Univariate Exploratory Data Analysis

Shizhi Chen-10307389

1. A description of the data, including its origin and quality issues.

2. A description of the analysis methods used and their appropriate-ness to the problem.

3. Results on the data's: (a) Central Tendency; (b) Variability; (c) Overall structure including modes.

4. Appropriate figures and tables to support these results.

5. R or Python code used to produce the analysis.

Background and dataset description

Bike sharing systems are a new generation of traditional bike rentals where the whole process from membership, rental and return back has become automatic. Through these systems, the user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which are composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

I'm going to work with bike sharing dataset I downloaded from the UCI Machine Learning Repository. The origin data source is from Capital Bike-

share Company data system. The dataset contains a daily count of rental bikes of 2012 in Capital Bikeshare system.

The data are collected automatic by the company back-end data system so there are no missing values and the data are reasonably accurate. However, metrology is complicated, even with advanced modern measurement techniques, it can not be expected to be completely accurate. Therefore, I think the quality of this dataset is 3☆.

Preliminary preparation

```
# Load the libraries needed
%matplotlib inline
import numpy as np
import scipy as sp
import scipy.stats as st
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set_style("whitegrid", {'axes.grid' : False})
from statsmodels.distributions.empirical_distribution import ECDF
import statsmodels.api as sm
```

Here we focus on the univariate case – one measurement per 'thing' – so for the bike sharing system data I might be interested in how many times is the shared bike used in a day only. In order to find the information we are interested in from the dataset, we need to visualize the data and study their central tendency, variability and overall structure including modes.

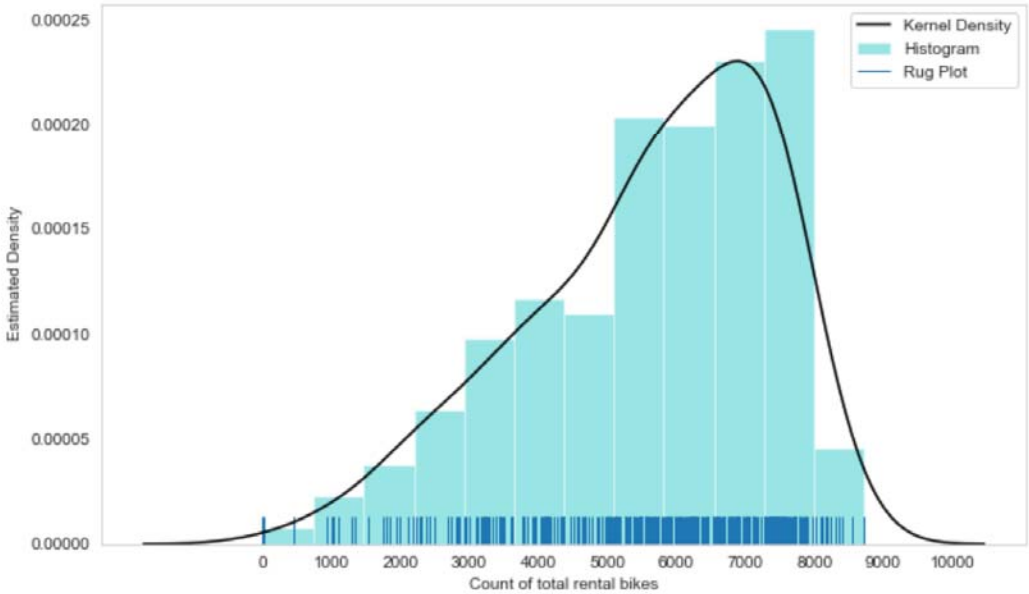
```
# Read in the data as a Pandas frame
f = pd.read_csv('./bike_sharing_day.csv')
#Pull in a univariate dataset consisting of the cnt
x = np.double(f.cnt.values)
```

```
dataset=f[["dteday","cnt"]]
dataset.head()
```

	dteday	cnt
0	2012/1/1	2294
1	2012/1/2	1951
2	2012/1/3	2236
3	2012/1/4	2368
4	2012/1/5	3272

Then I try to use rug plot, histogram and kernel density plots to visualize the data.

```
plt.figure(figsize=(10,6))
ax = sns.distplot(x,rug=True,
                  kde_kws={"label": "Kernel Density", "color" : 'k'},
                  hist_kws={"label": "Histogram","color":'c'})
sns.rugplot(np.array([1]),label="Rug Plot")
plt.xlabel('Count of total rental bikes')
plt.ylabel('Estimated Density')
plt.xticks((0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000))
plt.legend()
plt.show()
```

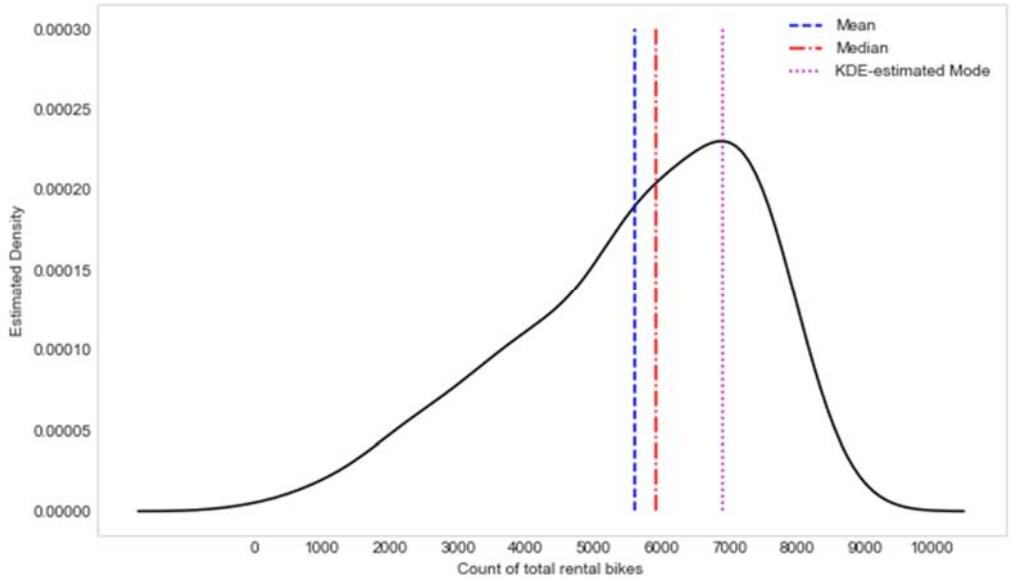


The figure shows that the data is unimodal and suitable to fit the kernel density. Next, we measure the central tendency of the data.

```
xk, yk = ax.get_lines()[0].get_data()
mm = np.mean(x)
md = np.median(x)
mo = xk[np.argmax(yk)]
print(mm, md, mo)
```

```
5599.934426229508 5927.0 6910.006071706808
```

```
plt.figure(figsize=(10,6))
xx = np.ones(2)
yy = np.array([0, 0.0003])
plt.plot(xk, yk, '-k')
plt.plot(mm*xx, yy, '--b', label='Mean')
plt.plot(md*xx, yy, '-.r', label='Median')
plt.plot(mo*xx, yy, ':m', label='KDE-estimated Mode')
plt.xlabel('Count of total rental bikes')
plt.ylabel('Estimated Density')
plt.xticks((0, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000))
plt.legend()
plt.show()
```



From the frequency distribution, we can see the data are left-skewed, and as a consequence of this the mode is largest and the mean is smallest. (Mode>Median>Mean)

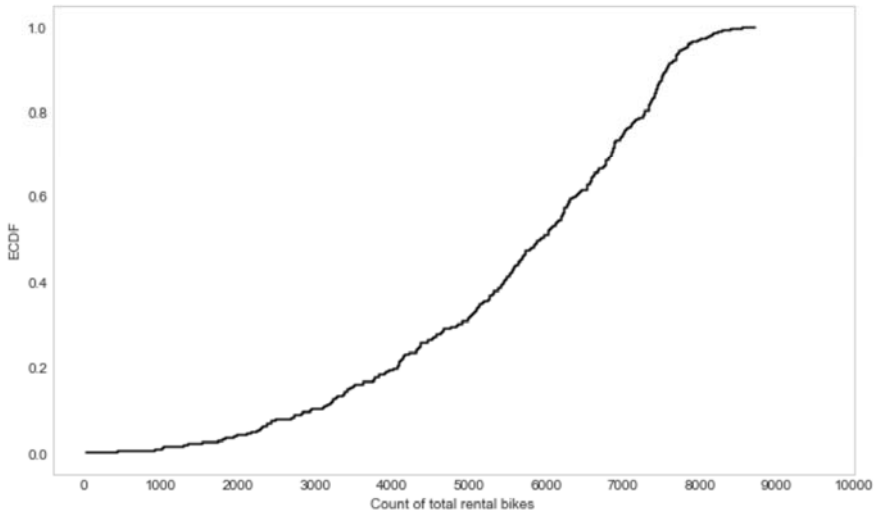
```
print("Mean=", np.mean(x))
print("Unbiased Var:", np.var(x, ddof=1))
ss = np.sqrt(np.var(x))
print("Skewness:", sp.stats.moment(x, 3)/(ss**3))
print("Kurtosis:", (sp.stats.moment(x, 4)/(ss**4)) - 3)
```

```
Mean= 5599.934426229508
Unbiased Var: 3199332.7408937793
Skewness: -0.7015849208668804
Kurtosis: -0.22077357032653122
```

We can also know the data is left-skewed by the negative skewness value of data. The bike-sharing data also has a negative value of excess kurtosis and so it is platykurtic.

ECDF empirical cumulative distribution function

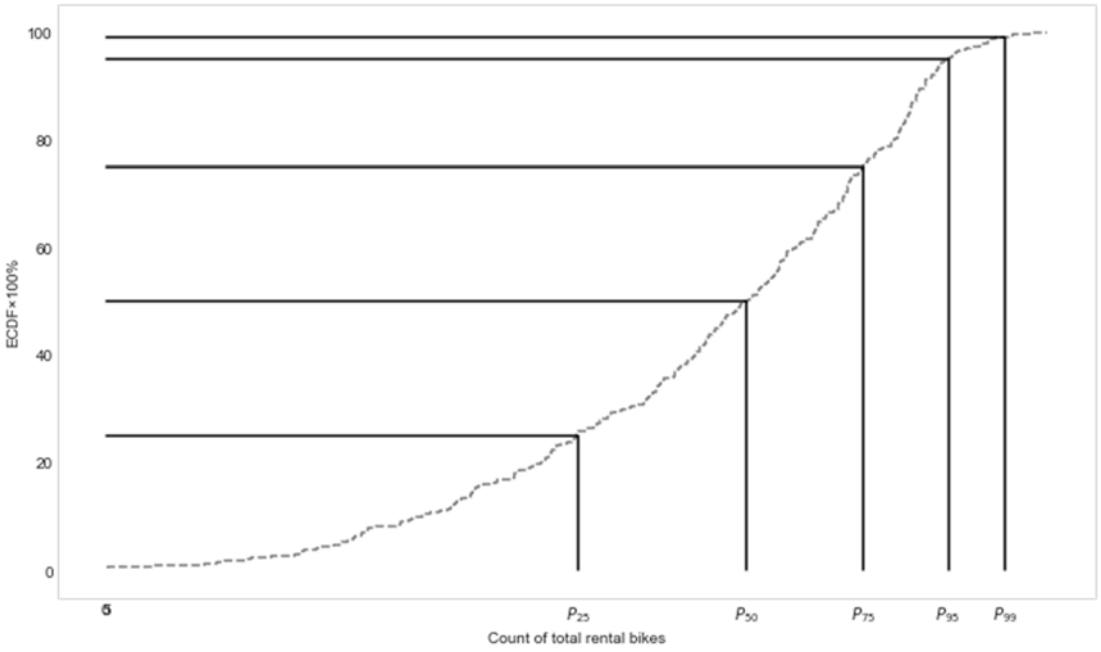
```
ecdf=ECDF(x)
plt.figure(figsize=(10,6))
plt.step(ecdf.x,ecdf.y,c='k')
plt.xlabel('Count of total rental bikes')
plt.ylabel('ECDF')
plt.xticks((0,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000))
plt.show()
```



The plot of this function is a lossless visualisation of the data. From the cumulative distribution figure, we can conclude that the user usage of this bicycle sharing system is constantly increasing.

Order Statistics

```
p25 = np.percentile(x,25)
p50 = np.percentile(x,50)
p75 = np.percentile(x,75)
p95 = np.percentile(x,95)
p99 = np.percentile(x,99)
plt.figure(figsize=(10,6))
plt.step(ecdf.x,100*ecdf.y,linestyle='--',c=[0.5,0.5,0.5])
plt.plot([0,p25,p25],[25, 25, 0],'-k')
plt.plot([0,p50,p50],[50, 50, 0],'-k')
plt.plot([0,p75,p75],[75, 75, 0],'-k')
plt.plot([0,p95,p95],[95, 95, 0],'-k')
plt.plot([0,p99,p99],[99, 99, 0],'-k')
plt.xlabel('Count of total rental bikes')
plt.ylabel('ECDF×100%')
plt.xticks((0,p25,p50,p75,p95,p99,5),('0','$P_{25}$','$P_{50}$','$P_{75}$','$P_{95}$','$P_{99}$','5'))
plt.tight_layout()
plt.show()
```



This figure shows the different quantile of the data, the z-th percentile P_z is the value of x for which z% of the data is ≤ x. So the P_{50} is the median of the data. This is related to the ECDF as shown above. A measure of dispersal of the data is the inter-quartile range. For example, IQR(x) = P_{75} − P_{25}=2642.25.

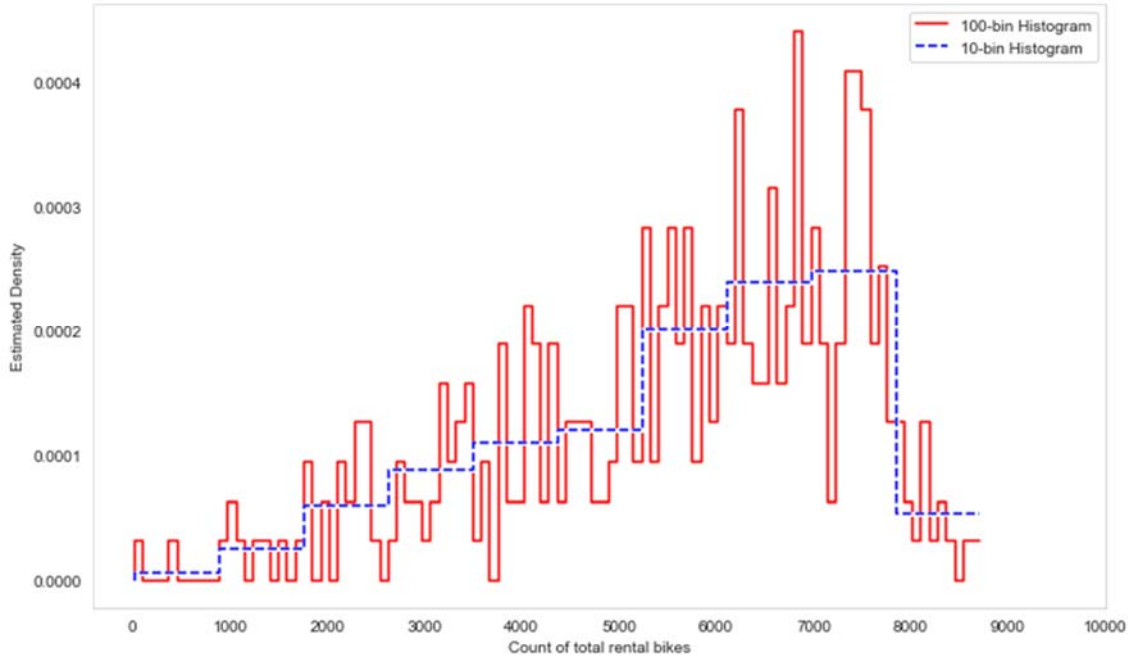
Estimating Population Density – Histograms

The histogram is essentially an estimate of probability density:

$$\hat{f}(\xi|\mathbf{b}) = \frac{1}{n}\sum_{a=1}^{q}\frac{\mathbf{1}_{\{b_a\leq\xi<b_{a+1}\}}}{b_{a+1}-b_a}\sum_{i=1}^{n}\mathbf{1}_{\{b_a\leq x_i<b_{a+1}\}}$$

$$= \sum_{a=1}^{q}\frac{\mathbf{1}_{\{b_a\leq\xi<b_{a+1}\}}}{b_{a+1}-b_a}\left\langle\mathbf{1}_{\{b_a\leq x<b_{a+1}\}}\right\rangle .$$

```
yh10, xh10 = np.histogram(x,10, density=True)
yh100, xh100 = np.histogram(x,100, density=True)
plt.figure(figsize=(10,6))
plt.step(xh100,np.concatenate((np.zeros(1),yh100)),'r',label='100-bin Histogram')
plt.step(xh10,np.concatenate((np.zeros(1),yh10)),'w',linewidth=3)
plt.step(xh10,np.concatenate((np.zeros(1),yh10)),'b',linestyle='dashed',label='10-bin Histogram')
plt.xlabel('Count of total rental bikes')
plt.ylabel('Estimated Density')
plt.xticks((0,1000,2000,3000,4000,5000,6000,7000,8000,9000,10000))
plt.legend()
plt.tight_layout()
plt.show()
```



It can be seen from the histogram that in the whole year of 2012, the highest probability that a shared bicycle is used 6000-8000 times a day.

Estimating Population Density – Kernels

Kernel density methods are an alternative to histograms:

$$\hat{f}(\xi|w) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{K}(\xi - x_i; w) = \langle \mathcal{K}(\xi - x; w)\rangle$$

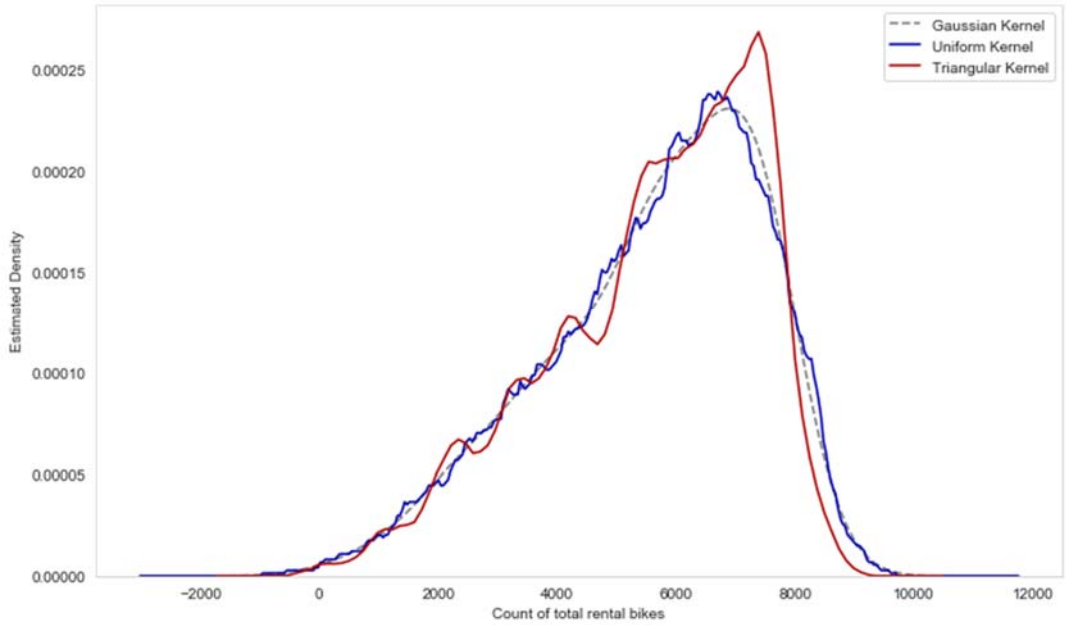
The kernel function $\mathcal{K}$ is a symmetric function with a width, w – which can be tuned in different ways – and which integrates to one. We have 3 different kernels they are Gaussian, Uniform and Triangular.

Fit the data to these models and visualized in a figure:

```
mykde = sm.nonparametric.KDEUnivariate(x)
mykde.fit(kernel="uni", fft=False)
xuni = mykde.support
yuni = mykde.density
mykde2 = sm.nonparametric.KDEUnivariate(x)
mykde2.fit(kernel="tri", fft=False)
xtri = mykde2.support
ytri = mykde2.density
mykde3 = sm.nonparametric.KDEUnivariate(x)
mykde3.fit(kernel="gau")
xgau = mykde3.support
ygau = mykde3.density
```

```
plt.figure(figsize=(10,6))
plt.plot(xgau, ygau, label='Gaussian Kernel',color=[0.5,0.5,0.5],linestyle='--')
plt.plot(xuni, yuni, label='Uniform Kernel',color=[0,0,0.7],linestyle='-')
sns.kdeplot(x,kernel='tri',color=[0.7, 0, 0],label='Triangular Kernel')
plt.xlabel('Count of total rental bikes')
plt.ylabel('Estimated Density')
plt.legend()
plt.tight_layout()
plt.show()
```

As the figure shown below, three model have similar results in fit this dataset and Gaussian Kernel is the most suitable one.



Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics. In this case, the dataset's central tendency, variability and overall structure including modes are analysed by different methods and obtain some interesting figures and conclusion.

Through the exploratory analysis of univariate, the overview of the entire dataset's characteristics is clearer. If we want to further research about this dataset, adding relevant variables to do some regression analysis and fitting more complex models will help to access more useful information.