[redacted]

[redacted]

1 Program overview

- Exploratory data analysis
- Look at relationships between income & different attributes.
- Extract important features from a Random Forest Model.

In [1]:
```
import os
os.chdir(r"C:\Users\Arnold\OneDrive\R_Python_working_directory")
import pandas as pd
import numpy as np
from plotnine import *
%matplotlib inline
(theme_update(plot_background = element_rect(fill = "gold"),panel_background = element_
colour = "blue",size = 1.99,linetype = "solid"),plot_title = element_text(hjust = 0.5))
from sklearn.metrics import confusion_matrix as cm
```

2 Read in data

In [2]:
```
df=pd.read_csv('cencus income.csv',na_values='?')
```

3 Check the shape

In [3]:
```
df.shape
```

Out[3]: (48842, 15)

4 Check % of rows with missing values

In [4]:
```
np.mean(pd.isnull(df).any(axis=1))*100
```

Out[4]: 7.411653904426519

7% is low, so I'll drop the rows with missing values

5 Drop rows with missing values

In [5]:
```
df.dropna(inplace=1==1)
```

6 Age binning

```
In [6]: df.age=pd.cut(df.age,bins=[16,20,30,40,50,60,70,100])
        df.age=df.age.astype(str)
```

7 Remove columns that I'm not going to use

```
In [7]: df.drop(columns=['relationship','capital-gain','capital-loss','hours-per-week'],
                inplace=1==1)
```

8 Data preview

```
In [8]: df.head()
```

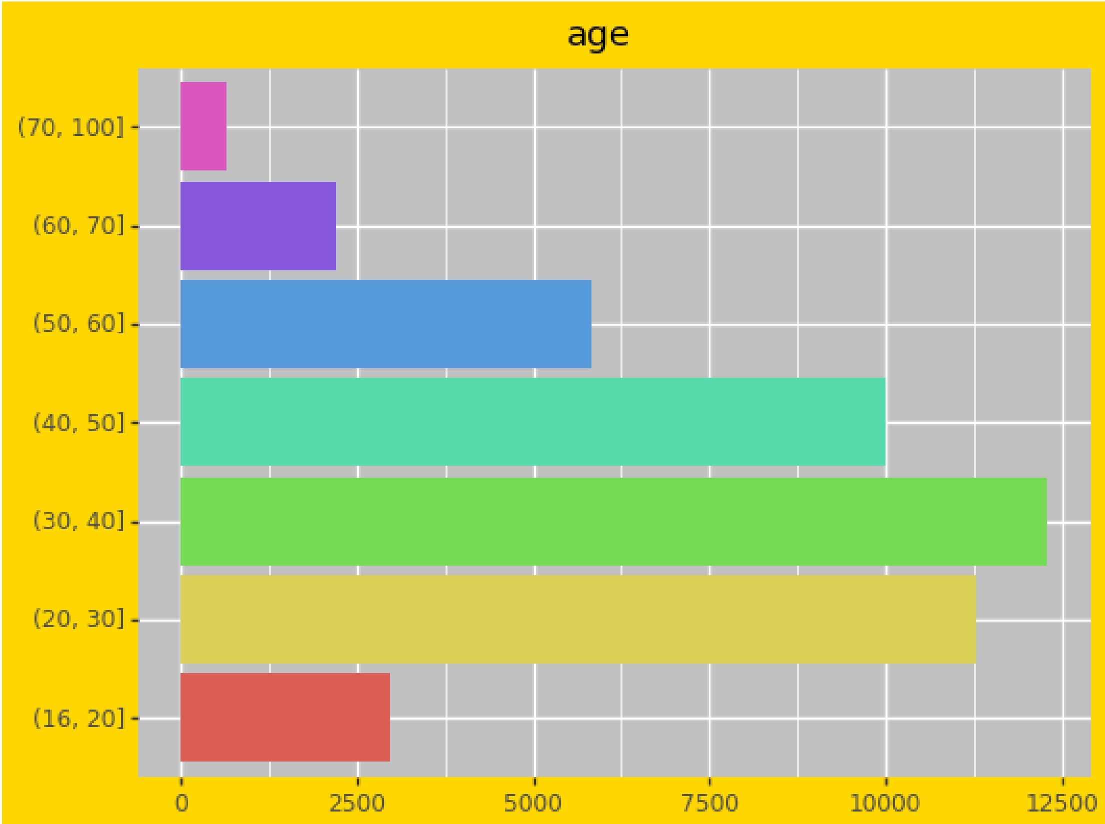
```
Out[8]:         age  workclass  fnlwgt     education  educational-num  \
        0  (20, 30]    Private  226802          11th                7
        1  (30, 40]    Private   89814       HS-grad                9
        2  (20, 30]  Local-gov  336951    Assoc-acdm               12
        3  (40, 50]    Private  160323  Some-college               10
        5  (30, 40]    Private  198693          10th                6

               marital-status         occupation   race gender native-country income
        0       Never-married  Machine-op-inspct  Black   Male  United-States  <=50K
        1  Married-civ-spouse    Farming-fishing  White   Male  United-States  <=50K
        2  Married-civ-spouse    Protective-serv  White   Male  United-States   >50K
        3  Married-civ-spouse  Machine-op-inspct  Black   Male  United-States   >50K
        5       Never-married      Other-service  White   Male  United-States  <=50K
```

9 Data summary

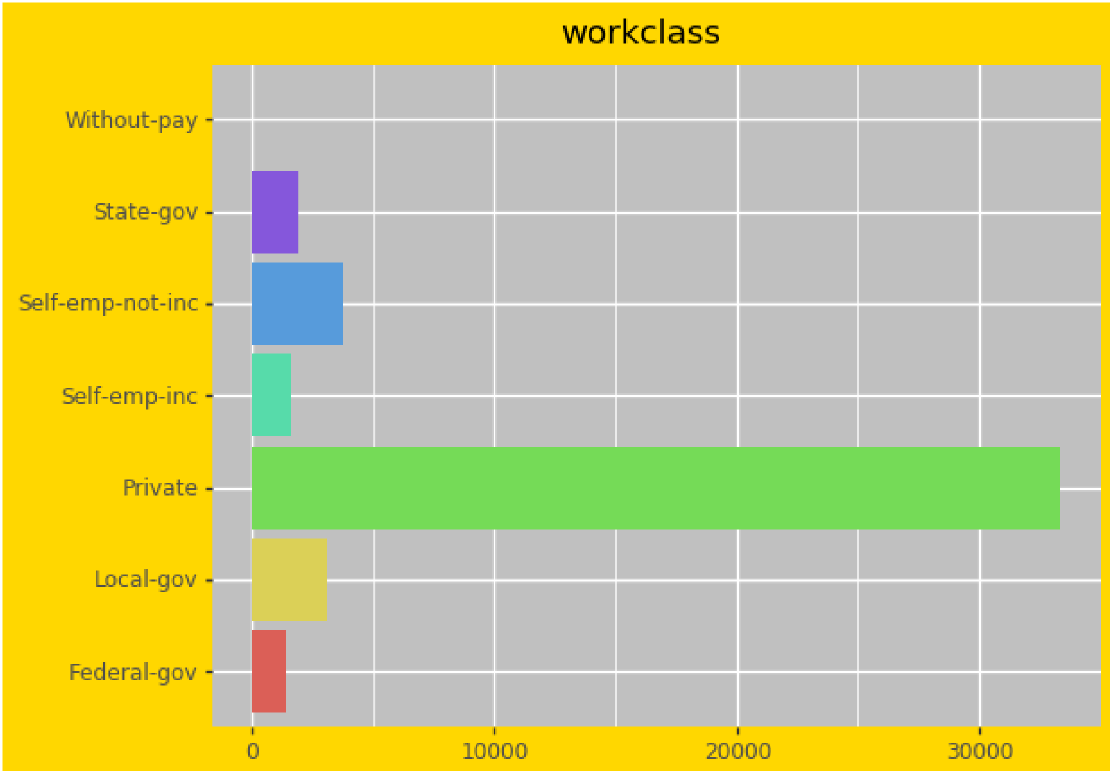
9.1 Function to make bargraphs for all columns from a given data frame

```
In [9]: def ggbar(data=df):
            for clm in data.columns:
                if clm == 'native-country':
                    print('U.S.A with the count of 41292')
                    print(ggplot(data[df.loc[:,'native-country'] != 'United-States'],aes(clm))
                    show_legend=1==0)+coord_flip()+xlab('')+ylab(''))
                elif data.loc[:,clm].dtype == 'O':
                    print(ggplot(data,aes(clm))+geom_bar(aes(fill=clm),show_legend=1==0)+\
                          coord_flip()+xlab('')+ylab('')+ggtitle(clm))
```

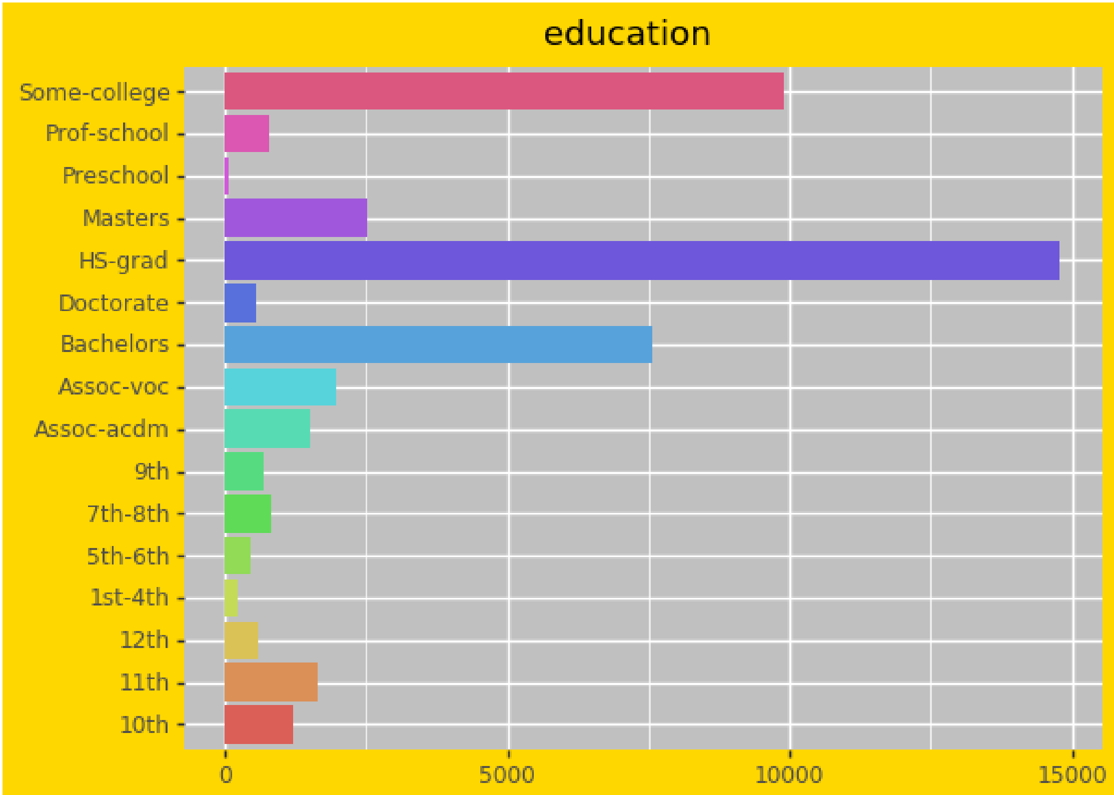
```
In [10]: ggbar()
```



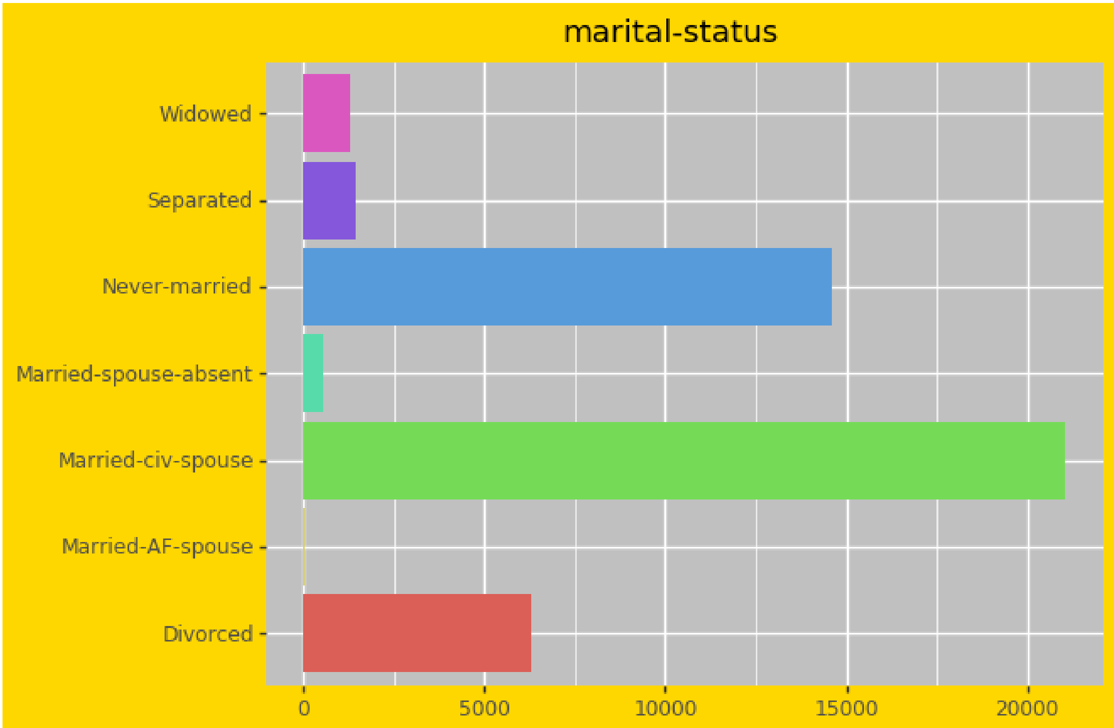

```
<ggplot: (-9223371897051248461)>
```



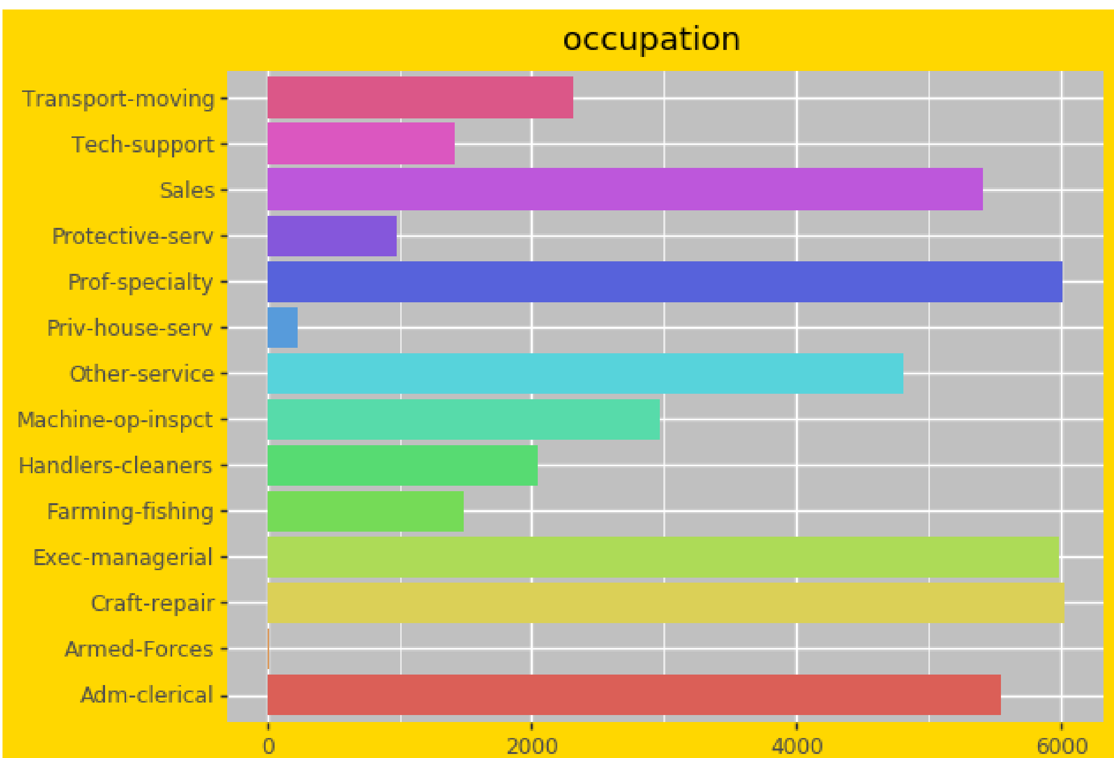

```
<ggplot: (-9223371897050902314)>
```



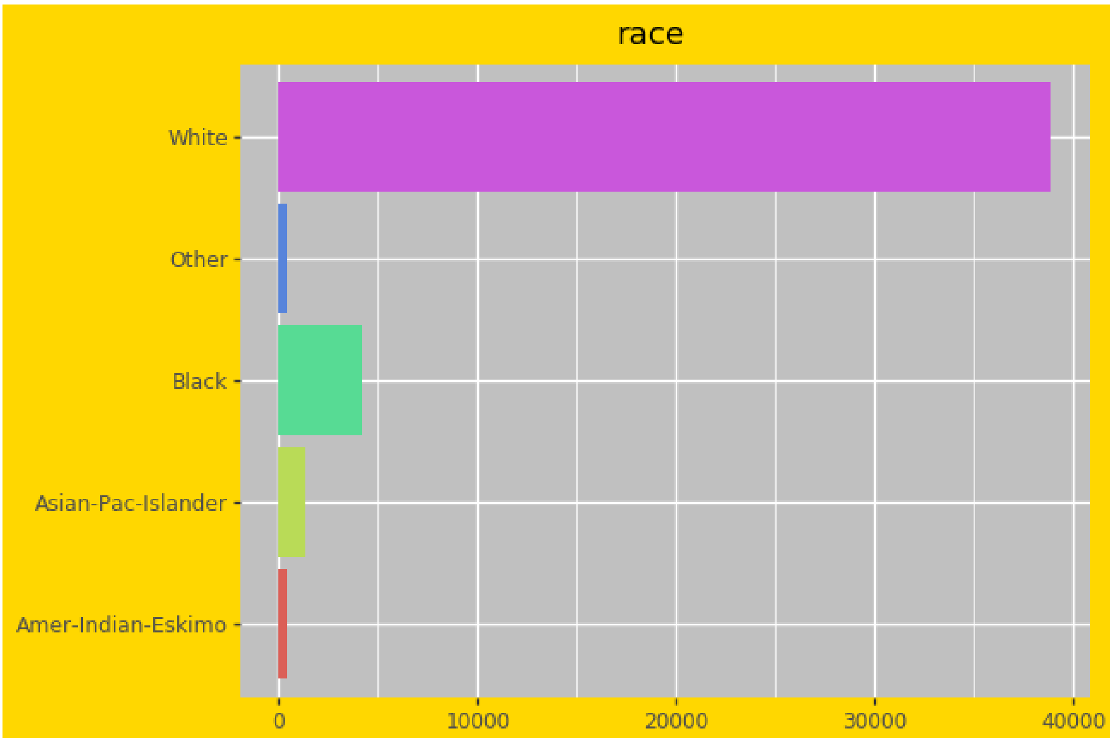

```
<ggplot: (139803527330)>
```



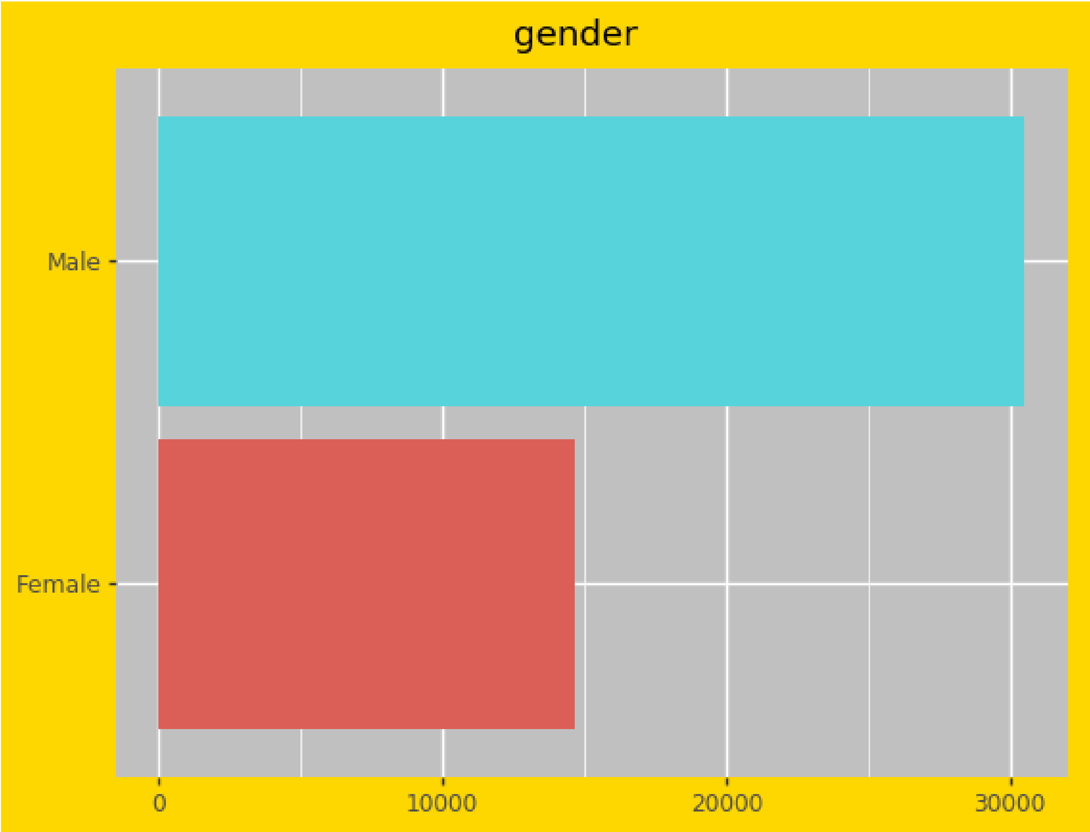
<ggplot: (139803924565)>



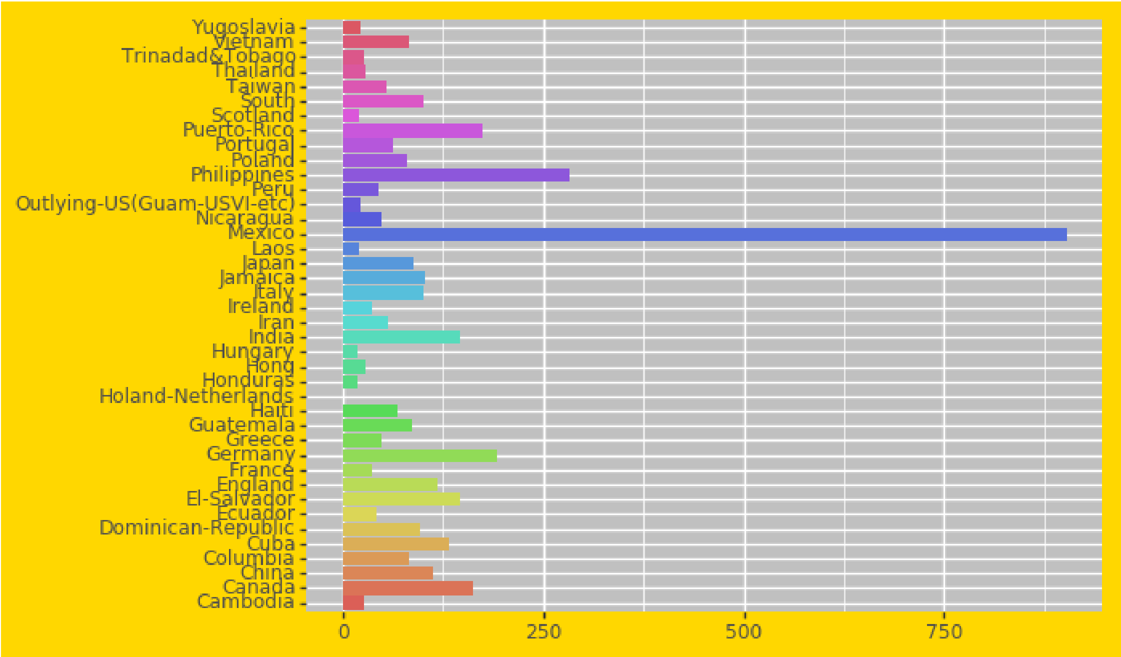
```
<ggplot: (-9223371897050851163)>
```



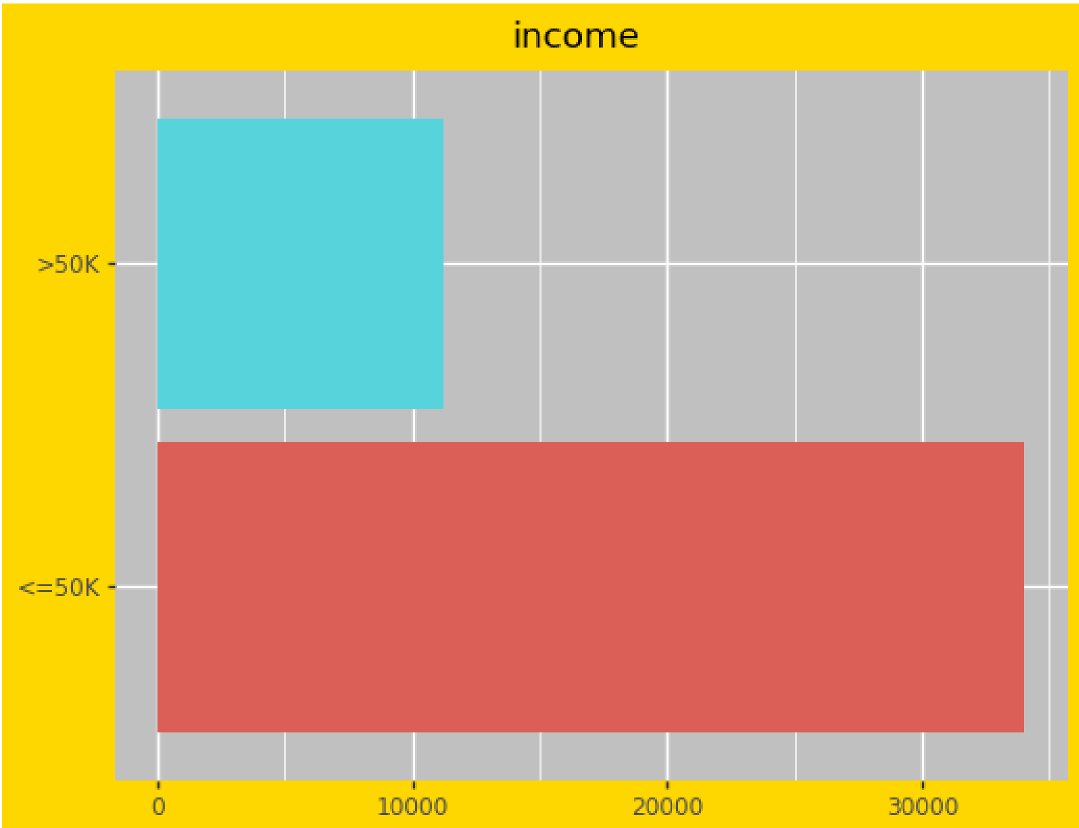

```
<ggplot: (-9223371897049750370)>
```




```
<ggplot: (-9223371897050854747)>
U.S.A with the count of 41292
```



```
<ggplot: (-9223371897049623457)>
```




```
<ggplot: (139805213679)>
```

9.2 Function to look at the relationships between different attributes & income

9.2.1 This function only works with categorical variables.

```
In [11]: def relation(attr):
             tdf=df.pivot_table(values='fnlwgt',columns=['income'],index=attr,aggfunc=len,
                                margins=1==1,fill_value=0)
             tdf.iloc[:-1,0]=tdf.iloc[:-1,0].values/(tdf.All[:-1]).values
             tdf.iloc[:-1,1]=1-tdf.iloc[:-1,0].values
             tdf.sort_values(by=['>50K'],inplace=1==1,ascending=1==0)
             tdf.iloc[1:,:2]=tdf.iloc[:,:2].applymap(lambda x: round(x*100))
             tdf=tdf.iloc[1:,:-1].stack().reset_index()
             tdf.rename(columns={0:'%'},inplace=1==1)
```

```
        return tdf
    def rggplot(attr,size=8):
        graph=ggplot(tdf,aes(x=attr,y='%',fill=attr))+geom_bar(stat='identity',
                                show_legend=1==0)+geom_text(aes(label='%',y=5),
        size=size,format_string='{}%')+coord_flip()+ggtitle('% of income > 50K')
        return graph
```

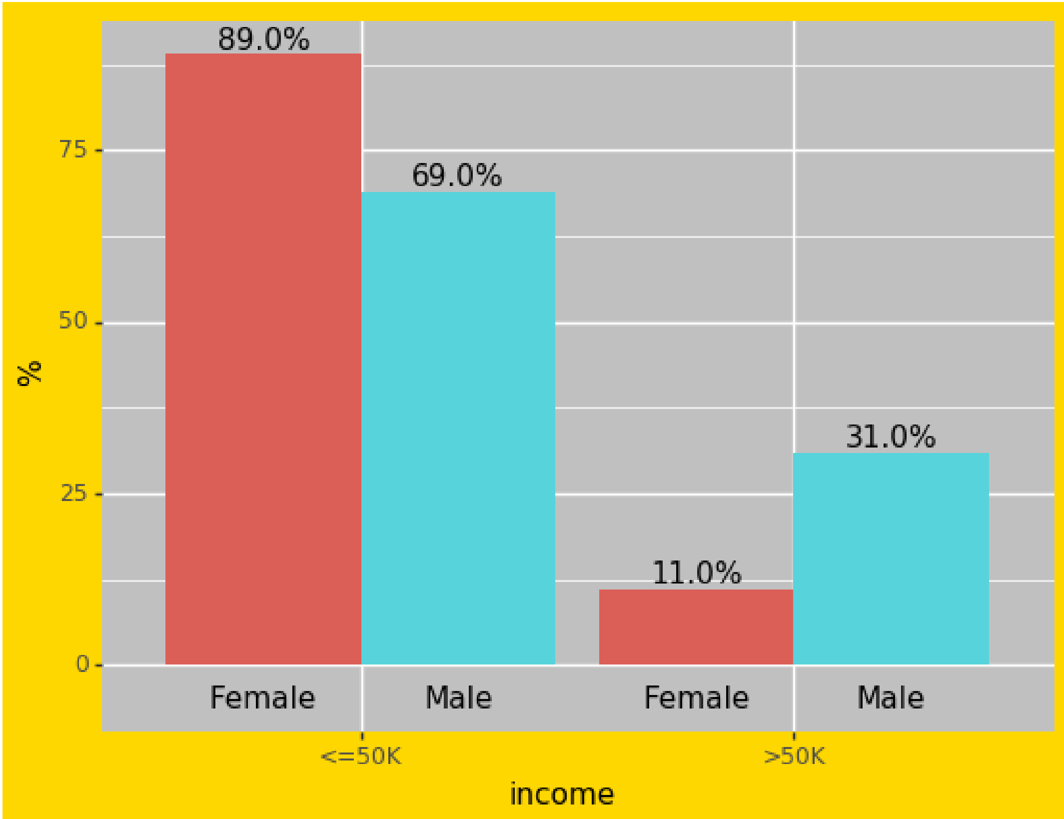
10 Gender vs Income

```
In [12]: tdf=relation('gender')
         tdf
```

```
Out[12]:    gender income     %
         0    Male  <=50K  69.0
         1    Male   >50K  31.0
         2  Female  <=50K  89.0
         3  Female   >50K  11.0
```

```
In [13]: (ggplot(tdf,aes(x='income',y='%',fill='gender'))+geom_bar(stat='identity',position=
         'dodge',show_legend=1==0)+geom_text(aes(y=-5,label='gender'),position=
         position_dodge(width=0.9))+geom_text(aes(label='%'),position=
         position_dodge(width=0.9),va='bottom', format_string='{}%'))
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```




```
Out[13]: <ggplot: (-9223371897049452491)>
```

We can see that males are more likely to have high income than females.

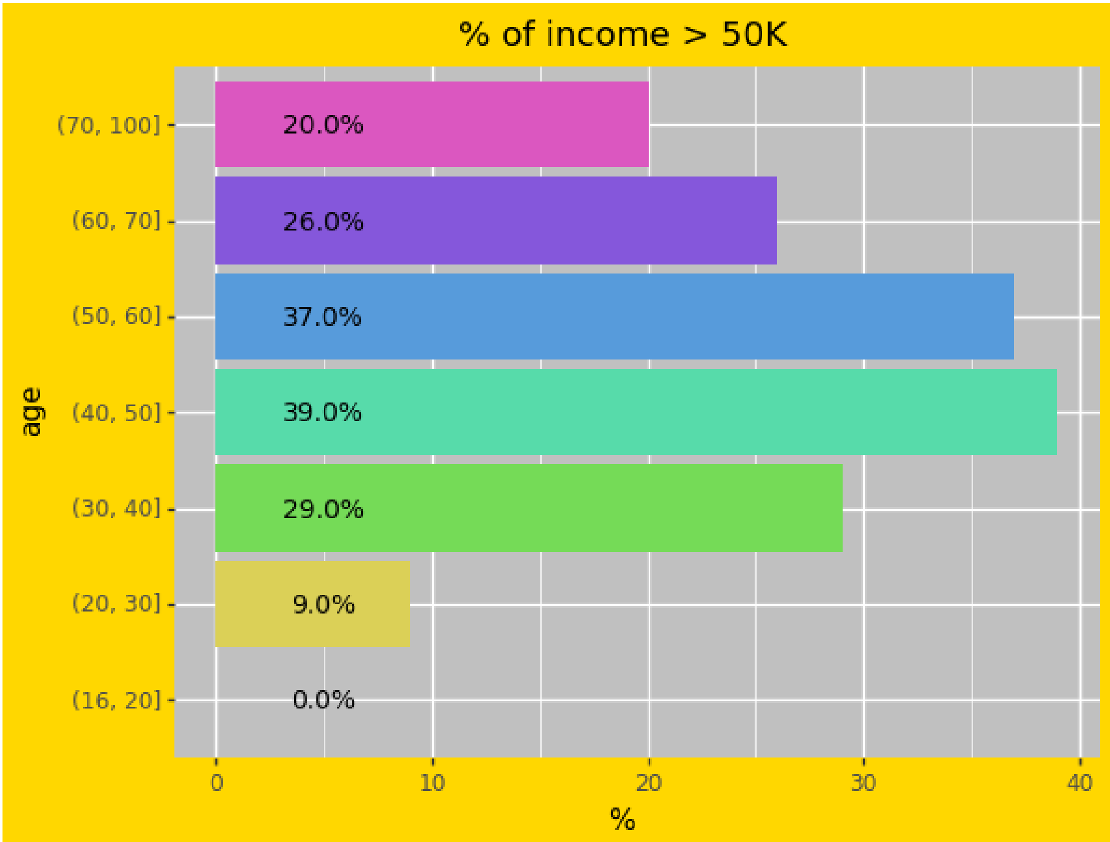
11 Age VS Income

```
In [14]: tdf=relation('age')
         tdf=tdf[tdf.income=='>50K']
         tdf
```

```
Out[14]:            age  income     %
         1     (40, 50]    >50K  39.0
         3     (50, 60]    >50K  37.0
         5     (30, 40]    >50K  29.0
         7     (60, 70]    >50K  26.0
         9    (70, 100]    >50K  20.0
         11    (20, 30]    >50K   9.0
         13    (16, 20]    >50K   0.0
```

```
In [15]: rggplot('age',10)
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```




```
Out[15]: <ggplot: (139805330359)>
```

We can see that 40~60 years old people are more likely to have high income than other age groups.

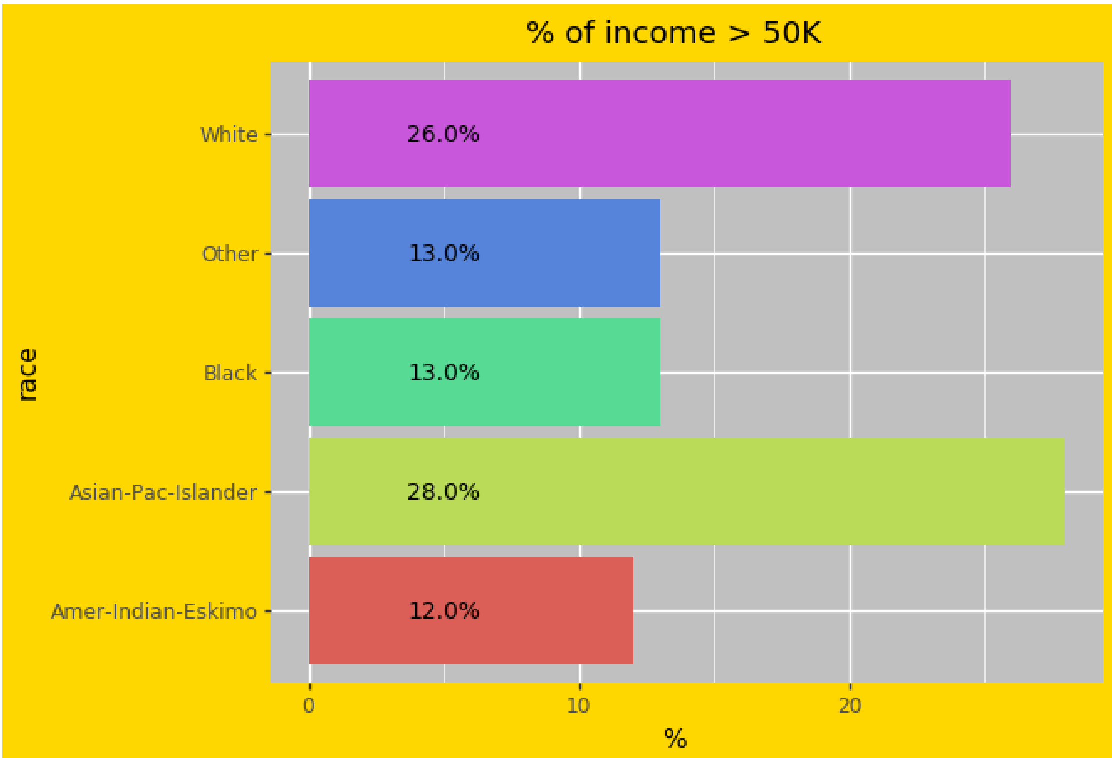
12 Race VS Income

```
In [16]: tdf=relation('race')
         tdf=tdf[tdf.income=='>50K']
         tdf
```

```
Out[16]:                  race income     %
         1  Asian-Pac-Islander   >50K  28.0
         3               White   >50K  26.0
         5               Other   >50K  13.0
         7               Black   >50K  13.0
         9  Amer-Indian-Eskimo   >50K  12.0
```

```
In [17]: rggplot('race',10)
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```




```
Out[17]: <ggplot: (-9223371897049251226)>
```

We can see that Asian-Pac-Islander & White races are more likely to have high income than other races.

13 Occupation VS Income

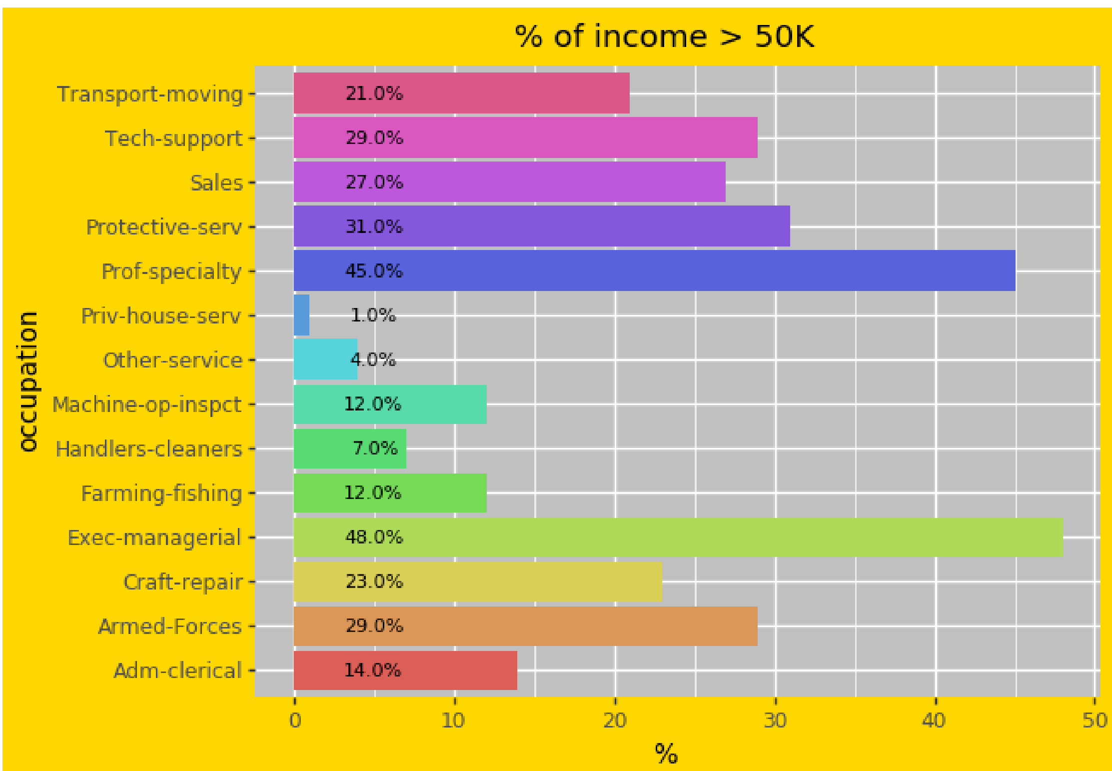
```
In [18]: tdf=relation('occupation')
         tdf=tdf[tdf.income=='>50K']
         tdf
```

```
Out[18]:          occupation income     %
         1    Exec-managerial   >50K  48.0
         3     Prof-specialty   >50K  45.0
         5    Protective-serv   >50K  31.0
         7       Tech-support   >50K  29.0
         9       Armed-Forces   >50K  29.0
```

```
11                Sales    >50K   27.0
13          Craft-repair    >50K   23.0
15      Transport-moving    >50K   21.0
17          Adm-clerical    >50K   14.0
19     Machine-op-inspct    >50K   12.0
21       Farming-fishing    >50K   12.0
23     Handlers-cleaners    >50K    7.0
25         Other-service    >50K    4.0
27       Priv-house-serv    >50K    1.0
```

```
In [19]: rggplot('occupation')
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```




```
Out[19]: <ggplot: (139805540735)>
```

We see that Exec-managerial, Prof-specialty, & Protective-serv are more likely to have high income than other occupations.

14 Workclass VS Income

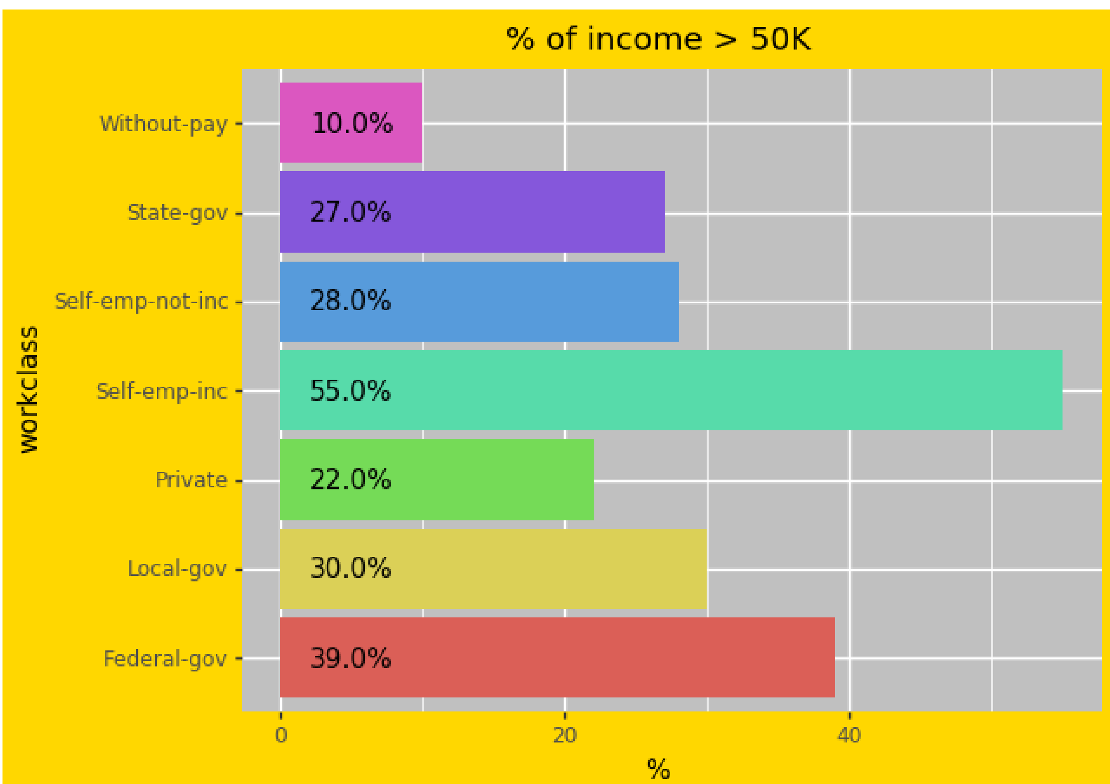
In [20]:
```
tdf=relation('workclass')
tdf=tdf[tdf.income=='>50K']
tdf
```

Out[20]:

	workclass	income	%
1	Self-emp-inc	>50K	55.0
3	Federal-gov	>50K	39.0
5	Local-gov	>50K	30.0
7	Self-emp-not-inc	>50K	28.0
9	State-gov	>50K	27.0
11	Private	>50K	22.0
13	Without-pay	>50K	10.0

In [21]:
```
(ggplot(tdf,aes('workclass','%',fill='workclass'))+geom_bar(stat='identity',
show_legend=1==0)+geom_text(aes(label='%',y=5),format_string='{}%')+\
 coord_flip()+ggtitle('% of income > 50K'))
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```



Out[21]: <ggplot: (-9223371897049765190)>

We see that Self-emp-inc, Federal-gov, Local-gov are more likely to have high income than other workclass.

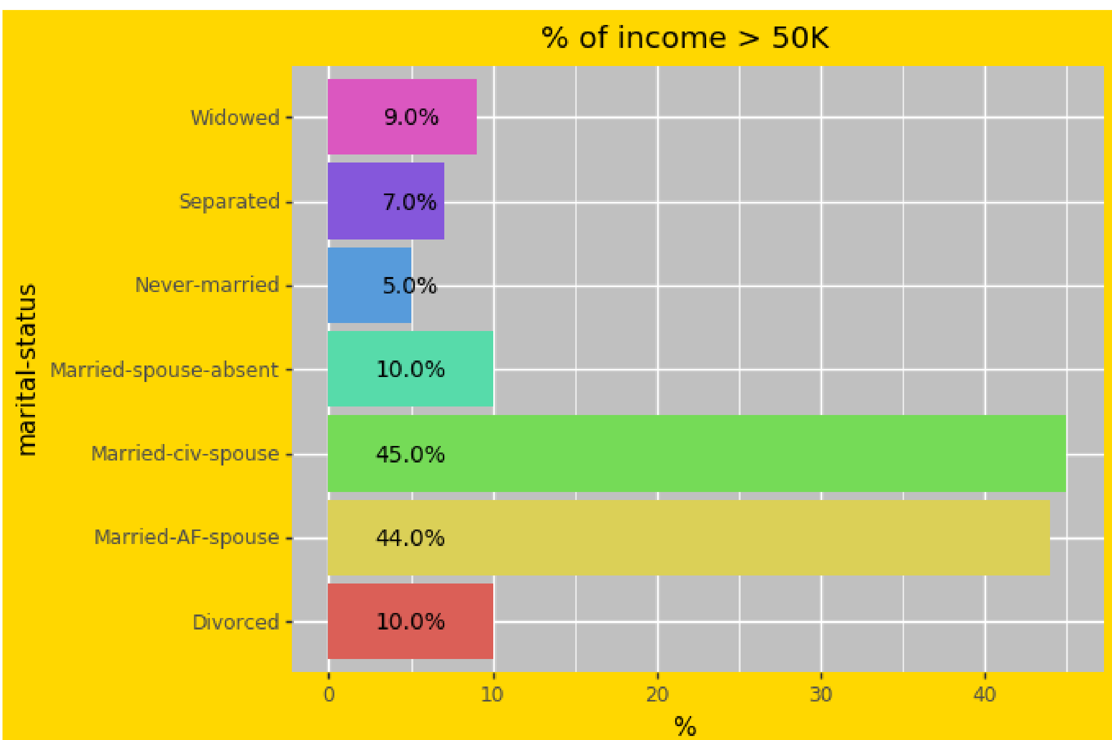
15 Marital-status VS Income

```
In [22]: tdf=relation('marital-status')
         tdf=tdf[tdf.income=='>50K']
         tdf
```

```
Out[22]:            marital-status income     %
         1      Married-civ-spouse   >50K  45.0
         3       Married-AF-spouse   >50K  44.0
         5                Divorced   >50K  10.0
         7   Married-spouse-absent   >50K  10.0
         9                 Widowed   >50K   9.0
         11              Separated   >50K   7.0
         13          Never-married   >50K   5.0
```

```
In [23]: rggplot('marital-status',10)
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```



```
Out[23]: <ggplot: (139805013340)>
```

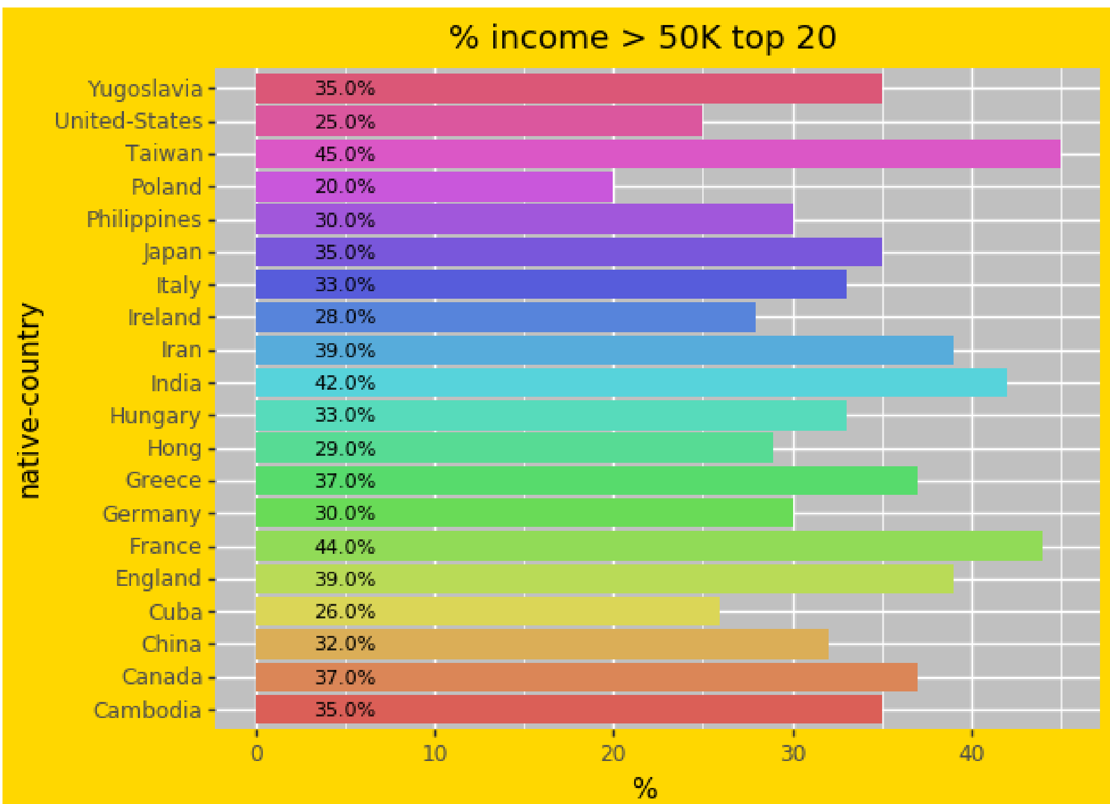
We see people who has the marital status of Married-civ-spouse, or Married-AF-spouse are much more likely to have high income than others.

16 Native-country VS Income

```
In [24]: tdf=relation('native-country')
         tdf=tdf[tdf.income=='>50K']
         tdf
         tdf=tdf.iloc[:20,:]
```

```
In [25]: rggplot('native-country')+ggtitle('% income > 50K top 20')
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```




```
Out[25]: <ggplot: (-9223371897049405524)>
```

- We see that top 3 native country that have higher % of high income individuals are Taiwan, France, & India.
- U.S.A. as native country is ranked in the middle with 25% of people with high income.

17 Random Forest model

```
In [26]: from custom import tts
         from sklearn.ensemble import RandomForestClassifier as rfc
```

17.1 Getting data ready for modeling

```
In [27]: from sklearn.preprocessing import LabelEncoder as le
         from pandas import get_dummies as getd

In [28]: tdf=df.loc[:,['educational-num','workclass','marital-status','occupation','race',
                        'gender','native-country','income']]

In [29]: tdf.loc[:,'age']=le().fit_transform(df.age)
         clm=['workclass','marital-status','occupation','race','gender','native-country',
         'income']
         tdf=getd(tdf,columns=clm,prefix=clm,drop_first=1==1)
```

17.2 Split data into training and testing group

```
In [30]: xtr,xte,ytr,yte=tts(tdf,dep='income_>50K')

C:\Anocanda\lib\site-packages\sklearn\model_selection\_split.py:2069: FutureWarning: From vers
  FutureWarning)
```

17.3 Model fitting

```
In [31]: mod=rfc(n_estimators=100).fit(xtr,ytr)
```

17.4 Look at feature importance

```
In [32]: imp=pd.DataFrame({'Attribute':tdf.columns[:-1],
         'Score':mod.feature_importances_.round(2)})
```

17.5 Clean up

```
In [33]: def clean(string):
             if '_' in string:
                 return string[:string.find('_')]
             return string

In [34]: imp.Attribute=imp.Attribute.map(clean)

In [35]: tdf=imp.groupby('Attribute').mean().reset_index().sort_values('Score',ascending=1==0)
         tdf
```

```
Out[35]:           Attribute     Score
         1   educational-num  0.240000
         0               age  0.150000
         3    marital-status  0.046667
         2            gender  0.040000
         5        occupation  0.013077
         7         workclass  0.011667
         6              race  0.007500
         4    native-country  0.000500
```

18 Confusion Martrix

```
In [46]: lb=['<=50K','>50K']
         tdf=pd.DataFrame(cm(yte,mod.predict(xte)),columns=lb,index=lb)
         tdf.columns.name='Actual Value';tdf.index.name='Prediction'
         tdf
```

```
Out[46]: Actual Value  <=50K  >50K
         Prediction
         <=50K          9226   979
         >50K           1500  1863
```

- Model does a great job on predicting income <= 50K, but not > 50K.

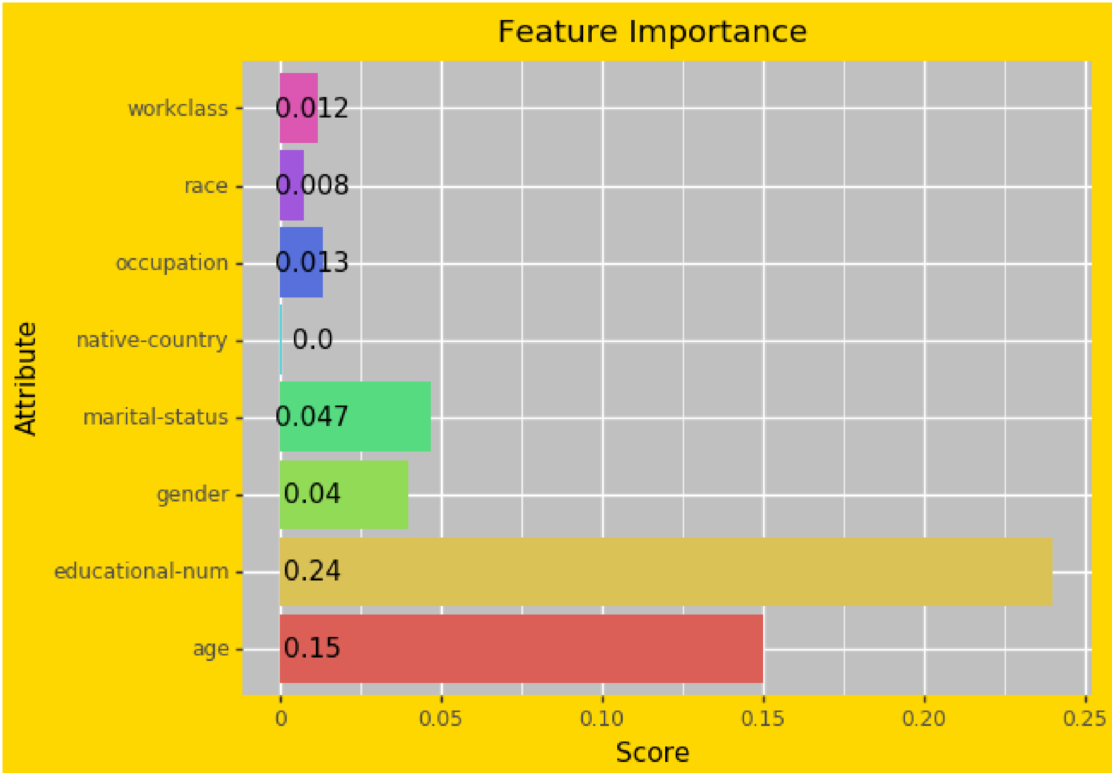
18.1 Accuracy of the model

```
In [37]: mod.score(xte,yte)
```

```
Out[37]: 0.8172906839622641
```

```
In [36]: ggplot(tdf,aes('Attribute','Score',fill='Attribute'))+geom_bar(stat='identity',
         show_legend=1==0)+geom_text(aes(label='round(Score,3)',y=.01))+coord_flip()+\
         labs(title='Feature Importance',caption='Random Forest')
```

```
C:\Anocanda\lib\site-packages\plotnine\layer.py:517: MatplotlibDeprecationWarning: isinstance(
  return not cbook.iterable(value) and (cbook.is_numlike(value) or
```



Out[36]: <ggplot: (-9223371897049515155)>

We see that education & age are by far the most important features according to this model.