Question 1. [15 marks]

Consider the matrix$M$ given below.

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 1 & 2 \\ 9 & 5 & 1 \end{bmatrix}$$

```
import numpy as np
D = np.diag([1.29769972, 1.27691898, 0.67262161])
E = np.diag([0.07280195, 0.13820884, 0.14045822])
```

(a) Write Python code to enter the matrix$M$ as a numpy array, print out M, and calculate thesum of each row and column using `numpy.sum()`.*Include both Python code and output in your answer.*

[5 marks]

```
import numpy as np

arr= np.array([1, 2, 3, 5, 1, 2, 9, 5, 1])

a = arr.reshape(3, 3)

print("M=",a,"\n")
x=np. sum(a, dtype=np. int32)
rows = len(a);
cols = len(a[0]);

for i in range(0, rows):
   sumRow = 0;
   for j in range(0, cols):
      sumRow = sumRow + a[i][j];
   print("Sum of " + str(i+1) +" row: " + str(sumRow));

for i in range(0, rows):
   sumCol = 0;
   for j in range(0, cols):
      sumCol = sumCol + a[j][i];
   print("Sum of " + str(i+1) +" column: " + str(sumCol));

print("sum of all rows",x)
print("sum of all columns",x)
```

OUTPUT

```
M= [[1 2 3]
 [5 1 2]
 [9 5 1]]

Sum of 1 row: 6
Sum of 2 row: 8
Sum of 3 row: 15
Sum of 1 column: 15
Sum of 2 column: 8
Sum of 3 column: 6
sum of all rows 29
sum of all columns 29
.
```

(b) Consider the diagonal matrices D and E given in the Python code above. Calculate the matrix product $S = DME$ and print out S. What property do the rows and columns of S have? Hint: S is called a "doubly stochastic" matrix.*Include both Python code and output in your answer.*

[5 marks]

```
import numpy as np
M= ([[1,2,3],[5,1,2],[9,5,1]])
D = np.diag([1.29769972, 1.27691898, 0.67262161])
E = np.diag([0.07280195, 0.13820884, 0.14045822])
S = D@M@E
print("matrix product  S=DME\n")
print(S)
row= np.sum(S,axis = 1)
print('\nsum of rows', row)
column = np.sum(S,axis = 0)
print('\nsum of columns',column)
```

OUTPUT

```
matrix product  S=DME

[[0.09447507 0.35870715 0.54681778]
 [0.46481096 0.17648149 0.35870753]
 [0.44071348 0.46481126 0.09447523]]

sum of rows [0.99999999 0.99999998 0.99999998]

sum of columns [0.99999951 0.9999999  1.00000055]

The matrix product S has the property of stochastic ,the entries of each column sum to 1.
```

(c) Write Python code to build the matrices $A = \frac{1}{2}(M + M^T)$ and $B = \frac{1}{2}(M - M^T)$. Use `numpy.allclose()` to demonstrate that one of A and B is a symmetric matrix, the other one is an antisymmetric matrix, and that $A + B$ gives M. *Include both Python code and output in your answer.*

[5 marks]

```
import numpy as np

M= np.array([[1,2,3],[5,1,2],[9,5,1]])
Mt=(M.transpose())

print ("M Input array:\n", M)
print ("M^T Input array:\n", Mt)

out_arr= np.add(M,Mt)
print ("added (M+M^T) array:\n",out_arr)
out_arr1 = np.subtract(M,Mt)
print ("subtracted (M-M^T) array:\n",out_arr1)
A=(out_arr/2)
print ("\nA=(1/2)(M+M^T)  =\n",A)
B=(out_arr1/2)
print ("\nB=(1/2)(M-M^T)=\n",B)
```

```
print("\n A is symmetric matrix ")
print("\n B is anti-symmetric matrix")
addb = np.add(A,B)
print ("\nAdded (A+B) array:\n", addb)
print('\nA+B is equal to M:', np.allclose(addb,M))
print ("\nM Input array:\n", M)
```

OUTPUT

```
M Input array:
 [[1 2 3]
 [5 1 2]
 [9 5 1]]

M^T Input array:
 [[1 5 9]
 [2 1 5]
 [3 2 1]]

added (M+M^T) array:
 [[ 2  7 12]
 [ 7  2  7]
 [12  7  2]]

subtracted (M-M^T) array:
 [[ 0 -3 -6]
 [ 3  0 -3]
 [ 6  3  0]]

A=(1/2)(M+M^T)  =
 [[1.  3.5 6. ]
 [3.5 1.  3.5]
 [6.  3.5 1. ]]

B=(1/2)(M-M^T)=
 [[ 0.  -1.5 -3. ]
[ 1.5  0.  -1.5]
[ 3.   1.5  0. ]]

 A is symmetric matrix
 B is anti-symmetric matrix

Added (A+B) array:
```

```
 [[1. 2. 3.]
 [5. 1. 2.]
 [9. 5. 1.]]

A+B is equal to M: True

M Input array:
 [[1 2 3]
 [5 1 2]
 [9 5 1]]
```

Question 2. [10 marks]

Consider the *triangular* distribution which has three parameters: L (left), M (middle), and R (right). The Python code given below plots a particular triangular distribution with the given values of L, M and R.

```
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt
L, M, R = (10,20,60)
plt.figure()
x = np.arange(L-5,R+6)
y = stats.triang.pdf(x, c=(M-L)/(R-L), loc=L, scale=R-L)
plt.plot(x,y,'b-')
plt.grid()
plt.show()
```

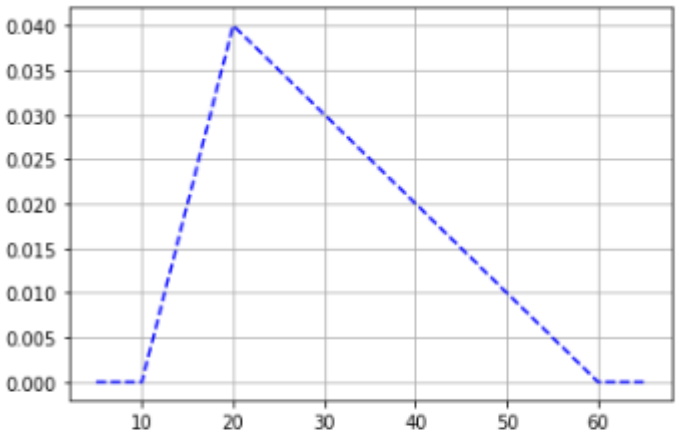
(a) The *mean* of a triangular distribution is given by $(L + M + R)/3$. Calculate the mean, the median, and the upper quartileof the particular triangular distribution above and add them to the plot using vertical dashed lines.*Include both Python code and output in your answer.*

[4 marks]

```
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt
import statistics
import datetime
L,M,R=(10,20,60)
plt.figure()
x = np.arange(L-5,R+6)
y = stats.triang.pdf(x,c=(M-L)/(R-L),loc=L,scale=R-L)
plt.plot(x,y,'b--')
plt.grid()
plt.show()
n = 10000
gfg = [10,20,60]
print("mean=", statistics.mean(gfg))
print("Median=",statistics.median(gfg))
data = [10,20,60]
upper_quartile = np.quantile(data, 0.75)
print("upper_quartile=",upper_quartile)
```

```
mu = mean
sigma = np.std(x)
sample = stats.norm.rvs(size = n,loc = mu,scale = sigma)
print("\n",sample)
plt.figure()
plt.hist(sample,bins=50)
plt.plot(x,n*y/50, 'k-')
plt.show()
```

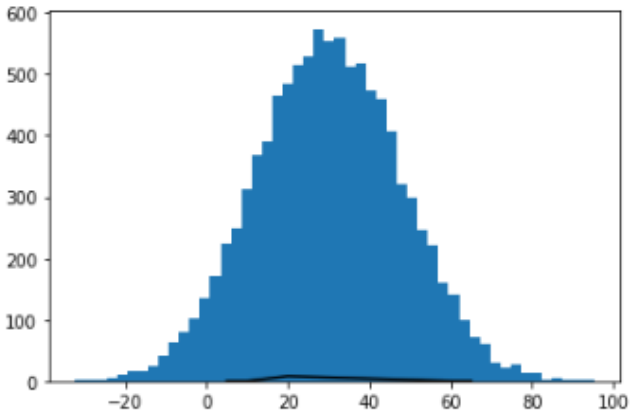
OUTPUT




```
Mean value is= 30.0

 [ 5.2653643  37.51036732 13.3747905  ... 23.0175185  47.79617578
 40.01855722]
```

Median= 20

upper_quartile= 40.0




```
Median and the upper quartile of the particular triangular distribution is described above
```

(b) Generate a random sample of 10000 values from the particular triangular given above. Plot a histogram of your sample using 50 bins. Calculate the mean, median, and upper quartile from your sample and compare to the equivalent values from part (a).*Include both Python code, histogram and output in your answer.*

[6 marks]

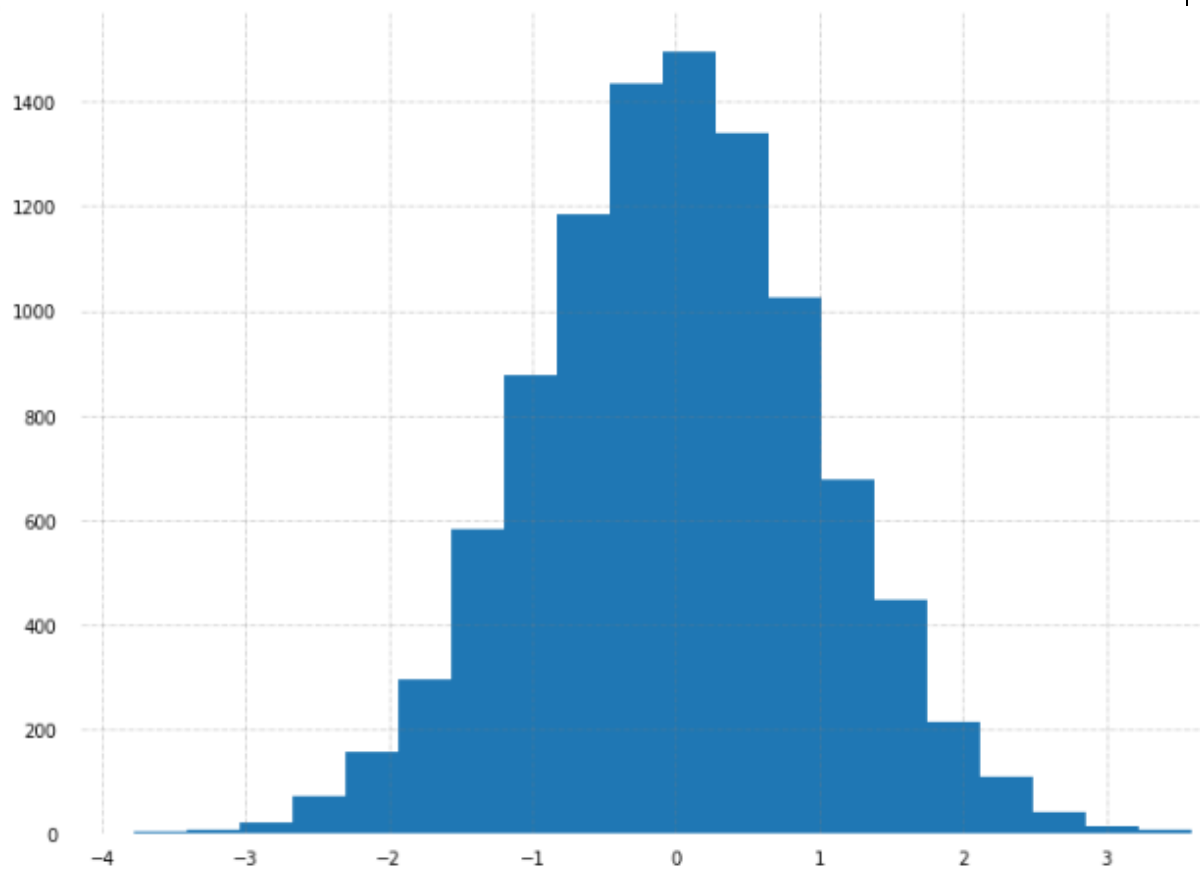
Code

```
import scipy.stats as stats
import numpy as np
import matplotlib.pyplot as plt
import statistics
import datetime
gfg = [5,10,20]
print("mean=", statistics.mean(gfg))
print("Median=",statistics.median(gfg))
data = [10,20,60]
upper_quartile = np.quantile(data, 0.75)
print("upper_quartile=",upper_quartile)
np.random.seed(23685752)
N_points = 10000
n_bins = 20

x = np.random.randn(N_points)
y = .8 ** x + np.random.randn(10000) + 25
legend = ['distribution']

fig, axs = plt.subplots(1, 1,
                        figsize =(10, 7),
                        tight_layout = True)
for s in ['top', 'bottom', 'left', 'right']:
    axs.spines[s].set_visible(False)

axs.xaxis.set_ticks_position('none')
axs.yaxis.set_ticks_position('none')

axs.xaxis.set_tick_params(pad = 5)
axs.yaxis.set_tick_params(pad = 10)

axs.grid(b = True, color ='grey',
         linestyle ='-.', linewidth = 0.5,
         alpha = 0.6)

fig.text(0.9, 0.15, '',
         fontsize = 12,
         color ='red',
         ha ='right',
         va ='bottom',
         alpha = 0.7)
```

```
N, bins, patches = axs.hist(x, bins = n_bins)
fracs = ((N**(1 / 5)) / N.max())
norm = colors.Normalize(fracs.min(), fracs.max())

for thisfrac, thispatch in zip(fracs, patches):
    color = plt.cm.viridis(norm(thisfrac))
    thispatch.set_facecolor(color)
plt.xlabel("X-axis")
plt.ylabel("y-axis")
plt.legend(legend)
plt.title('Customized histogram')
```

OUTPUT

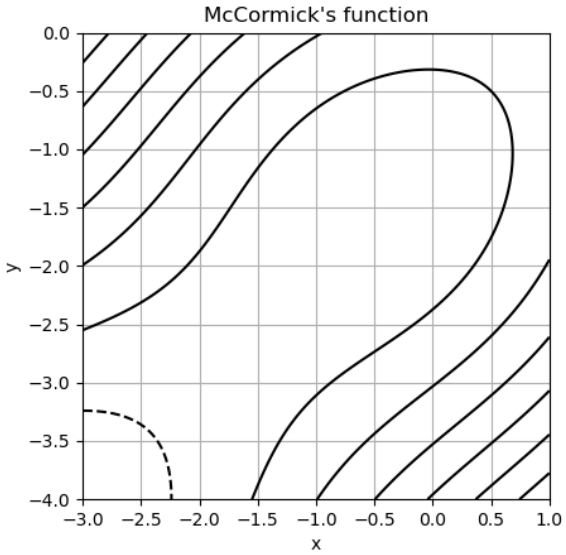
```
Median= 10
upper_quartile= 40.0
mean= 17.0
```



Question 3. [10 marks]

Consider *McCormick's* function and the corresponding contour plot given below.

$$z = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1$$



(a) Briefly explain what is mean by a "contour" on a contour plot, i.e., what does each individual contour represent?

[2 marks]

3. A) McCormick's function is depicted on the chart, which has two axes, x and y. A lobed graph is usually required to depict the relationship between three variables. A contour plot is a two-dimensional figure where the third axis is represented by circles (typically coloured). As in third axis, each and every point mostly on circle will have the same result. To get the value of z, the two parameters are frequently joined, and lines are drawn to link the (x,y) Coordinate where all that value obtained appears. t. The contours chart, often known as contours, is a geometric method of representing a tri surface on the double surface by applying constant z slices. In line chart, the very first button adds a label, whereas the second click (or both mouse movements at once) finishes generating labels. The final button would be used to delete the most recent label, but only when the markings are still not inline.

(b) Write Python code to reproduce the contour plot of McCormick's function given above (this plot uses only the default levels). You may find the function `numpy.sin()` helpful. *Include only your Python codein your answer.*

[4 marks]

CODE

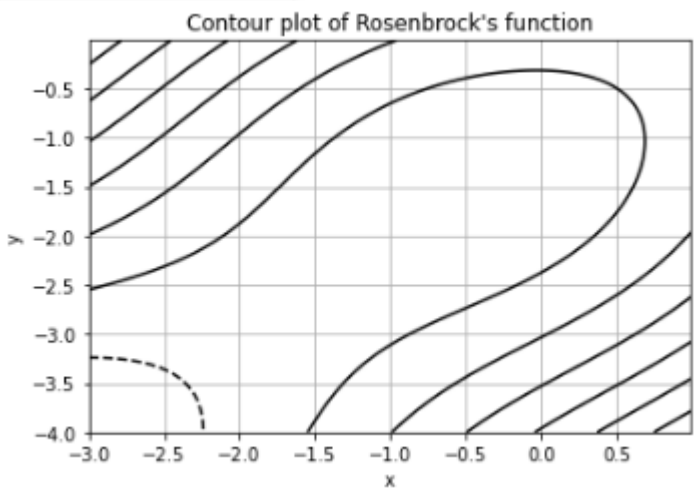
```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(-3,1,0.01)
y = np.arange(-4,0,0.01)
X,Y = np.meshgrid(x,y)
Z = np.sin(X+Y) + (X-Y)**2 - 1.5*X + 2.5*Y + 1
print(Z)
print("minimum of z",np.min(Z))
print("maximum of Z",np.max(Z))
plt.figure()
L1 = plt.contour(X,Y,Z,colors='black')
plt.grid()
plt.title("Contour plot of Rosenbrock's function")
plt.xlabel('x')
plt.ylabel('y')
plt.xlim([-3.0,1.0,0.5])
plt.ylim([-4.0,0.0,0.5])
plt.axis('equal')
plt.show()
```

Output

```
[[-4.1569866  -4.14431485 -4.13137817 ... 14.13453881 14.20910674
  14.28388681]
 [-4.14431485 -4.13177817 -4.1189773  ... 14.05030674 14.12468681
  14.19927999]
 [-4.13137817 -4.1189773  -4.10631303 ... 13.96628681 14.04047999
  14.11488729]
 ...
 [14.13453881 14.05030674 13.96628681 ...  1.2775581   1.2885155
   1.29959157]
 [14.20910674 14.12468681 14.04047999 ...  1.2885155   1.29919157
   1.30998571]
 [14.28388681 14.19927999 14.11488729 ...  1.29959157  1.30998571
   1.32049737]]
```

minimum of z -4.156986598718789

maximum of Z 14.283886807849182



(c) Add further Python code to label the contours and add further contours that clearly highlight the valley bottom near the point $(-0.5, -1.5)$ and investigate the behaviour of the function near the point$(-1.5, -2.5)$.

[4 marks]

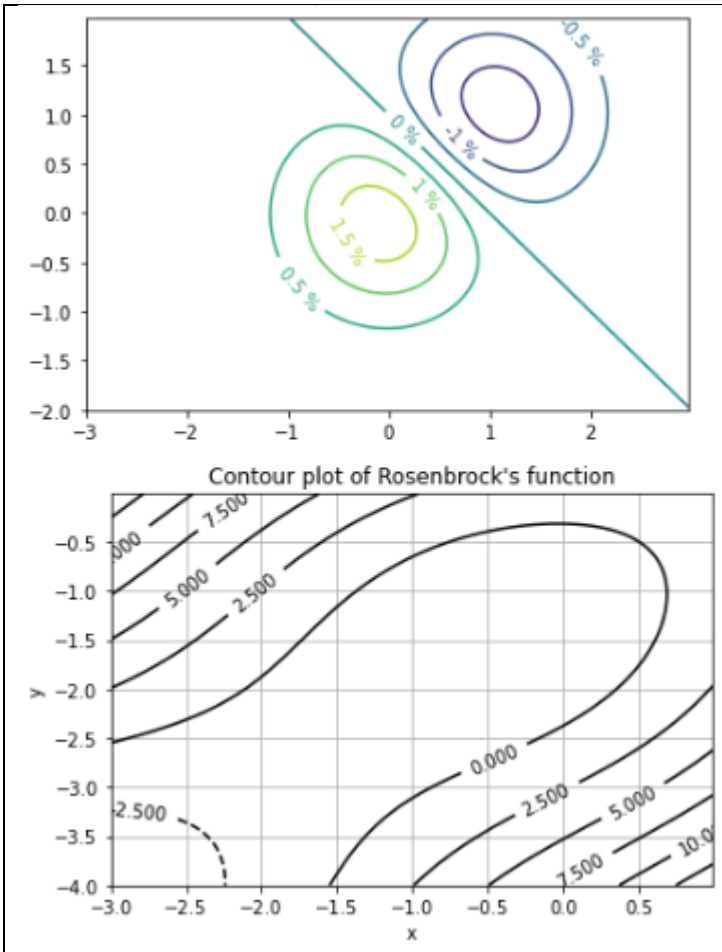
Code

```
import numpy as np
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt
delta = 0.025
x = np.arange(-3.0, 3.0, delta)
y = np.arange(-2.0, 2.0, delta)
X, Y = np.meshgrid(x, y)
Z1 = np.exp(-X**2 - Y**2)
Z2 = np.exp(-(X - 1)**2 - (Y - 1)**2)
Z = (Z1 - Z2) * 2
def fmt(x):
    s = f"{x:.1f}"
    if s.endswith("0"):
        s = f"{x:.0f}"
    return rf"{s} \%" if plt.rcParams["text.usetex"] else f"{s} %"
fig, ax = plt.subplots()
CS = ax.contour(X, Y, Z)

ax.clabel(CS, CS.levels, inline=True, fmt=fmt, fontsize=10)
x = np.arange(-3,1,0.01)
```

```
y = np.arange(-4,0,0.01)
X,Y = np.meshgrid(x,y)
Z = np.sin(X+Y) + (X-Y)**2 - 1.5*X + 2.5*Y + 1
print(Z)
plt.figure()
Lab = plt.contour(X,Y,Z,colors='black')
plt.clabel(Lab)
plt.grid()
plt.title("Contour plot of Rosenbrock's function")
plt.xlabel('x')
plt.ylabel('y')
plt.xlim([-0.5,1.0,1.5])
plt.ylim([-0.5,0.0,1.5])
plt.axis('equal')
plt.show()
```

OUTPUT

```
[[-4.1569866  -4.14431485 -4.13137817 ... 14.13453881 14.20910674
  14.28388681]
 [-4.14431485 -4.13177817 -4.1189773  ... 14.05030674 14.12468681
  14.19927999]
 [-4.13137817 -4.1189773  -4.10631303 ... 13.96628681 14.04047999
  14.11488729]
 ...
 [14.13453881 14.05030674 13.96628681 ...  1.2775581   1.2885155
   1.29959157]
 [14.20910674 14.12468681 14.04047999 ...  1.2885155   1.29919157
   1.30998571]
 [14.28388681 14.19927999 14.11488729 ...  1.29959157  1.30998571
   1.32049737]]
```


-0.5 %
0 %
-1 %
1 %
1.5 %
0.5 %
Contour plot of Rosenbrock's function
7.500
5.000
2.500
0.000
2.500
5.000
7.500
10.0
2.500
x
y

Question 4. [20 marks]

Consider the dataset in the file "animals.csv" (provided along with these questions on Moodle) is extract from the Living Planet Index database https://www.livingplanetindex.org/.Each row of the dataset corresponds to a published survey of one particular species of animal.

In RStudio, make sure you go to the Session menu, select Set Working Directory and then Source File Location. Save the "animals.csv" file to the same folder as where your R code is saved.

```
library(tidyverse)
animals = read_csv('animals.csv')
colnames(animals)
```

(a) The variable (column) *System* gives the ecosystem that each species of animal was found in. Write R code to produce a summary table that gives a count of the number of rows of the dataset corresponding each ecosystem. *Include both R code and output in your answer.*

[3 marks]

Write your answer in this box.
CODE

```
library(readxl)
animals <- read_excel("D:/User/Desktop/animals.xlsx")
view(animals)
animals %>%
group_by(System) %>%
summarise(count=n())
```

Output

```
# A tibble: 3 x 2
  System      count
<chr><int>
1 Freshwater   3150
2 Marine       3943
3 Terrestrial  4542
```

From the output we are grouping the animal under system such as freshwater, marine and terrestrial. The total count of species in fresh water is 3150, total count of species in marine is 3943 and total count of species in 4542.

(b) Write R code using a dplyr pipe to find each *Country* in which a Hippopotamus has been found. Make sure that each country appears at most once. *Include both R code and list (or table) of countries in your answer.*

[5 marks]

Write your answer in this box.

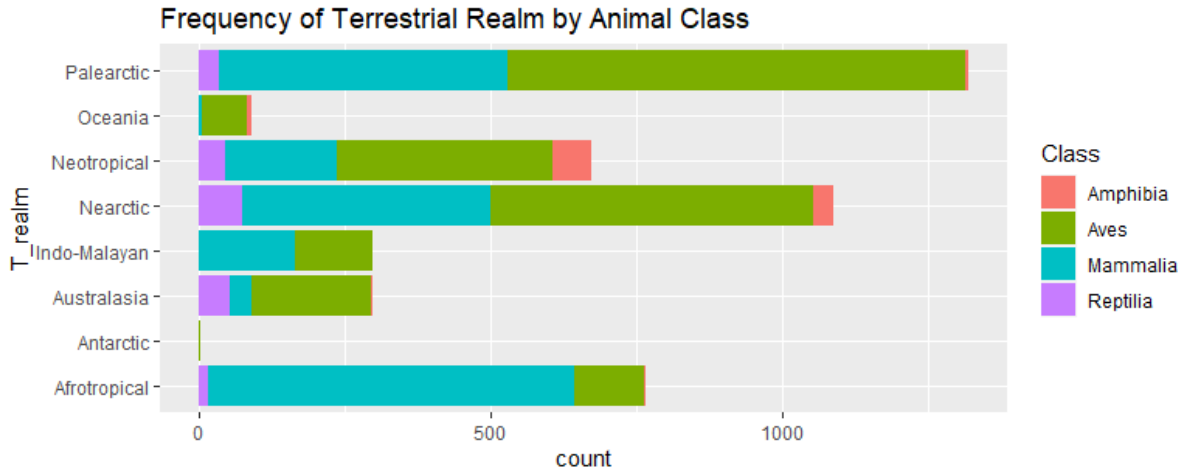
Code

```
colnames(animals)
animals%>%
filter(Genus=='Hippopotamus')%>%
select(Country)%>%
distinct(Country)
```

Output:

```
  Country
<chr>
1 Tanzania, United Republic Of
2 Uganda
3 Congo, The Democratic Republic Of The
4 Malawi
5 Kenya
6 Zimbabwe
7 Zambia
8 Mozambique
```

(c) Write R code to reproduce the graphical plot below as accurately as possible.*Include both R code and the plot that your code produces in your answer.*



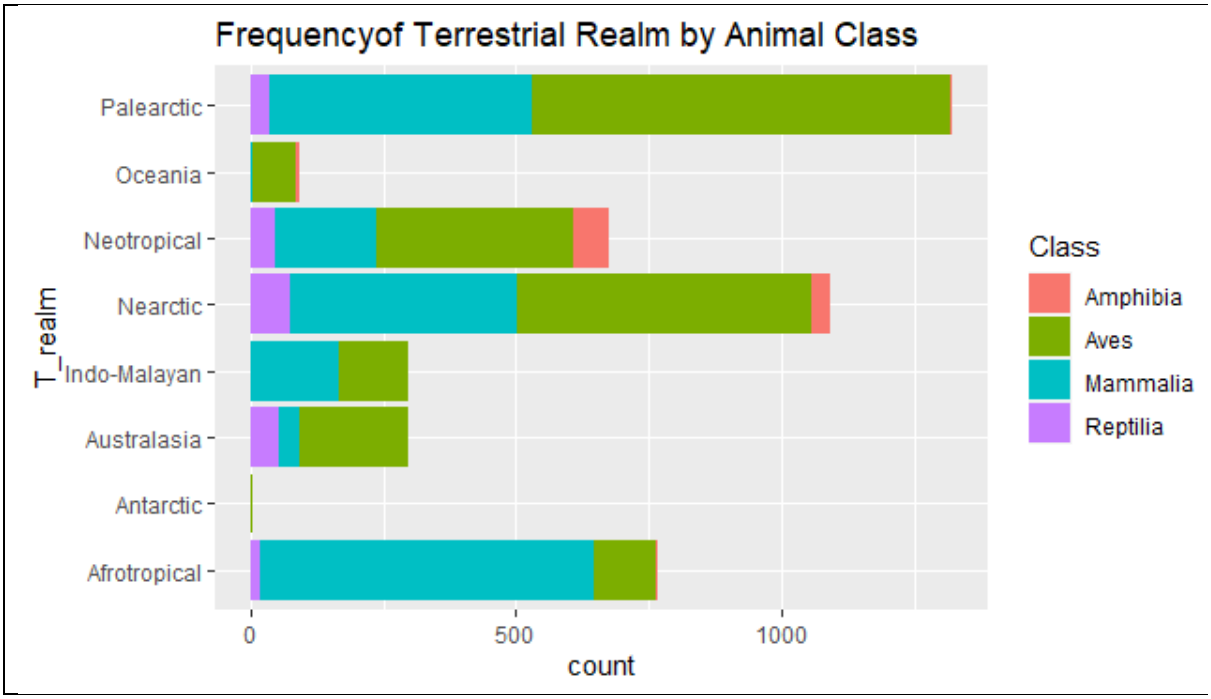
[6 marks]

Write your answer in this box.

Code

```
library((GGally))

Trealmcolumn=animals%>%
group_by(T_realm,Class)%>%
  summarise(count=n()) %>%
  filter(T_realm!="NULL")

ggplot(Trealmcolumn,aes(x=count,y=T_realm,fill=Class))+
geom_col()+
ggtitle('Frequencyof Terrestrial Realm by Animal Class')
```

Output



(d) Consider the Cheloniamydas (known by its common name of "Green turtle"). Use the geographical coordinates *Latitude* (degrees North-South) and *Longitude* (degrees East-West) to plot the locations where Green turtles have been found. Use the categorical variable *Region* to colour each point (together with a legend). What is the name of the island that is the southern-most *Location* where Green turtles have been found? *Include both R code and the output in your answer.*

[6 marks]

Write your answer in this box.

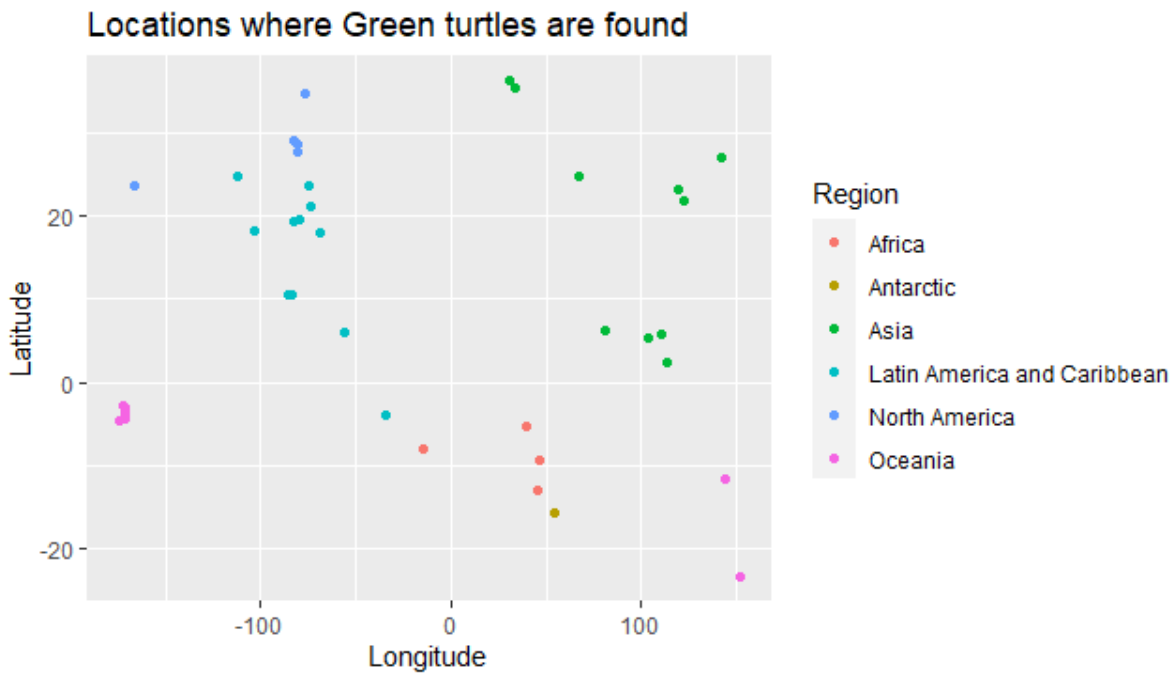
Code

```
data=animals%>%
filter(Common_name=='Green turtle')%>%
select(Latitude,Longitude,Region)

ggplot(animals,aes(y=Latitude,x=Longitude,color=Region))+
geom_point()+
ggtitle("Green turtles location")


land=animals%>%
filter(Common_name=="Green turtle"&Region=="Oceania")%>%
select(Latitude,Longitude,Region,Location)

land%>%
```

```
filter(Longitude==max(Longitude))%>%
select(Location)
```

Output:

```
# A tibble: 1 x 1
  Location
<chr>
1 Heron Island in the southern Great Barrier Reef region, Australia
```



Question 5. [25 marks]

Consider the dataset "bull_sales.csv" (provided along with these questions on Moodle)which gives the sale price (the variable *sale_price*) of bulls sold at auction and the *breed* of bull: Angus (a), Hereford (h), or Simmental (s). Several further measurements are given for each bull, including height at one year old (*yearling_height* in inches), percentage of fat free body weight (*fat_free_body*), amount of back fat (*back_fat* in inches), height when sold (*sale_height* in inches) and weight when sold (*sale_weight* in pounds).

Angus bull – image from https://www.aberdeen-angus.co.uk/sire-verification/

In RStudio, make sure you go to the Session menu, select Set Working Directory and then Source File Location. Save the "bull_sales.csv" file to the same folder as where your R code is saved.

```
library(tidyverse)
bulls = read_csv('bull_sales.csv')
colnames(bulls)
library(GGally)
ggpairs(bulls, aes(colour=breed))
```

(a) The R code above produces a scatter matrix. Comment on the relationship between *sale_price* and *yearling_height* and the relationship between *sale_height* and*back_fat*.

[4 marks]

Write your answer in this box.

Code

```
library(GGally) #  importing gggally
bulls = read_csv('bull_sales.csv') # reading the csv dataset
cor(bull_sales$sale_price,bull_sales$yearling_height)
#correlation between sale_price and yearling_height
cor(bull_sales$sale_height,bull_sales$back_fat)
```

#correlation between sale_height and the back_fat.

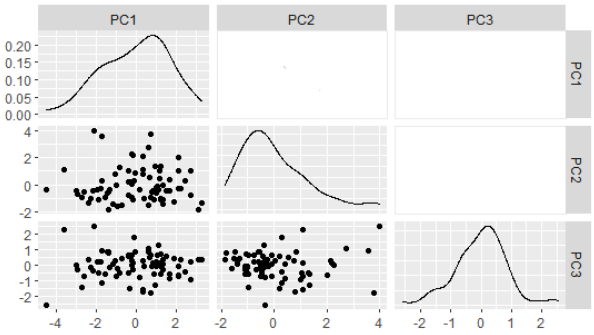
Output

```
>cor(bull_sales$sale_price,bull_sales$yearling_height)
[1] 0.4231607
>cor(bull_sales$sale_height, bull_sales$back_fat)
[1] -0.2820899
```

Comments

- From the above results found the correlation value of relationship between sales_price and yearling_height is positive.
- The correlation value of relationship between sales_height and back_fat is negative.
- so, first relationship get highest correlation.

(b) Suppose the dataset is scaled and Principal Component Analysis (PCA) is applied. Explain why the following scatter matrix is (mostly) reassuring and fill in the missing parts of the scatter matrix.

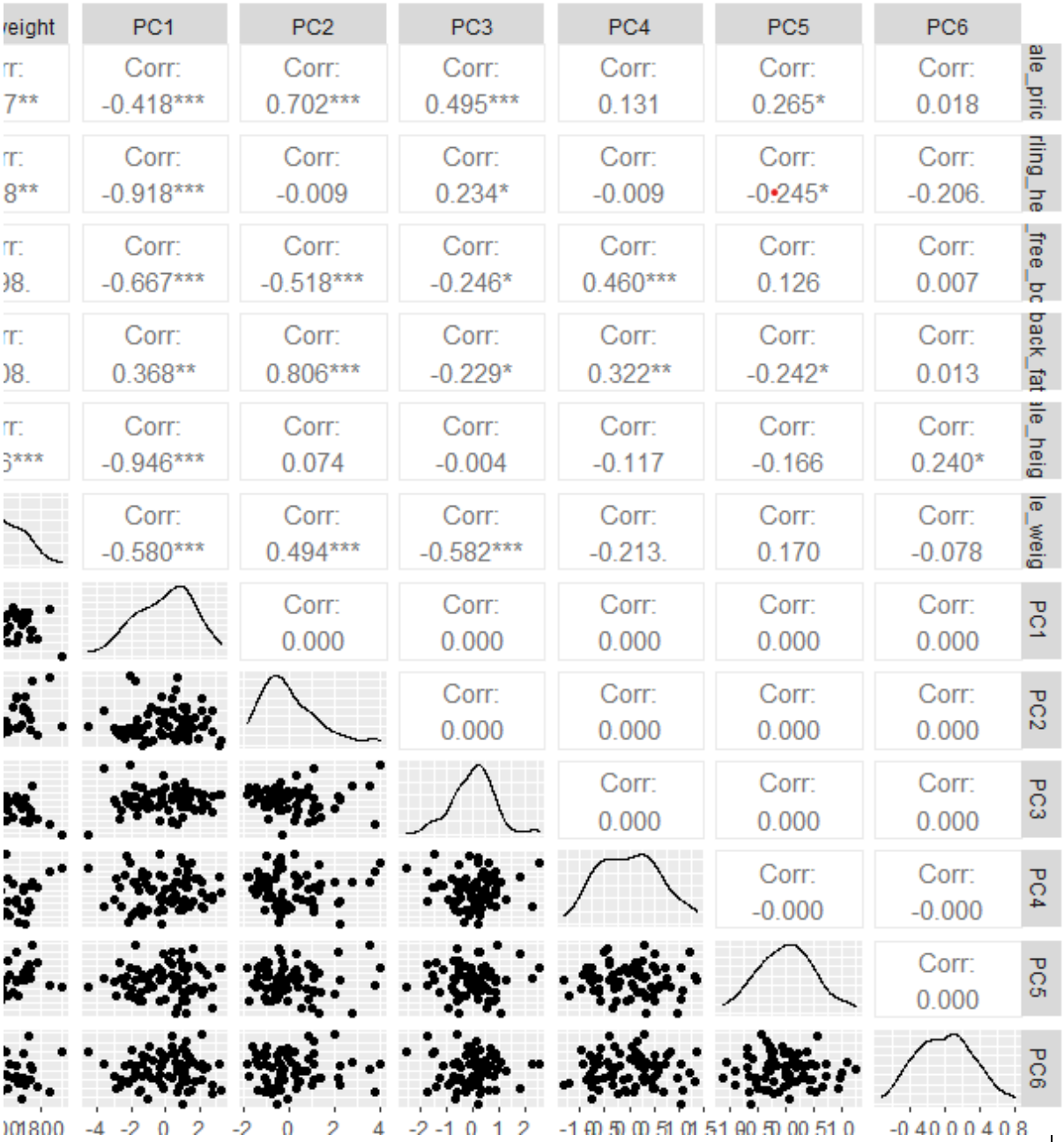


[4 marks]

Write your answer in this box.
Code

```
library(readxl)
library(GGally)
library(ggfortify)
bull_sales<- read_excel("D:/User/Desktop/bull_sales.xlsx")
dataframe = select(bull_sales,breed)
data = prcomp(dataframe,scale=TRUE)
ggpairs(data)
```

Output



Comments

The scatter matrix is gives the complete correlation matrix which is mostly reassuring because the correlation between the principle component analysis is zero. The significance is less compared to the other models. So the scatter matrix is mostly reassuring and changes the values according to the matrix. The missing values in the given correlation are pc1, pc2 and pc3. *The correlation between PC1 and PC2 is 0. The correlation between PC1 and PC3 is 0. The correlation between PC2 and PC3 is 0. The highest correlation is between the variables pc1 and back fat is 0.809. The lowest correlation value is between the variables pc6 and free_bc is 0.007.*

(c) Scaling the dataset and applying PCA gives the partial scatter matrix showing the correlations between the first three principal components (PC1, PC2, and PC3) and the sixoriginal quantitative variables.

sale_price	earling_heigh	fat_free_body	back_fat	sale_height	sale_weight	
Corr: -0.418***	Corr: -0.918***	Corr: -0.667***	Corr: 0.368**	Corr: -0.946***	Corr: -0.580***	PC1
Corr: 0.702***	Corr: -0.009	Corr: -0.518***	Corr: 0.806***	Corr: 0.074	Corr: 0.494***	PC2
Corr: 0.495***	Corr: 0.234*	Corr: -0.246*	Corr: -0.229*	Corr: -0.004	Corr: -0.582***	PC3

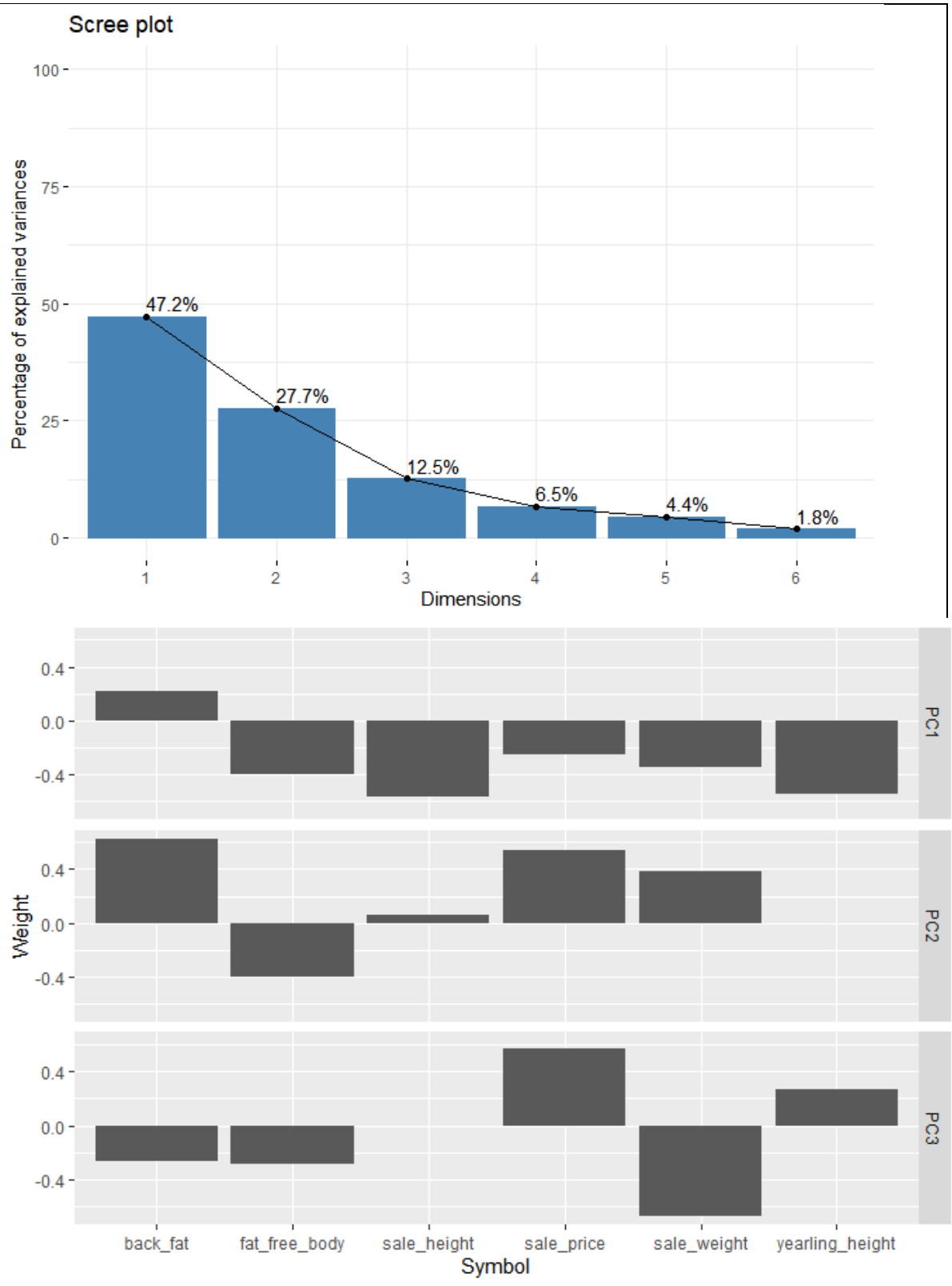
Use R to produce a PCA loadings plot showingPC1, PC2 and PC3. By considering the loadings plot, the variance explained by each principal component, and the partial scatter matrix above, give one possibleinterpretation of each of PC1, PC2 and PC3.Justify whether {PC1, PC2, PC3} are sufficient*Include your R code and output in your answer.*

[9 marks]

Code

```
library("factoextra")
library("FactoMineR")
fviz_screeplot(data, addlabels = TRUE, ylim = c(0, 100))

loadings = as.data.frame(data$rotation[,1:3])
loadings$Symbol = row.names(loadings)
loadings = gather(loadings, key='Component', value='Weight', -Symbol)
ggplot(loadings, aes(x=Symbol,y=Weight)) +
  geom_bar(stat='identity') +
  facet_grid(Component~.)

ggplot(data=dataframe) +
  geom_bar(aes(x=Symbol, y=Weight), stat="identity") +
  facet_grid(Component)
```

Output



Reason

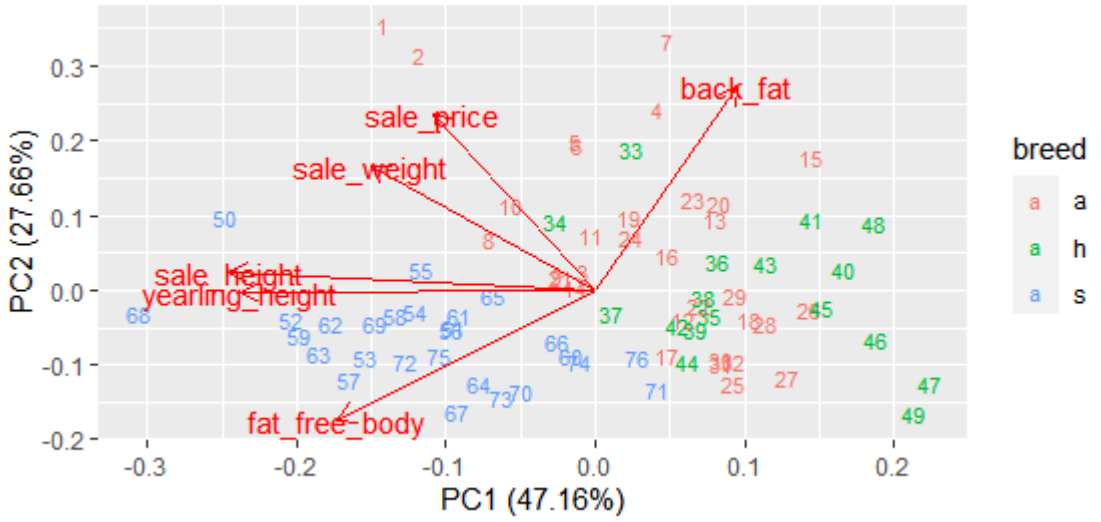
From the scree plot shows the variance present in each principal component analysis. The variance value range to 0.80 and values of PC1,PC2 and PC3 are present in second graph.

The PC1 value is -0.4 to 0, PC2 value is 0 to 0.4 and PC3 value is -0.4 to 0.4 the variance is explained in the screeplot whereas the pc1 and pc2 and pc3 is given for each of the variables in the dataset.

Comments

- By looking at the scree plot:
- Scree plot shows the variance present in each principle component
- 87.4% of information present in these three PC1,PC2,PC3.
- so it is PC1,PC2,PC3 sufficient.
- PC1:Back_fat is negatively correlated with other elements
- PC2:here,yearling_height and sale_height have lesser contribution compared to other elements
- PC3:sale_weight has higher contribution among all
- whereas sale_height has lesser contribution.

(d) Consider the PCA biplotgiven below. Use R to produce an appropriatedendrogram giving a hierarchical clustering of individual bulls. Compare the clusters formed in the dendrogram(using five clusters) with the clusters of individual bullsvisible in the PCA biplot(consider both location of clusters and mixing of breeds).*Include any R code and dendrogram you produce in your answer.*

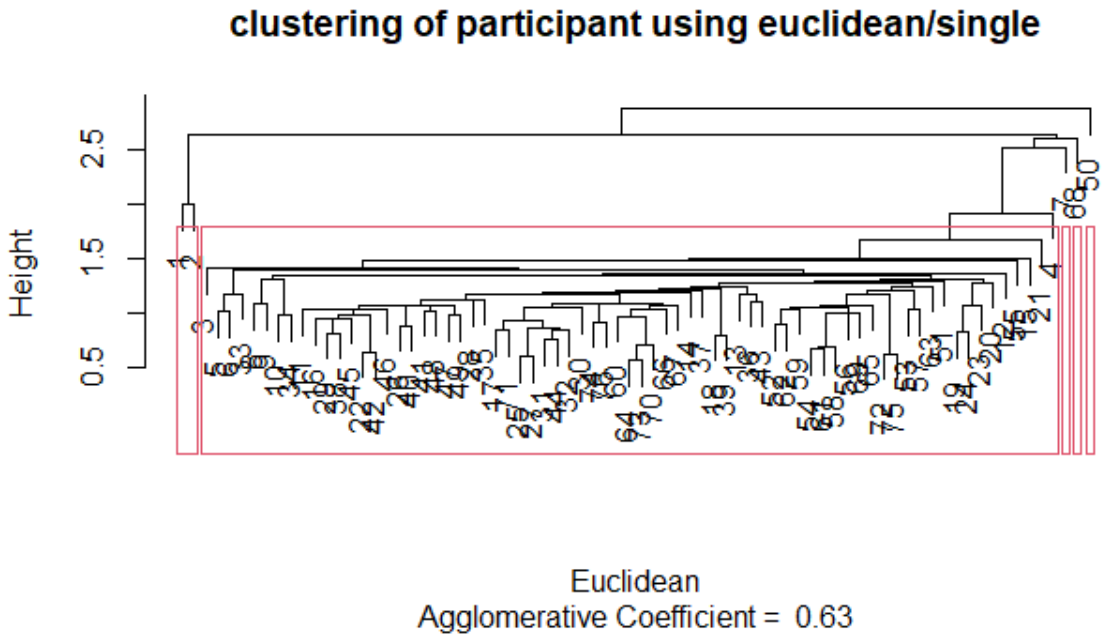


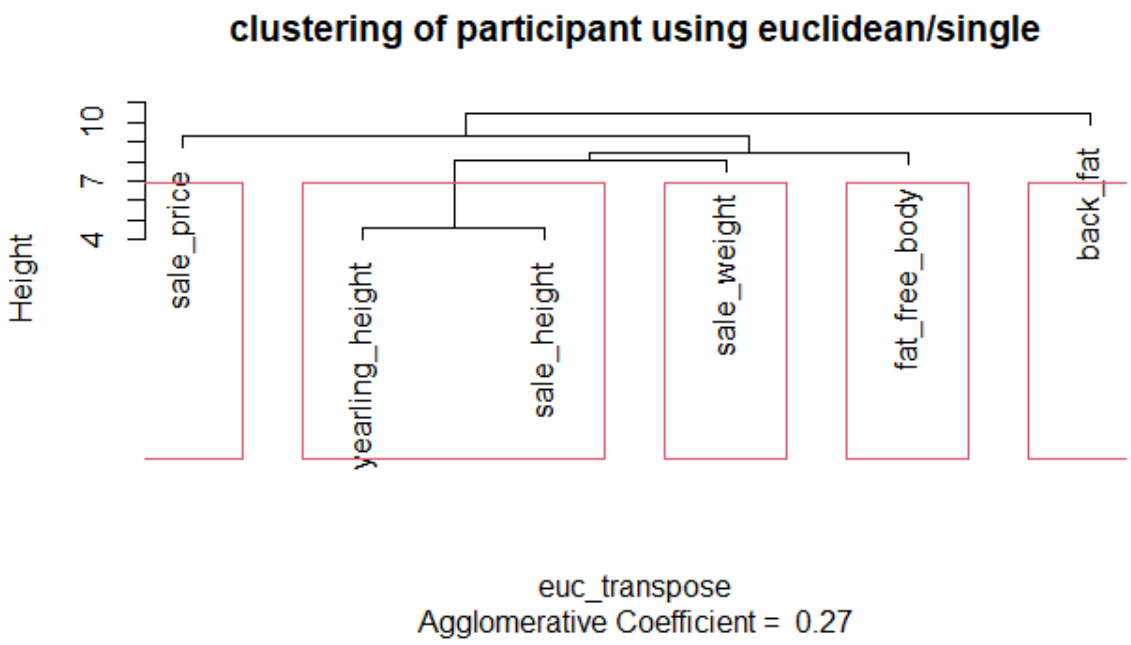
[8 marks]

Write your answer in this box.
Code

```
library(readxl)
library(cluster)
bull_sales<- read_excel("D:/User/Desktop/bull_sales.xlsx")
Euclidean=dist(scale(select(bull_sales,-breed)),method='euclidean')
 # the distance for the scale is been calculated
cluster_results=agnes(Euclidean,method='single')
 # dendogram with cluster
plot(cluster_results,which.plots = 2,
     main="clustering of participant using euclidean/single")
rect.hclust(cluster_results,k=5,border=2)
# rectangular cluster
euclidean_transpose=dist(t(scale(select(bull_sales,-breed))),method='euclidean')
cluster_results=agnes(euclidean_transpose,method='single')
plot(cluster_results,which.plots = 2,
     main="clustering of participant using euclidean/single")
rect.hclust(cluster_results,k=5,border=2)
# dendogram for the clusters.
```

Output





Justification

1. By comparing biplot and cluster plot using Euclidean.
2. I observed that yearling_height and sale_height have strong correlation in both plots.
3. Back_fat and sale_price have no correlation.
4. These 1,2,7,50,68 data are less relatable among other data i.e., it has higher distance from other data.

Question 6. [20 marks]

This question uses the same "bull_sales.csv" dataset as Question 5.

```
library(tidyverse)
bulls = read_csv('bull_sales.csv')
colnames(bulls)
```

(a) Write R code to fit the linear model "sale_price~yearling_height" to the dataset and construct an appropriate scatterplot including the line of best fit. Write down the equation of the fitted model. *Include your R code and scatterplot in your answer.*

[6 marks]

Write your answer in this box.

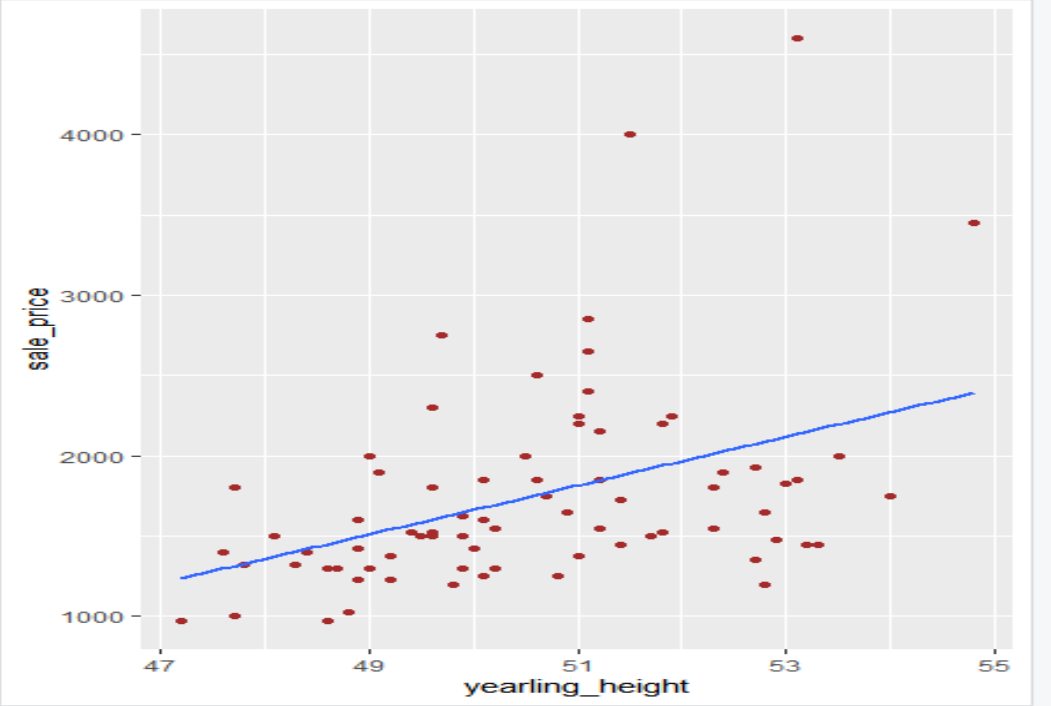
Code

```
library(tidyverse)
library(readxl)
bull_sales <- read_excel("D:/User/Desktop/bull_sales.xlsx")
library(ggfortify)
model=lm(sale_price~yearling_height,data=bull_sales)
summary(model)
ggplot(bull_sales,aes(x=yearling_height,y=sale_price))+
geom_point(colour='brown')+
geom_smooth(method = 'lm',se=FALSE)
```

Output

```
Call:
lm(formula = sale_price ~ yearling_height, data = bull_sales)

Residuals:
   Min     1Q Median     3Q    Max
-889.2 -324.0 -135.8  178.4 2465.1

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     -5950.0     1915.8  -3.106  0.00269 **


yearling_height    152.3       37.9   4.018  0.00014 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 568.3 on 74 degrees of freedom
Multiple R-squared: 0.1791, Adjusted R-squared: 0.168
F-statistic: 16.14 on 1 and 74 DF, p-value: 0.0001398



1. From using the given dataset, we find the linear regression model using linear model function(lm) for fit the linear model.
2. From the result we got equation *yearling_height=(-5950+152.3*sale_price)*
3. we got the R^2 value in multiple R-square is 0.1791 and P-value is 0.0001398.
4. we got the scatterplot of linear model using gglorify library and geom point function.

(b) Assess whether the linear model from part (a) is a good fit to the dataset by looking carefully at the diagnostic plots. *Include your comments only. You may find the R code below useful.*

```
library(ggfortify)
autoplot(model, label.n=5)
autoplot(model, data=bulls, colour='breed')
```

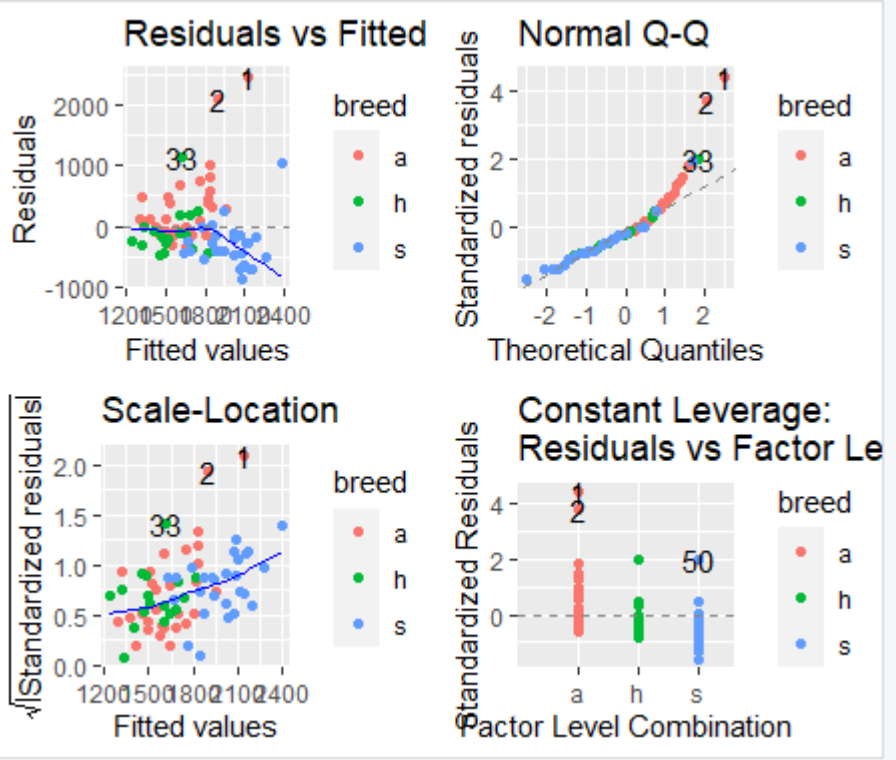
[6 marks]

Write your answer in this box.

Code

```
library(ggfortify)
autoplot(model, label.n=5)
autoplot(model, data=bull_sales, colour='breed')
```

Output



1. *1,2,33 are outliers which need further investigations.*
2. *If we remove data 1,2,50 then it has a huge impact in the Linear Regression and we will get different plot which gives more best fit model.*

3.We got the residual vs fitted model using the library ggfortify and autoplot function is used to plot the graph.

4.In fitted values graph a value got above 2000 but s value is on -1000.

5.In theoretical quantities graph theoretical quantities are increased randomly to 4

6.In this graph h value got more standardised residuals.

7.In fitted value vs residual graph all the points are lies in regression line which represent a good fit.

8.The squareroot of standardised residual got maximum value of 2.0

9.In factor level combination a got maximum value.

10.In factor 1 got the maximum residual value of 4.

(c) Write R code to fit the linear model "sale_price~yearling_height+breed" to the dataset and clearly interpret the fitted model.*Include your R code in your answer.*

[5 marks]

Write your answer in this box.

Code

```
library(readxl)
bull_sales <- read_excel("D:/User/Desktop/bull_sales.xlsx")
model=lm(sale_price~yearling_height+breed,data=bull_sales)
summary(model)
```

Output

```
Min      1Q  Median      3Q     Max
-685.82 -312.63  -51.19  196.02 1635.63

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    -13698.97    2190.58  -6.254 2.54e-08 ***
yearling_height    313.81      43.88   7.152 5.80e-10 ***
breedh          -283.09     141.64  -1.999   0.0494 *
breeds          -984.40     155.45  -6.333 1.83e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 459.5 on 72 degrees of freedom
Multiple R-squared:  0.4777,  Adjusted R-squared:  0.4559
F-statistic: 21.95 on 3 and 72 DF,  p-value: 3.367e-10
```

Result

#R-squared: 0.4777

#sale_price=-13698.97+313.81 yearling_height + (-283.09(*1 if breedh)) + (-984.40(*1 if breeds))+0*

#breedh is less significant

#p-value: 3.367e-10

#R-squared: 0.4777

(d) Would you recommend adding *sale_weight* to the model from part (c) as an additional predictor? *Justify your answer.Include any R code in your answer.*

[3 marks]

Write your answer in this box.

Code

```
library(readxl)
library(olsrr)
bull_sales <- read_excel("D:/User/Desktop/bull_sales.xlsx")
model=lm(sale_price~.,data=bull_sales)
ols_step_best_subset(model)
summary(model)
AIC(model)
```

output

```
Best Subsets Regression
-------------------------------------------------------------------------------
Model Index    Predictors
-------------------------------------------------------------------------------
    1        yearling_height
    2        breed yearling_height
    3        breed yearling_height fat_free_body
    4        breed yearling_height fat_free_body sale_height
    5        breed yearling_height fat_free_body back_fat sale_height
    6        breed yearling_height fat_free_body back_fat sale_height sale_weight
```

ols_step_best_subset(model)

1.I observed that fat_free_body is a additional predictor
2.AIC:1148.2013,R square : 0.5247

From the code we got R_square value for each predicitors are explained below

a)For first model = 0.1791 for saleprice

b) For second model =0.4777 for saleprice

c) For third model =0.5247 for saleprice

d) For fourth model =0.5537 for saleprice

e) For fifth model =0.5606 for saleprice

f) For sixth model =0.5607 for saleprice

From the output we got fat free body is a predictor in the indexes 3,4,5,6. we find fat free body is an additional predictor