Challenge URL https://www.freecodecamp.org/learn/data-analysis-with-python/data-analysis-with-python-projects/demographic-data-analyzer

Assignment: Demographic Data Analyzer

In this challenge you must analyze demographic data using Pandas. You are given a dataset of demographic data that was extracted from the 1994 Census database.

You must use Pandas to answer the following questions:

How many people of each race are represented in this dataset? This should be a Pandas series with race names as the index labels. (race column)

What is the average age of men?

What is the percentage of people who have a Bachelor's degree?

What percentage of people with advanced education (Bachelors, Masters, or Doctorate) make more than 50K?

What percentage of people without advanced education make more than 50K?

What is the minimum number of hours a person works per week?

What percentage of the people who work the minimum number of hours per week have a salary of more than 50K?

What country has the highest percentage of people that earn >50K and what is that percentage?

Identify the most popular occupation for those who earn >50K in India.

Use the starter code in the file demographic_data_analyzer. Update the code so all variables set to "None" are set to the appropriate calculation or code. Round all decimals to the nearest tenth.

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy.stats as stats
```

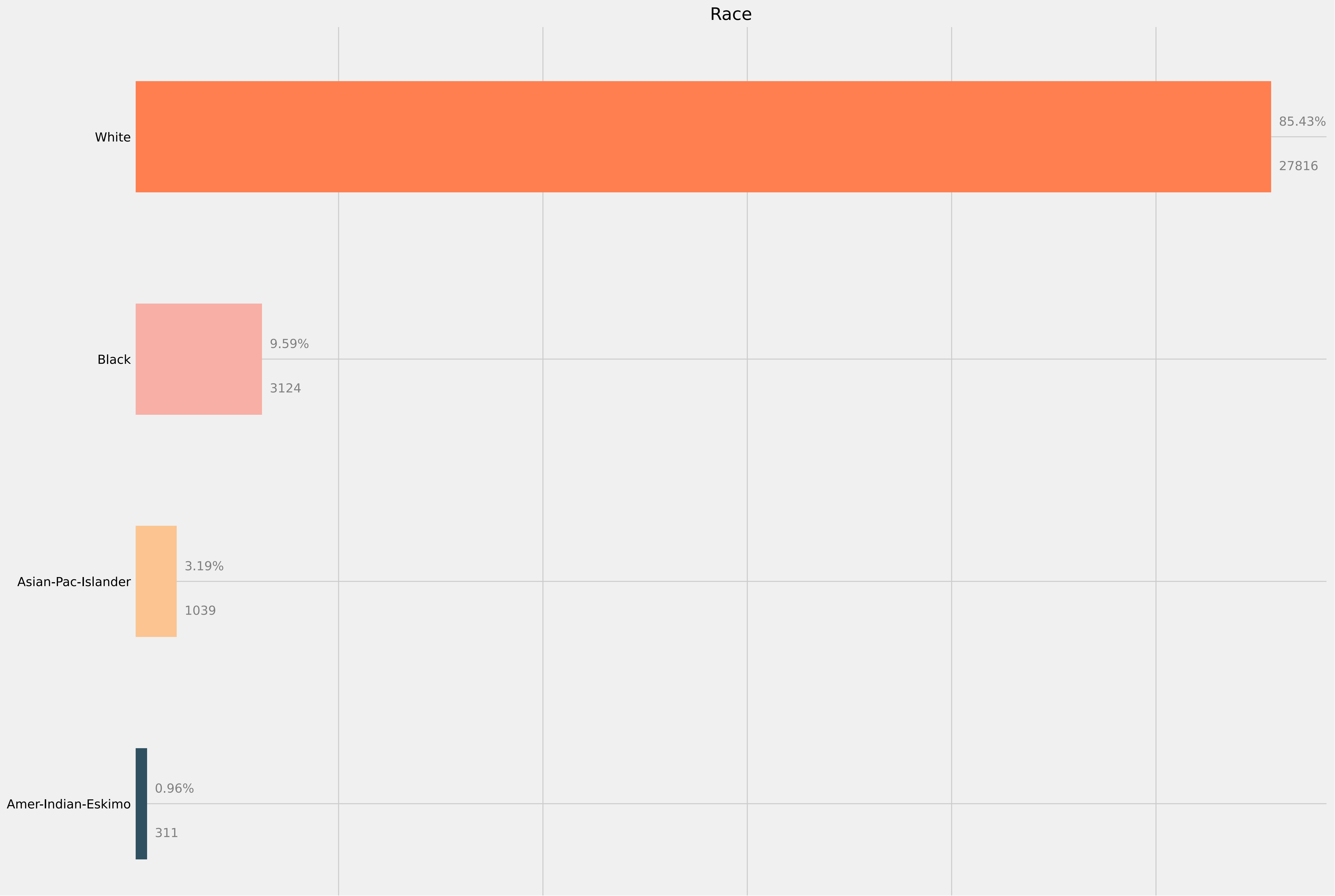
In [ ]:

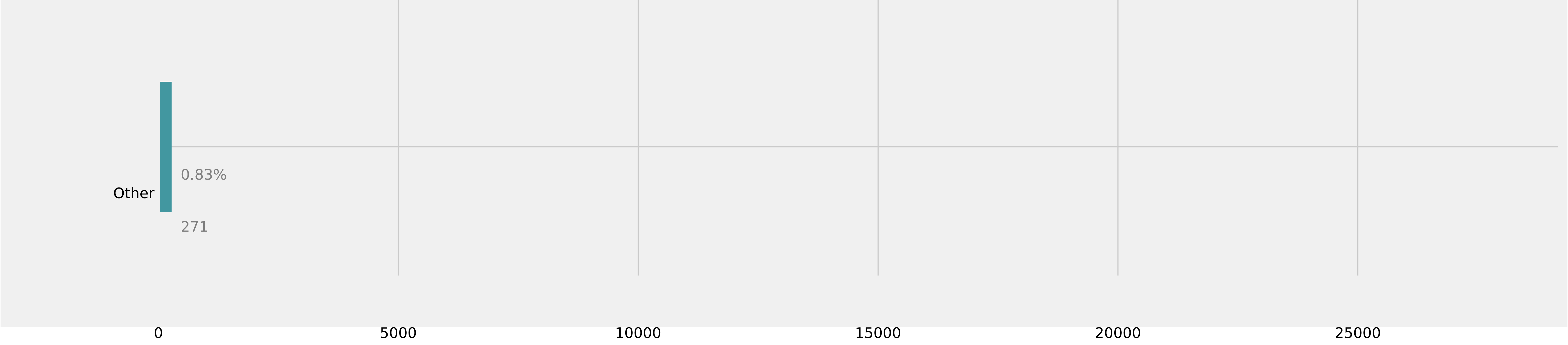
```
dataframe = pd.read_csv('adult.data.csv')

def calculate_demographic_data(print_data=True):
    # Read data from file
    dataframe = pd.read_csv('adult.data.csv')

    # How many of each race are represented in this dataset? This should be a Pandas series with race names as the index labels.
    race_count =  dataframe.race.value_counts()

    # What is the average age of men?
    average_age_men = round(dataframe[dataframe['sex'] == 'Male']['age'].mean(),1)

    # What is the percentage of people who have a Bachelor's degree?
    percentage_bachelors = round(((dataframe.education.values == 'Bachelors').sum()/ dataframe.education.count())*100,1)

    # What percentage of people with advanced education (`Bachelors`, `Masters`, or `Doctorate`) make more than 50K?
    # What percentage of people without advanced education make more than 50K?

    # percentage with salary >50K
    higher_education_rich = round(len(dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K'])/len(dataframe.loc[dataframe['education'].isin(['Bachelors',
    lower_education_rich = round(len(dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K'])/len(dataframe.loc[~dataframe['education'].isin(['Bachelors',


    # What is the minimum number of hours a person works per week (hours-per-week feature)?
    min_work_hours = dataframe['hours-per-week'].min()

    # What percentage of the people who work the minimum number of hours per week have a salary of >50K?
    rich_percentage = round(len(dataframe.loc[dataframe['hours-per-week']<=dataframe['hours-per-week'].min()][dataframe['salary']=='>50K'])/len(dataframe[dataframe['hours-per-week'] == dataframe['hours-per-week'

    # What country has the highest percentage of people that earn >50K?

    # its a bit slow but i could not think of a more efficient solution
    a = dataframe.loc[dataframe['salary']=='>50K']['native-country'].value_counts().keys()
    d = []
    for i in a:
        b = dataframe.loc[dataframe['salary']=='>50K']['native-country'].value_counts()[i]
        c = dataframe['native-country'].value_counts()[i]
        d.append(b/c)
    maximum_value = d[0]
    for i in d:
        if(i>maximum_value):
            maximum_value = i

    highest_earning_country = a[d.index(maximum_value)]
    highest_earning_country_percentage = round(maximum_value*100,1)

    # Identify the most popular occupation for those who earn >50K in India.
    top_IN_occupation = dataframe.loc[dataframe['salary']=='>50K'][dataframe['native-country']=='India']['occupation'].mode()[0]

    # DO NOT MODIFY BELOW THIS LINE

    if print_data:
        print("Number of each race:\n", race_count)
        print("Average age of men:", average_age_men)
        print(f"Percentage with Bachelors degrees: {percentage_bachelors}%")
        print(f"Percentage with higher education that earn >50K: {higher_education_rich}%")
        print(f"Percentage without higher education that earn >50K: {lower_education_rich}%")
        print(f"Min work time: {min_work_hours} hours/week")
        print(f"Percentage of rich among those who work fewest hours: {rich_percentage}%")
        print("Country with highest percentage of rich:", highest_earning_country)
        print(f"Highest percentage of rich people in country: {highest_earning_country_percentage}%")
        print("Top occupations in India:", top_IN_occupation)

    return {
        'race_count': race_count,
        'average_age_men': average_age_men,
        'percentage_bachelors': percentage_bachelors,
        'higher_education_rich': higher_education_rich
        'higher_education_rich': higher_education_rich,
        'lower_education_rich': lower_education_rich,
        'min_work_hours': min_work_hours,
        'rich_percentage': rich_percentage,
        'highest_earning_country': highest_earning_country,
        'highest_earning_country_percentage':
        highest_earning_country_percentage,
        'top_IN_occupation': top_IN_occupation
    }
```

Data Visualization

In [ ]:

```
dataframe = pd.read_csv('adult.data.csv')
race_data_barh_graph = dataframe['race'].value_counts().plot(kind='barh', color=["coral","#F8AFA6","#FBC490","#2F5061","#4297A0"], fontsize=13,figsize=(20,20));
race_data_barh_graph.set_title("Race", fontsize=18)
race_data_barh_graph.set_xlabel("Number of indviduals", fontsize=18);
totals = []
for i in race_data_barh_graph.patches:
    totals.append(i.get_width())
total = sum(totals)

for i in race_data_barh_graph.patches:
    race_data_barh_graph.text(i.get_width()+.40, i.get_y()+.40, \
            "  "+str(round(i.get_width())), fontsize=13,color='grey')
for i in race_data_barh_graph.patches:
    race_data_barh_graph.text(i.get_width()+.20, i.get_y()+.20, \
            "  "+str(round((i.get_width()/total)*100, 2))+'%', fontsize=13,color='grey')
race_data_barh_graph.invert_yaxis()
plt.style.use('fivethirtyeight')
```

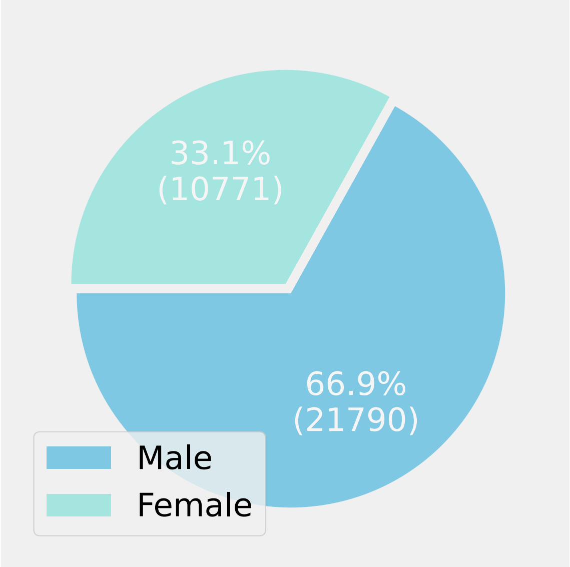

Race
White
85.43%
27816
Black
9.59%
3124
Asian-Pac-Islander
3.19%
1039
Amer-Indian-Eskimo
0.96%
311



In [ ]:

```
def func(pct, allvals):
    absolute = int(round(pct/100.*np.sum(allvals)))
    return "{:.1f}%\n({:d})".format(pct, absolute)
sex_pie_chart = plt.pie(dataframe['sex'].value_counts(),colors=['#7EC8E3','#A4E5E0'], autopct=lambda pct: func(pct, dataframe['sex'].value_counts()), startangle=180,explode=(0.05, 0),textprops=dict(color="whites
plt.legend(sex_pie_chart,loc="lower left",labels=['Male','Female'])
plt.style.use('fivethirtyeight')
```

```
<ipython-input-19-d732d4c8c9e1>:5: UserWarning: You have mixed positional and keyword arguments, some input may be discarded.
  plt.legend(sex_pie_chart,loc="lower left",labels=['Male','Female'])
```



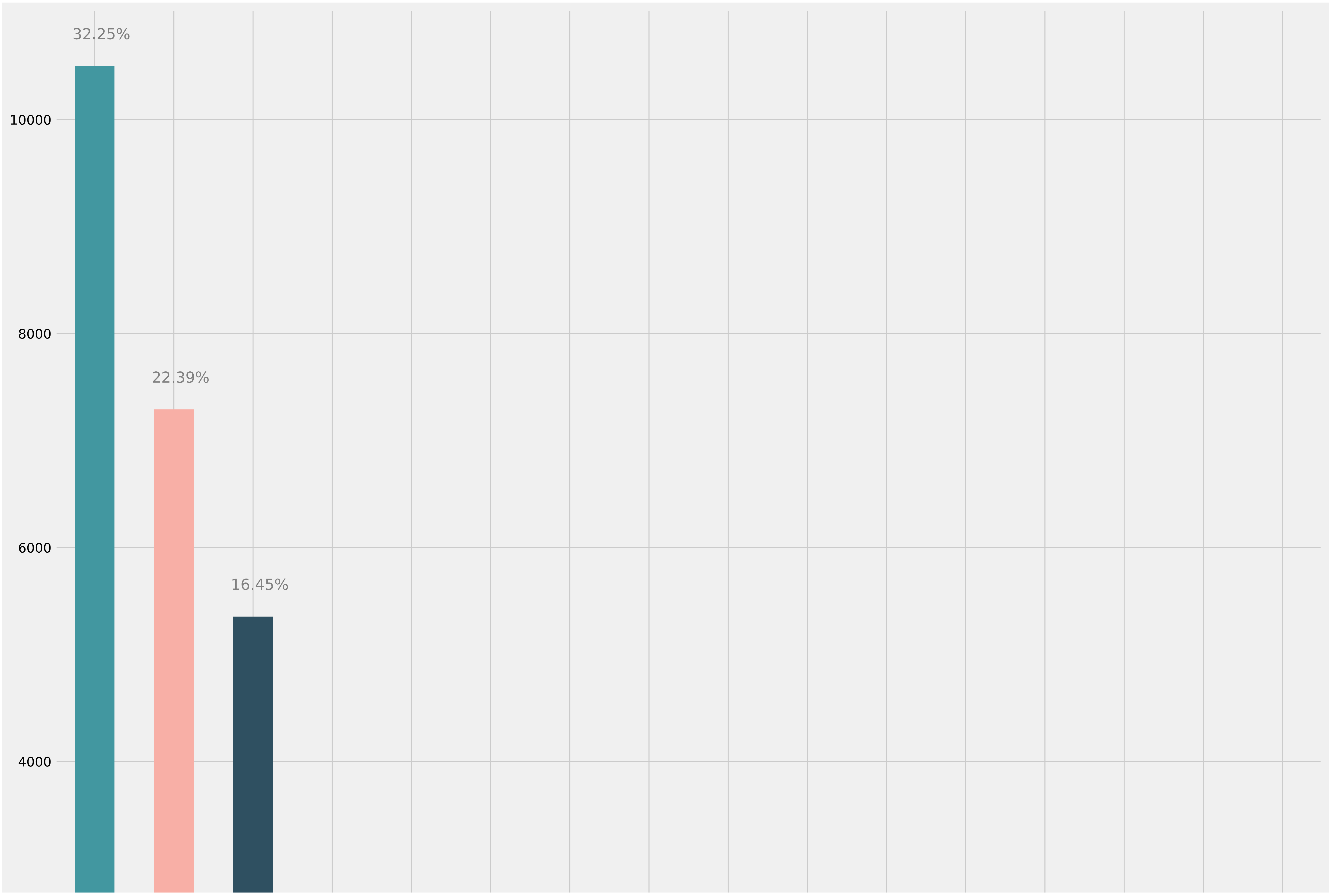
In [ ]:

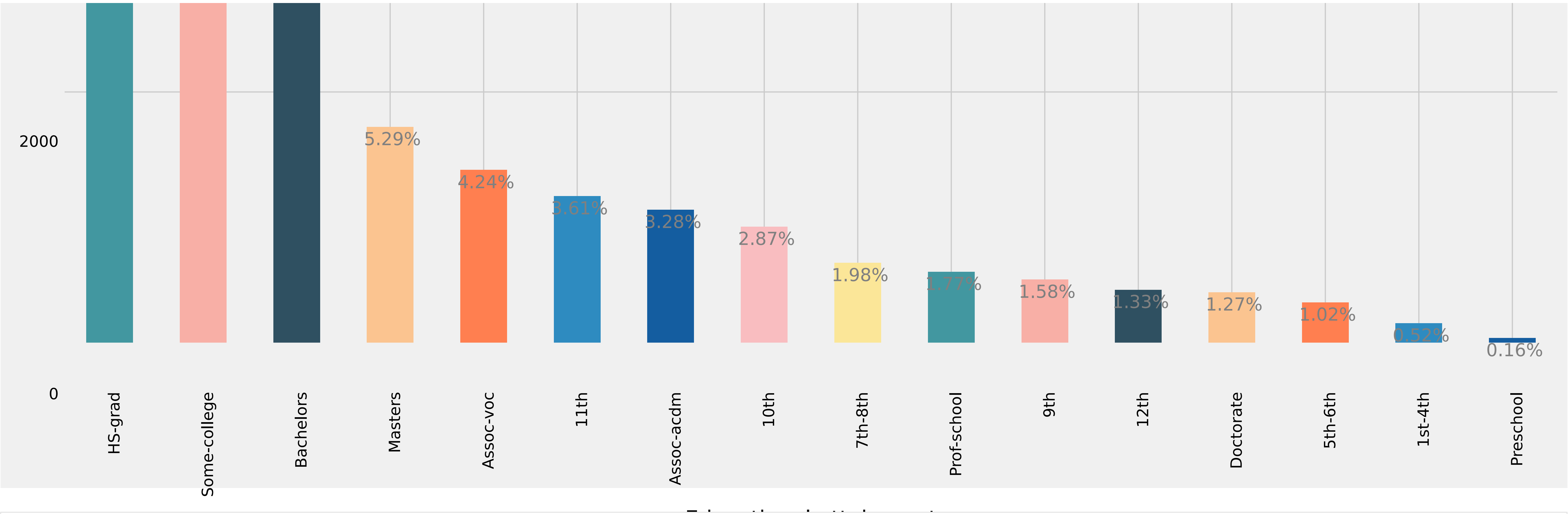
```
male_age_mean = round(dataframe[dataframe['sex'] == 'Male']['age'].mean(),2)
female_age_mean = round(dataframe[dataframe['sex'] == 'Female']['age'].mean(),2)
print("Average male age "+str(male_age_mean))
print("Average female age "+str(female_age_mean))
```

```
Average male age 39.43
Average female age 36.86
```

In [ ]:

```
education_data_bar_graph = dataframe['education'].value_counts().plot(kind='bar', color=["#4297A0","#F8AFA6","#2F5061","#FBC490","coral","#2E8BC0","#145DA0","#F9BDC0","#FBE698"], fontsize=13,figsize=(20,20));
education_data_bar_graph.set_xlabel("Educational attainment", fontsize=18)
totals = []
for i in education_data_bar_graph.patches:
    totals.append(i.get_height())
total = sum(totals)

for i in education_data_bar_graph.patches:
   education_data_bar_graph.text(i.get_x()-.03, i.get_height()+250, \
           str(round((i.get_height()/total)*100, 2))+'%', fontsize=15,
                color='grey')
plt.style.use('fivethirtyeight')
```

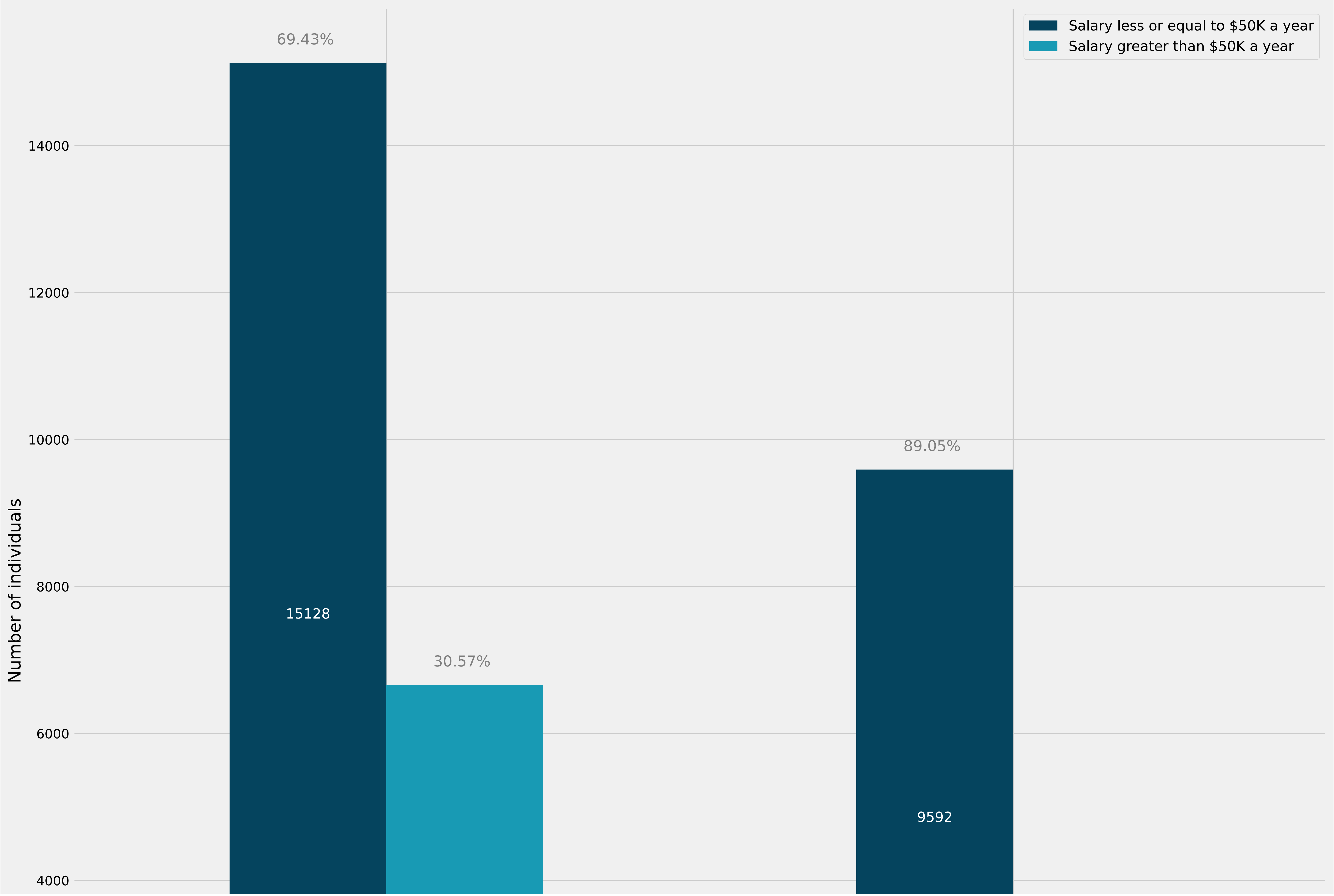

32.25%
22.39%
16.45%
10000
8000
6000
4000

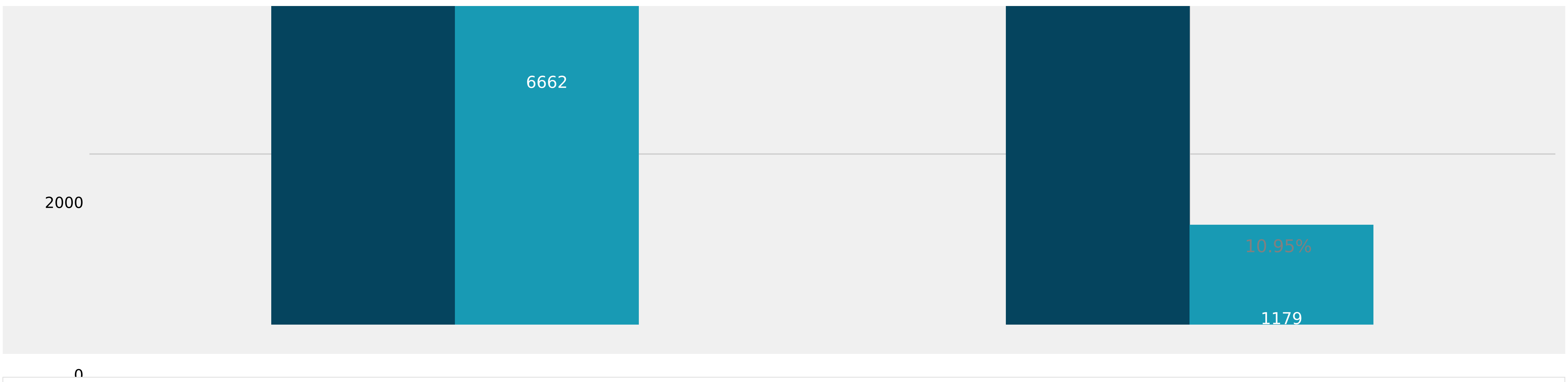


In [ ]:

```
data = pd.DataFrame({'sex': ['Male','Female'],
                     'Salary less or equal to $50K a year': [len(dataframe.loc[dataframe['sex']=='Male'][dataframe['salary']=='<=50K']), len(dataframe.loc[dataframe['sex']=='Female'][dataframe['salary']=='<=50K'
                     'Salary greater than $50K a year': [len(dataframe.loc[dataframe['sex']=='Male'][dataframe['salary']=='>50K']), len(dataframe.loc[dataframe['sex']=='Female'][dataframe['salary']=='>50K'])]
                     })
male_quantity_of_individuals = len(dataframe.loc[dataframe['sex']=='Male'])
female_quantity_of_individuals = len(dataframe.loc[dataframe['sex']=='Female'])

sex_salary_bar_graph = data.plot(kind='bar', figsize=(20,20), width=0.5,fontsize=13,color=['#05445E','#189AB4'])
sex_salary_bar_graph.set_ylabel('Number of individuals')
sex_salary_bar_graph.set_xticklabels(labels=['Male','Female'])
for label in sex_salary_bar_graph.get_xticklabels():
    label.set_ha("right")
    label.set_rotation(0)

sex_salary_bar_graph.text(sex_salary_bar_graph.patches[0].get_x()+.075, sex_salary_bar_graph.patches[0].get_height()+250, \
           str(round(sex_salary_bar_graph.patches[0].get_height()/male_quantity_of_individuals*100, 2))+'%', fontsize=15,
                color='grey')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[2].get_x()+.075, sex_salary_bar_graph.patches[2].get_height()+250, \
           str(round(sex_salary_bar_graph.patches[2].get_height()/male_quantity_of_individuals*100, 2))+'%', fontsize=15,
                color='grey')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[1].get_x()+.075, sex_salary_bar_graph.patches[1].get_height()+250, \
           str(round(sex_salary_bar_graph.patches[1].get_height()/female_quantity_of_individuals*100, 2))+'%', fontsize=15,
                color='grey')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[3].get_x()+.075, sex_salary_bar_graph.patches[3].get_height()+250, \
           str(round(sex_salary_bar_graph.patches[3].get_height()/female_quantity_of_individuals*100, 2))+'%', fontsize=15,
                color='grey')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[0].get_x()/2, sex_salary_bar_graph.patches[0].get_height()//2,sex_salary_bar_graph.patches[0].get_height(),ha = 'center',color = 'white')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[1].get_x()+.125, sex_salary_bar_graph.patches[1].get_height()//2,sex_salary_bar_graph.patches[1].get_height(),ha = 'center',color = 'white')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[2].get_x()+.125, sex_salary_bar_graph.patches[2].get_height()//2,sex_salary_bar_graph.patches[2].get_height(),ha = 'center',color = 'white')
sex_salary_bar_graph.text(sex_salary_bar_graph.patches[3].get_x()+.125, sex_salary_bar_graph.patches[3].get_height()//2,sex_salary_bar_graph.patches[3].get_height(),ha = 'center',color = 'white')
plt.style.use('fivethirtyeight')
```

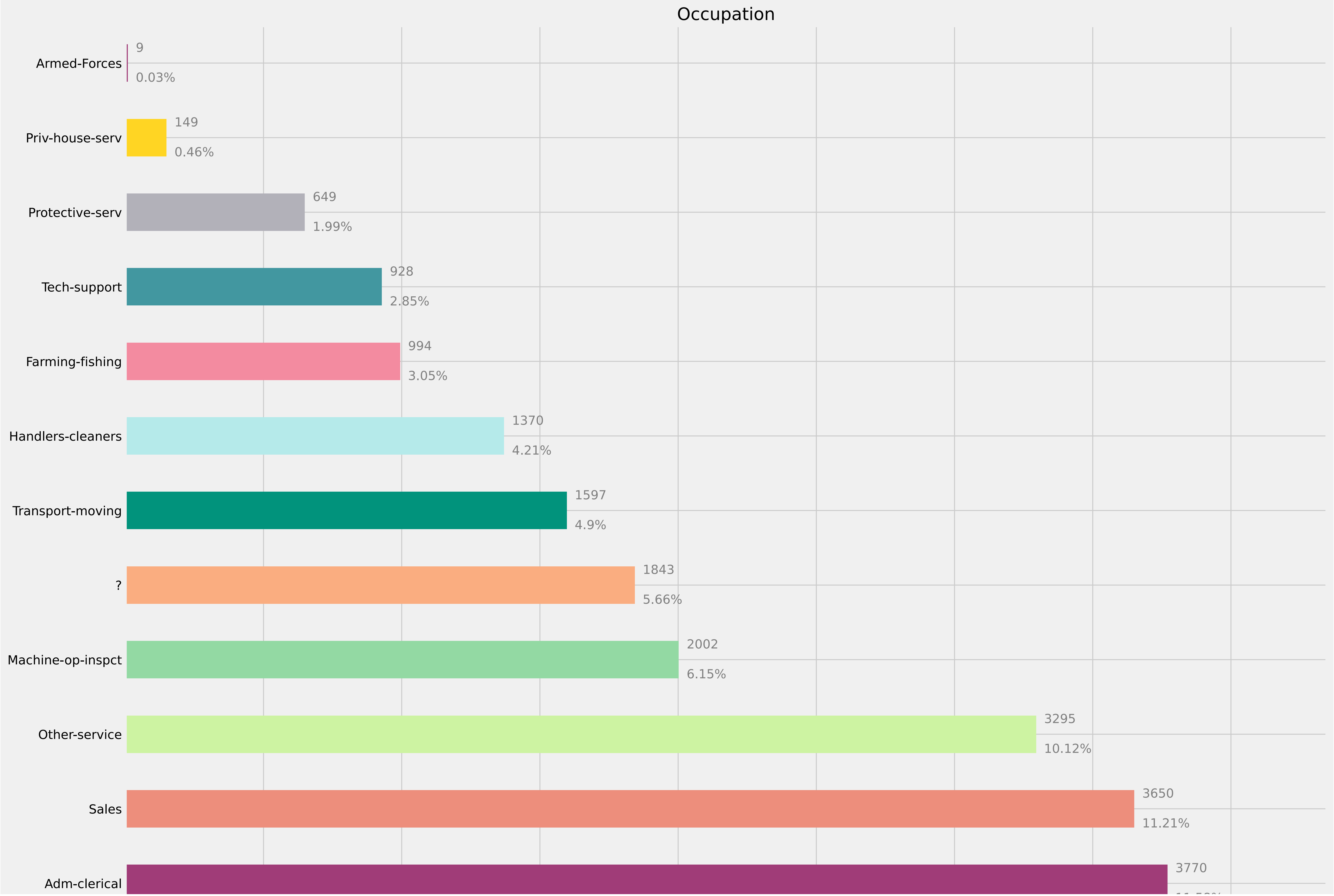
```
<ipython-input-22-a54f6e7c38ca>:2: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  'Salary less or equal to $50K a year': [len(dataframe.loc[dataframe['sex']=='Male'][dataframe['salary']=='<=50K']), len(dataframe.loc[dataframe['sex']=='Female'][dataframe['salary']=='<=50K'])],
<ipython-input-22-a54f6e7c38ca>:3: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  'Salary greater than $50K a year': [len(dataframe.loc[dataframe['sex']=='Male'][dataframe['salary']=='>50K']), len(dataframe.loc[dataframe['sex']=='Female'][dataframe['salary']=='>50K'])]
```

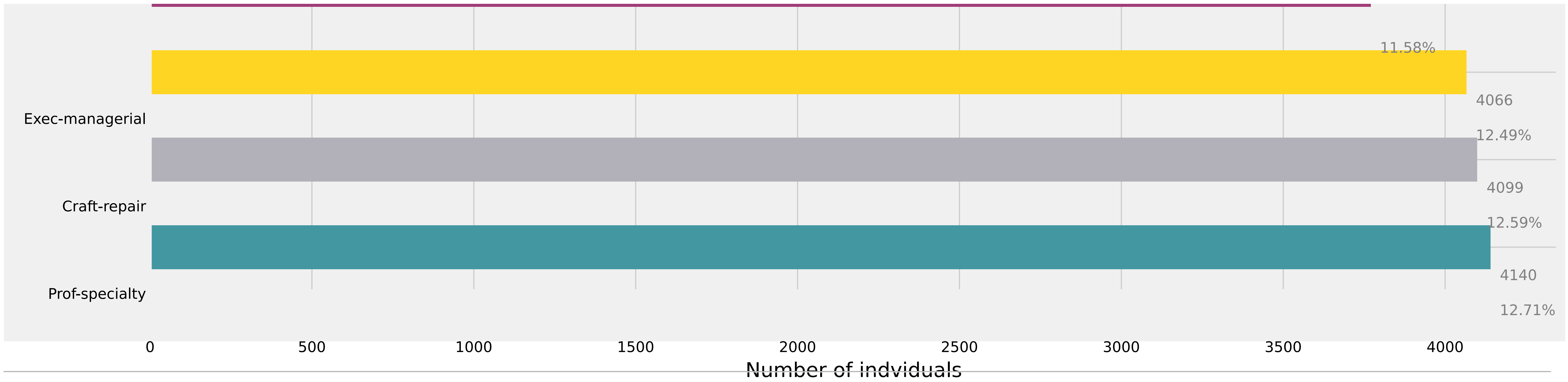

Number of individuals
14000
12000
10000
8000
6000
4000
69.43%
15128
30.57%
89.05%
9592
Salary less or equal to $50K a year
Salary greater than $50K a year



In [ ]:

```
occupation_data_barh_graph = dataframe['occupation'].value_counts().plot(kind='barh', color=['#4297A0','#B2B1B9','#FFD523','#A03C78','#ED8E7C','#CDF3A2','#93D9A3','#FAAD80','#01937C','#B5EAEA','#F38BA0'], fontsi
occupation_data_barh_graph.set_title("Occupation", fontsize=18)
occupation_data_barh_graph.set_xlabel("Number of indviduals", fontsize=18);
totals = []
for i in occupation_data_barh_graph.patches:
    totals.append(i.get_width())
total = sum(totals)

for i in occupation_data_barh_graph.patches:
    occupation_data_barh_graph.text(i.get_width()+.4, i.get_y()+.4, \
            "  "+str(round(i.get_width())), fontsize=13,color='grey')
for i in occupation_data_barh_graph.patches:
    occupation_data_barh_graph.text(i.get_width()+.20, i.get_y(), \
            "  "+str(round((i.get_width()/total)*100, 2))+'%', fontsize=13,color='grey')
plt.style.use('fivethirtyeight')
```


Occupation
Armed-Forces
9
0.03%
Priv-house-serv
149
0.46%
Protective-serv
649
1.99%
Tech-support
928
2.85%
Farming-fishing
994
3.05%
Handlers-cleaners
1370
4.21%
Transport-moving
1597
4.9%
?
1843
5.66%
Machine-op-inspct
2002
6.15%
Other-service
3295
10.12%
Sales
3650
11.21%
Adm-clerical
3770
11.58%



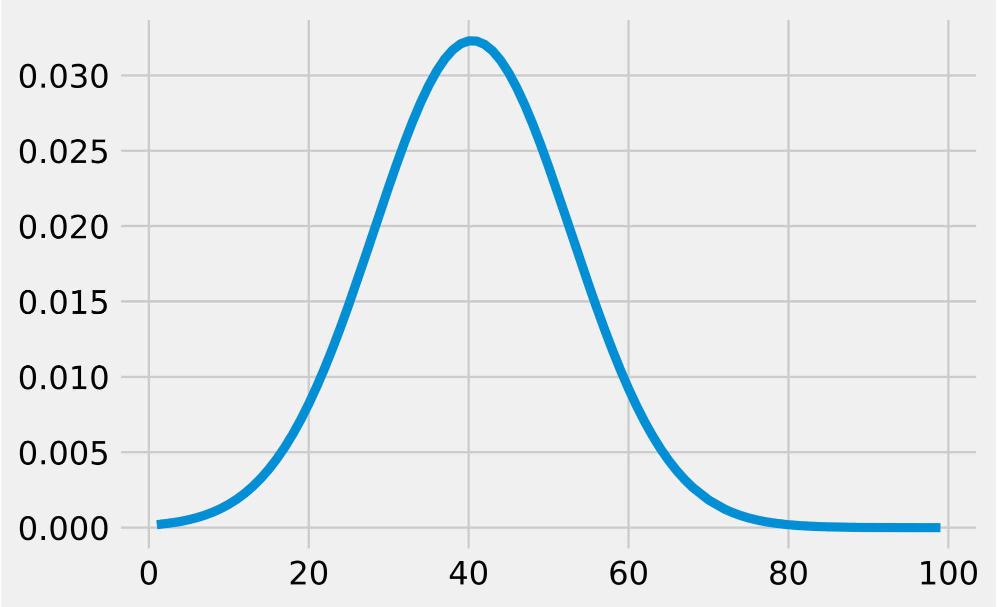
Gaussian Distributions

Gaussian Distribution of hours worked per week

In [ ]:

```
hours_per_week= dataframe['hours-per-week'].to_list()
hours_per_week.sort()
hours_per_week_mean= np.mean(hours_per_week)
hours_per_week_std = np.std(hours_per_week)
pdf = stats.norm.pdf(hours_per_week, hours_per_week_mean, hours_per_week_std)
plt.plot(hours_per_week, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week: "+str(dataframe['hours-per-week'].mean()))
```

```
Average hours worked per week: 40.437455852092995
```

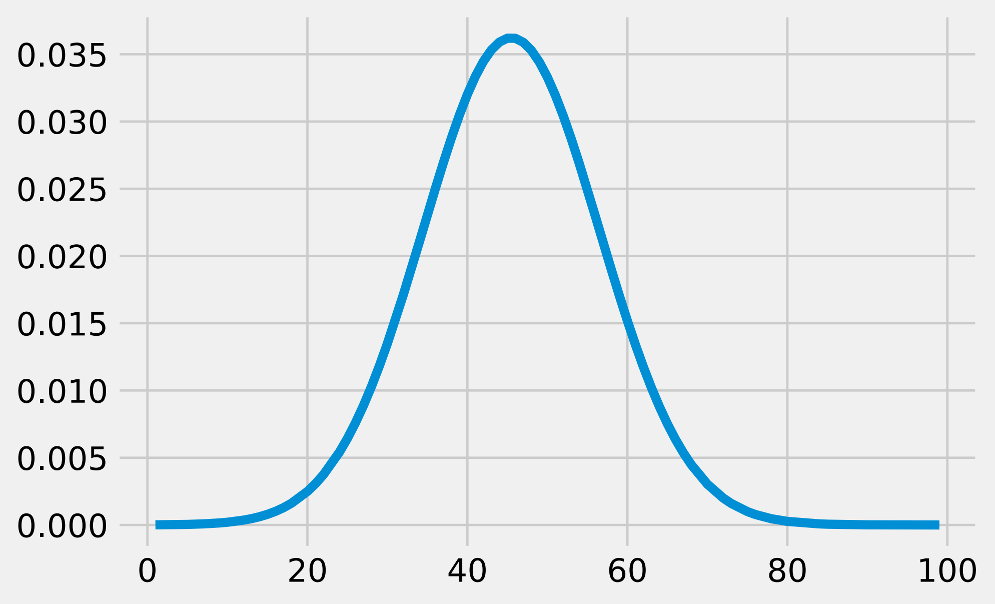


Gaussian Distribution of hours worked per week where salary is over 50k

In [ ]:

```
hours_per_week_salary_over_50k = dataframe['hours-per-week'][dataframe['salary']=='>50K'].to_list()
hours_per_week_salary_over_50k.sort()
hours_per_week_salary_over_50k_mean= np.mean(hours_per_week_salary_over_50k)
hours_per_week_salary_over_50k_std = np.std(hours_per_week_salary_over_50k)
pdf = stats.norm.pdf(hours_per_week_salary_over_50k, hours_per_week_salary_over_50k_mean, hours_per_week_salary_over_50k_std)
plt.plot(hours_per_week_salary_over_50k, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is greater than 50k: "+str(dataframe['hours-per-week'][dataframe['salary']=='>50K'].mean()))
```

```
Average hours worked per week were salary is greater than 50k: 45.473026399693914
```

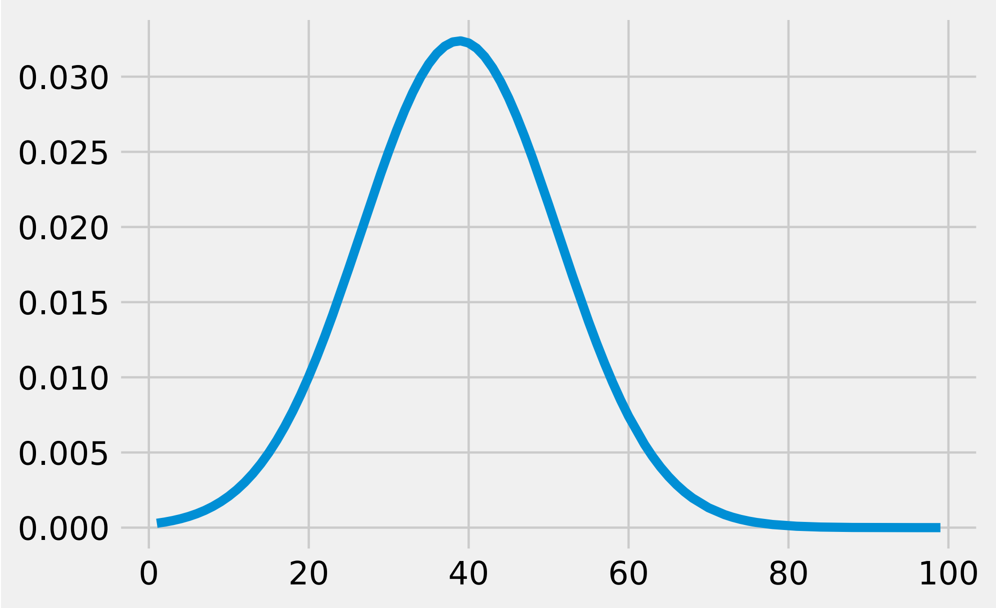


Gaussian Distribution of hours worked per week where salary is under or equal 50k

In [ ]:

```
hours_per_week_salary_under_50k = dataframe['hours-per-week'][dataframe['salary']=='<=50K'].to_list()
hours_per_week_salary_under_50k.sort()
hours_per_week_salary_under_50k_mean= np.mean(hours_per_week_salary_under_50k)
hours_per_week_salary_under_50k_std = np.std(hours_per_week_salary_under_50k)
pdf = stats.norm.pdf(hours_per_week_salary_under_50k,hours_per_week_salary_under_50k_mean, hours_per_week_salary_under_50k_std)
plt.plot(hours_per_week_salary_under_50k, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is less than or equal to 50k: "+str(dataframe['hours-per-week'][dataframe['salary']=='<=50K'].mean()))
```

```
Average hours worked per week were salary is less than or equal to 50k: 38.840210355987054
```

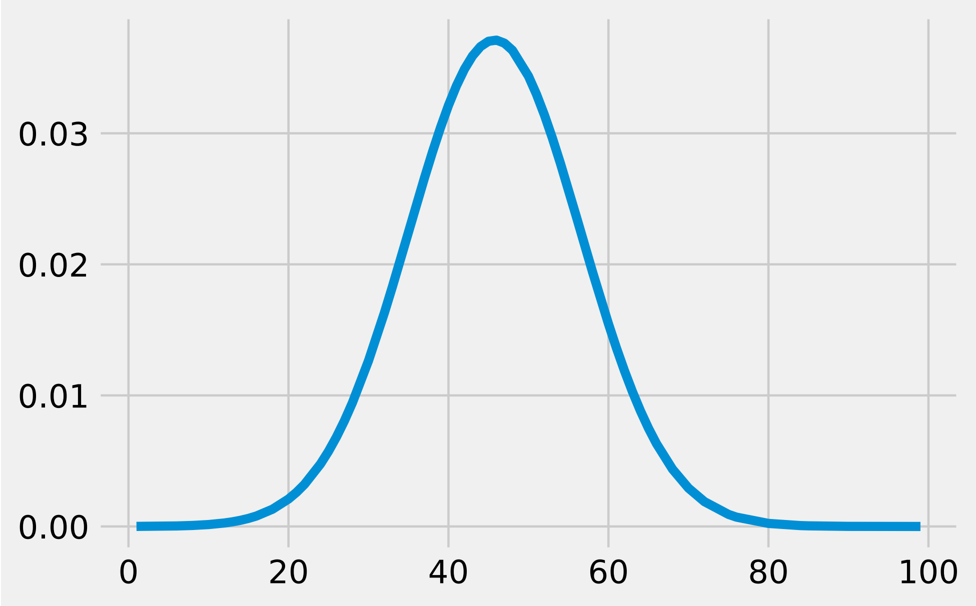


Gaussian Distribution of hours worked per week where salary is over 50k and individual has obtained a degree of some sort (Bachelors or Masters or Doctorate)

In [ ]:

```
hours_per_week_salary_over_50k_educated = dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K']['hours-per-week'].to_list()
hours_per_week_salary_over_50k_educated.sort()
hours_per_week_salary_over_50k_educated_mean= np.mean(hours_per_week_salary_over_50k_educated)
hours_per_week_salary_over_50k_educated_std = np.std(hours_per_week_salary_over_50k_educated)
pdf = stats.norm.pdf(hours_per_week_salary_over_50k_educated,hours_per_week_salary_over_50k_educated_mean, hours_per_week_salary_over_50k_educated_std)
plt.plot(hours_per_week_salary_over_50k_educated, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salar
```

```
Average hours worked per week were salary is greater than 50k and individual has obtained a degree: 45.77596098680436
<ipython-input-27-84119caf7943>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  hours_per_week_salary_over_50k_educated = dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K']['hours-per-week'].to_list()
<ipython-input-27-84119caf7943>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sal
ary']== '>50K']['hours-per-week'].mean()))
```

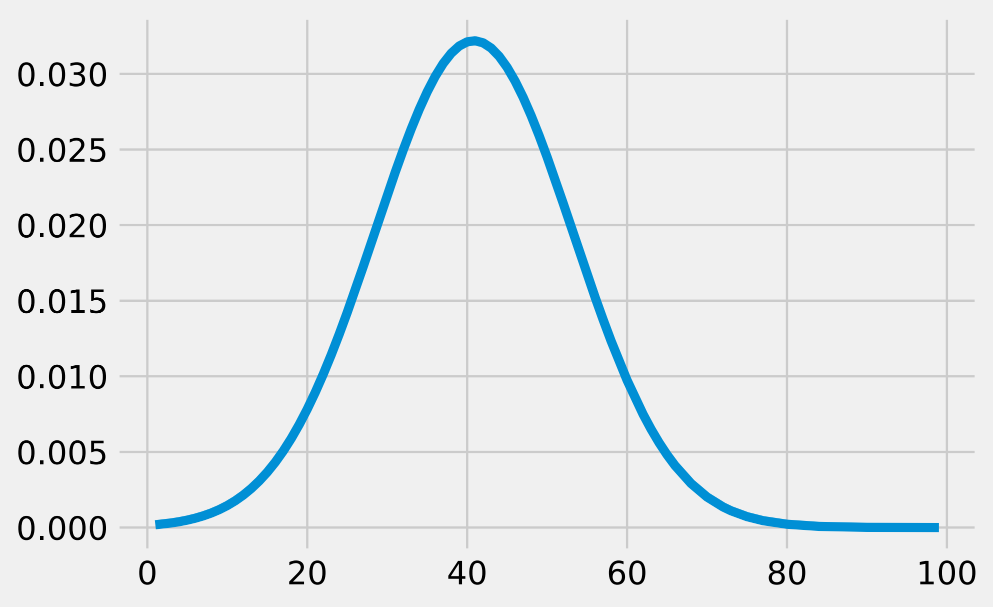


Gaussian Distribution of hours worked per week where salary is under or equal to 50k and individual has obtained a degree of some sort (Bachelors or Masters or Doctorate)

In [ ]:

```
hours_per_week_salary_under_50k_educated = dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '<=50K']['hours-per-week'].to_list()
hours_per_week_salary_under_50k_educated .sort()
hours_per_week_salary_under_50k_educated_mean= np.mean(hours_per_week_salary_under_50k_educated)
hours_per_week_salary_under_50k_educated_std = np.std(hours_per_week_salary_under_50k_educated)
pdf = stats.norm.pdf(hours_per_week_salary_under_50k_educated,hours_per_week_salary_under_50k_educated_mean, hours_per_week_salary_under_50k_educated_std)
plt.plot(hours_per_week_salary_under_50k_educated, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salar
```

```
<ipython-input-28-a62d17cb7b50>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  hours_per_week_salary_under_50k_educated = dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '<=50K']['hours-per-week'].to_list()
<ipython-input-28-a62d17cb7b50>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sal
ary']== '<=50K']['hours-per-week'].mean()))
Average hours worked per week were salary is greater than 50k and individual has obtained a degree: 40.83720349563046
```

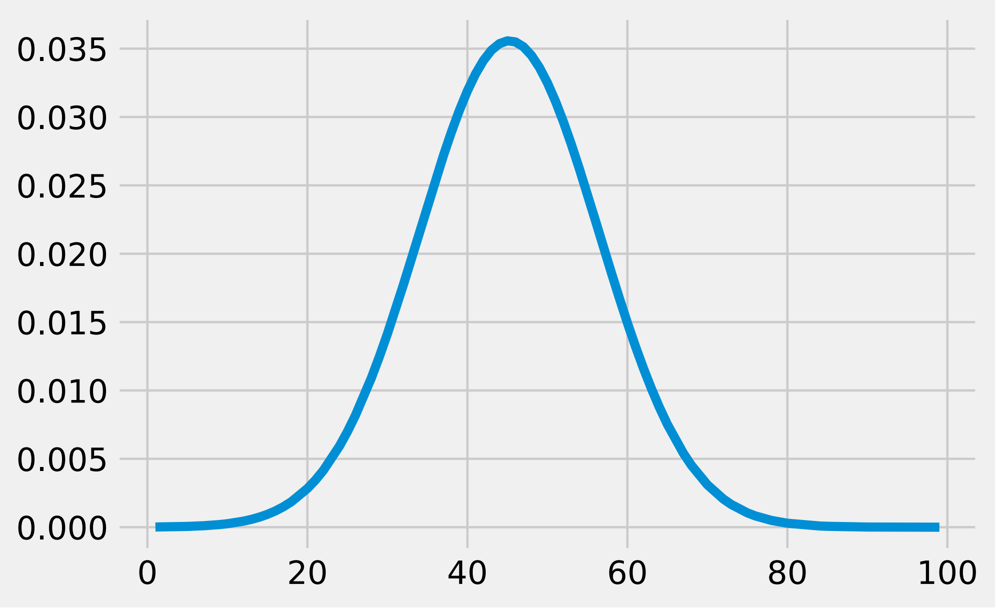


Gaussian Distribution of hours worked per week where salary is over 50k and individual has not obtained a degree of some sort (Bachelors or Masters or Doctorate)

In [ ]:

```
hours_per_week_salary_over_50k_noneducated = dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K']['hours-per-week'].to_list()
hours_per_week_salary_over_50k_noneducated.sort()
hours_per_week_salary_over_50k_noneducated_mean= np.mean(hours_per_week_salary_over_50k_noneducated)
hours_per_week_salary_over_50k_noneducated_std = np.std(hours_per_week_salary_over_50k_noneducated)
pdf = stats.norm.pdf(hours_per_week_salary_over_50k_noneducated,hours_per_week_salary_over_50k_noneducated_mean, hours_per_week_salary_over_50k_noneducated_std)
plt.plot(hours_per_week_salary_over_50k_noneducated, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sala
```

```
<ipython-input-29-a6f0b91294f0>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  hours_per_week_salary_over_50k_noneducated = dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '>50K']['hours-per-week'].to_list()
<ipython-input-29-a6f0b91294f0>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sa
lary']== '>50K']['hours-per-week'].mean()))
Average hours worked per week were salary is greater than 50k and individual has obtained a degree: 45.23053960964409
```

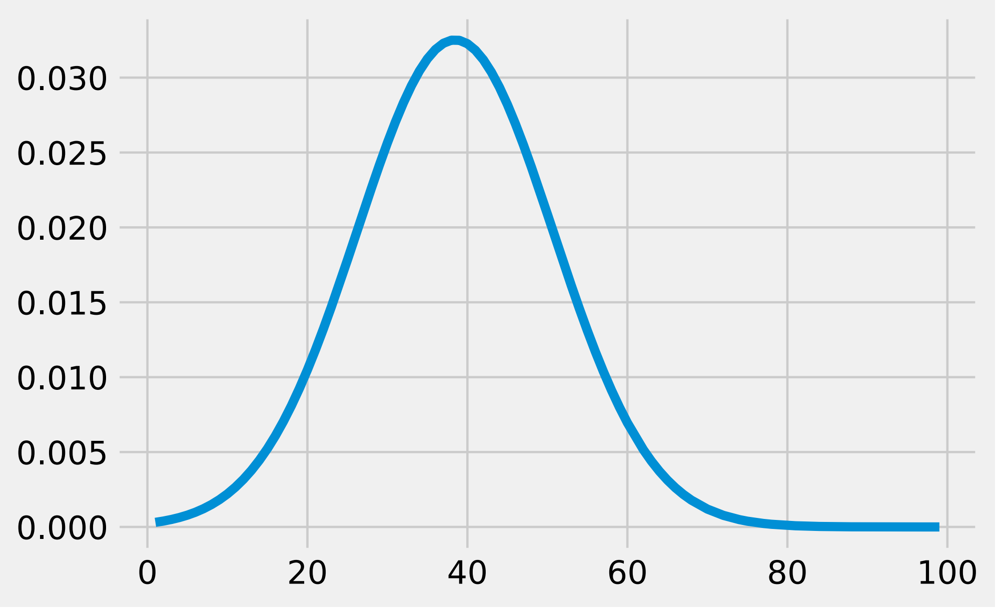


Gaussian Distribution of hours worked per week where salary is under or equal to 50k and individual has not obtained a degree of some sort (Bachelors or Masters or Doctorate)

In [ ]:

```
hours_per_week_salary_under_50k_noneducated = dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '<=50K']['hours-per-week'].to_list()
hours_per_week_salary_under_50k_noneducated.sort()
hours_per_week_salary_under_50k_noneducated_mean= np.mean(hours_per_week_salary_under_50k_noneducated)
hours_per_week_salary_under_50k_noneducated_std = np.std(hours_per_week_salary_under_50k_noneducated)
pdf = stats.norm.pdf(hours_per_week_salary_under_50k_noneducated,hours_per_week_salary_under_50k_noneducated_mean, hours_per_week_salary_under_50k_noneducated_std)
plt.plot(hours_per_week_salary_under_50k_noneducated, pdf)
plt.style.use('fivethirtyeight')
print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sala
```

```
<ipython-input-30-15a35d975776>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  hours_per_week_salary_under_50k_noneducated = dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['salary']== '<=50K']['hours-per-week'].to_list()
<ipython-input-30-15a35d975776>:8: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  print("Average hours worked per week were salary is greater than 50k and individual has obtained a degree: "+str(dataframe.loc[~dataframe['education'].isin(['Bachelors', 'Masters', 'Doctorate'])][dataframe['sa
lary']== '<=50K']['hours-per-week'].mean()))
Average hours worked per week were salary is greater than 50k and individual has obtained a degree: 38.45411537533189
```


0.030
0.025
0.020
0.015
0.010
0.005
0.000
0
20
40
60
80
100