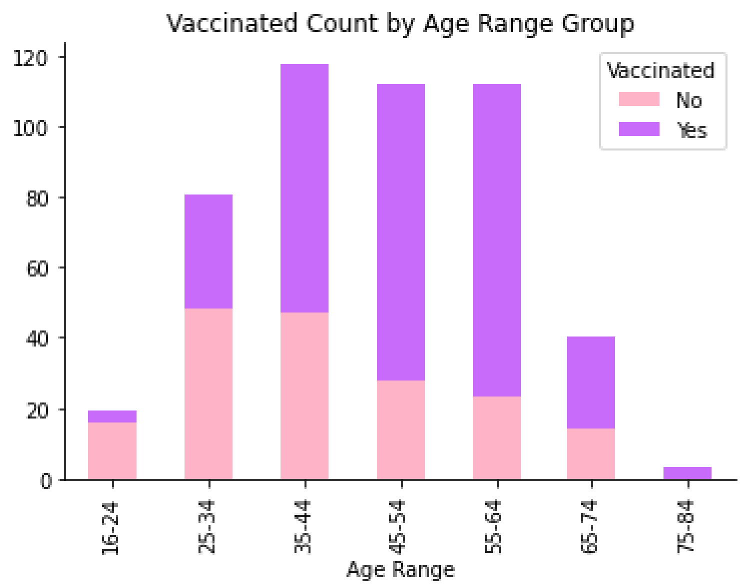
This graph shows us that there are more single employees in the dataset then there are married employees. However, a higher proportion of married employees are vaccinated compared to single employees.

In [35]:
```
h = pd.crosstab(vacdf['Skill'], vacdf['Vaccinated']).plot(kind='bar', stacked=Tr
plt.title("Vaccinated Count by Skill Group", fontsize = 12)
h.spines['top'].set_visible(False)
h.spines['right'].set_visible(False);
```



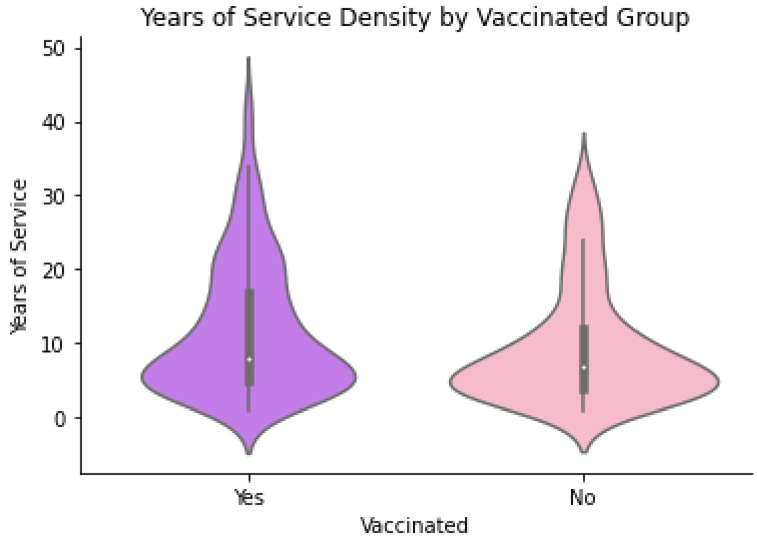
This graph shows us that skill group A is the most frequent within the dataset. Skill group A, B, and D all have similar proportions of vaccinated to unvaccinated, with a higher propotion than the other groups. Group E follows, and group C has the lowest proportion.

In [39]:
```
b = pd.crosstab(vacdf['Age Range'], vacdf['Vaccinated']).plot(kind='bar', stacke
plt.title("Vaccinated Count by Age Range Group", fontsize = 12)
b.spines['top'].set_visible(False)
b.spines['right'].set_visible(False);
```



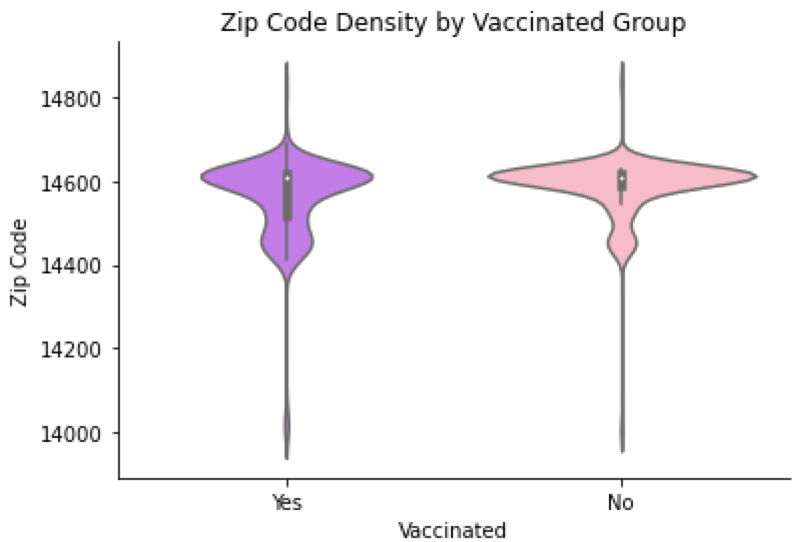
This chart shows us that as the age range increases, the proportion of vaccinated to unvaccinated also increases. This makes sense, as vaccinations have been rolling out to older age ranges first. Although the 75-84 age group has only a few datapoints in the set, everyone in the group is vaccinated.

In [40]:
```
p = sns.violinplot(data=vacdf, x='Vaccinated', y='Years of Service', palette=['#
plt.title("Years of Service Density by Vaccinated Group", fontsize = 12)
p.spines['top'].set_visible(False)
p.spines['right'].set_visible(False);
```



There are a few things we can see from this plot. We can see that those who are vaccinated have a slightly higher "Years of Service" median, which may correlate to age, which we have already seen has a positive relationship with being vaccinated. We can also see that the years of service for both groups is highly concentrated around 5 years, but is more concentrated in the unvaccinated group.

In [41]:
```
v = sns.violinplot(data=vacdf, x='Vaccinated', y='Zip Code', palette=['#c86bfa',
plt.title("Zip Code Density by Vaccinated Group", fontsize = 12)
v.spines['top'].set_visible(False)
v.spines['right'].set_visible(False);
```



This plot shows us that the median zip code for the two groups is extremely similar. We can also see that the zip codes for both groups are highly concentrated around 14620, but the unvaccinated groups has the highest density here. The vaccinated group has a slighly higher density for the zip codes lower than 14600.

Conclusions

By using feature selection, I was able to eliminate variables that seemed unhelpful in predicting whether or not an employee would be vaccinated. I was also able to take a closer look at the relationship between each feature and vaccinated to explore the relationship between them. Although the graphs can give us a good sense of how each feature may influence the dependant variable, the next step would be to run a regression model and look at the coefficients and log odds of each feature, to see which variables are significant and how they effect vaccinated. A stepAIC function would also be useful in determining which variables have predictive power here.

Additionally, some variables which were eliminated due to low variability may be powerful in predicting whether or not an employee was vaccinated - but the data collection may have been biased which resulted in the low variabilty. I would recommend taking a wider sample of employees if possible to see if some of the variabilty in these features could be improved.