Question 1

a. Read in the nail_fungal.csv data and call it nail_fungal. Convert all variables that need to be a factor into a factor. show R code.

```
nail_fungal = read.csv("nail_fungal.csv")
nail_fungal$person <- as.factor(nail_fungal$person)
str(nail_fungal)
```

```
## 'data.frame':    288 obs. of  5 variables:
##  $ observation     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ person          : Factor w/ 32 levels "1","2","3","4",..: 1 1 1 1 1 1 1
1 1 2 ...
##  $ month           : int  1 2 3 4 5 6 7 8 9 1 ...
##  $ length          : num  2.5 3 3.6 4.3 4.9 5.5 6.1 6.7 7.2 2.3 ...
##  $ fungal_treatment: chr  "Fungal_treat" "Fungal_treat" "Fungal_treat"
"Fungal_treat" ...
```

```
head(nail_fungal)
```

```
##   observation person month length fungal_treatment
## 1           1      1     1    2.5     Fungal_treat
## 2           2      1     2    3.0     Fungal_treat
## 3           3      1     3    3.6     Fungal_treat
## 4           4      1     4    4.3     Fungal_treat
## 5           5      1     5    4.9     Fungal_treat
## 6           6      1     6    5.5     Fungal_treat
```

b. Visualise the data using the following code and describe the main trends and seedling effects in less than 40 words.

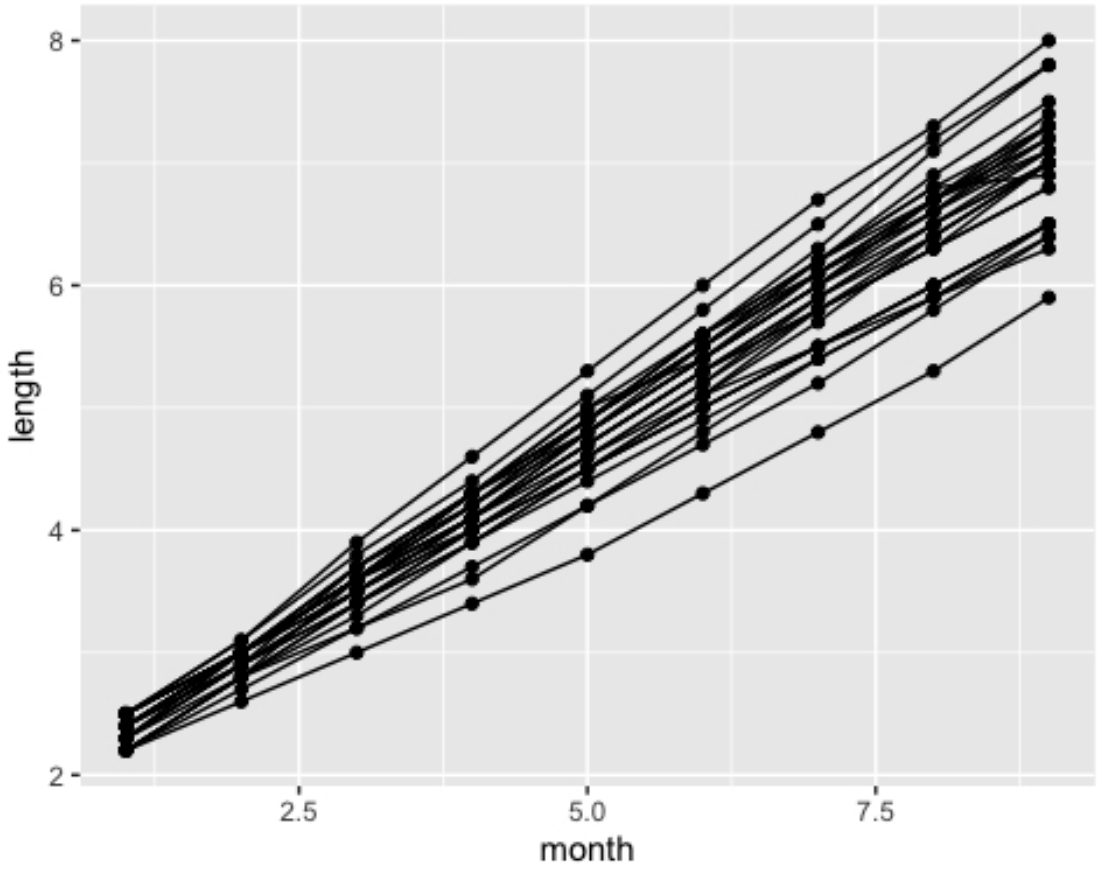
```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
nail_fungal %>%
  ggplot( aes(y=length, x=month, group=person))+
  geom_point()+
  geom_line()
```



The graph shows that as the month increases, the nail bed length increases as well. This shows a positive relationship. Furthermore, each person's nail bed length have a different trajectory/slope as the month increases.

c. Fit an appropriate model with: a random intercept, a random slope, a random slope and intercept (with correlation between the slopes and intercepts). Based on the plot you made in question 1.b, which of these three models would you expect to have the best fit and why? (less than 50 words)

random intercept

```
library(lme4)

## Loading required package: Matrix

m1 <- lmer(length ~ 1 + month + (1|person), data=nail_fungal)
print(summary(m1))

## Linear mixed model fit by REML ['lmerMod']
## Formula: length ~ 1 + month + (1 | person)
##    Data: nail_fungal
##
```

```
## REML criterion at convergence: -98.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.6709 -0.5417 -0.0099  0.4769  3.4124
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  person   (Intercept) 0.06833  0.2614
##  Residual             0.02819  0.1679
## Number of obs: 288, groups:  person, 32
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 1.786111   0.050992   35.03
## month       0.582292   0.003832  151.96
##
## Correlation of Fixed Effects:
##       (Intr)
## month -0.376
```

random slope

```
m2 <- lmer(length ~ 1 + month + (0 + month|person), data=nail_fungal)
print(summary(m2), correlation=FALSE)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: length ~ 1 + month + (0 + month | person)
##    Data: nail_fungal
##
## REML criterion at convergence: -410.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3454 -0.6450  0.0006  0.6500  2.3470
##
## Random effects:
##  Groups   Name  Variance Std.Dev.
##  person   month 0.002808 0.05299
##  Residual       0.008091 0.08995
## Number of obs: 288, groups:  person, 32
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  1.78611    0.01155  154.61
## month        0.58229    0.00959   60.72
```

a random slope and intercept

```
m3 <- lmer(length ~ 1 + month + (1 + month|person), data=nail_fungal)
print(summary(m3), correlation=FALSE)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: length ~ 1 + month + (1 + month | person)
##    Data: nail_fungal
##
## REML criterion at convergence: -474.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6529 -0.5556  0.0341  0.5664  2.7359
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  person   (Intercept) 0.013180 0.11480
##           month       0.003172 0.05632  -0.33
##  Residual             0.005056 0.07110
## Number of obs: 288, groups:  person, 32
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  1.78611    0.02225   80.26
## month        0.58229    0.01009   57.72
```

I expect m3 to have the best fit because it fits a random slope and a random intercept, similar to Question1b plot. So, each person's nail bed length is different at the beginning of the study and their nail bed length growth is also different as the month increases.

d. For model m3, what is the correlation between the random effects? How do you interpret this value?

The correlation in m3 is -0.33, which explains the variability between the person's nail bed length as the month increases. In other terms, how the slope varies between each person's nail bed length in the study.

e. Select the best model and based on that model answer the following questions: (4 marks) 1. Which model has the best fit? 2. How much does the average nail bed grow per month? 3. According to the model, which person has the lowest initial length? What is the predicted initial length for this person? 4. According to the model, which person has the highest growth rate? What is the predicted growth rate for this person?

1.

```
AIC(m1, m2, m3)
```

```
##    df        AIC
## m1  4  -90.84797
## m2  4 -402.05602
## m3  6 -462.76077
```

The model with the best fit is m3 because it has lowest AIC.

2.

```
fixef(m3)
```

```
## (Intercept)       month
##   1.7861111   0.5822917
```

The average nail bed growth per month is 0.58 mm.

3.

```
ranef(m3)
```

```
## $person
##     (Intercept)        month
## 1   0.059920417  0.021122705
## 2  -0.104171613  0.064123430
## 3   0.046056922  0.024009551
## 4  -0.092517030  0.039618427
## 5   0.090100329 -0.016071144
## 6   0.113073983  0.007846534
## 7   0.048509843 -0.007410607
## 8   0.046026421  0.030639340
## 9  -0.015932294 -0.049186907
## 10 -0.027830885  0.028356411
## 11 -0.106167018 -0.003903253
## 12 -0.155085765  0.092387969
## 13  0.138653064 -0.032804893
## 14 -0.034671129  0.009910466
## 15  0.011367683  0.037856454
## 16 -0.009641068  0.088595249
## 17 -0.119725503 -0.067314302
## 18 -0.076353119  0.038460371
## 19 -0.080831946  0.008483635
## 20  0.016029516  0.028054453
## 21  0.113500998 -0.084970518
## 22  0.032223928  0.020266607
## 23  0.164232144 -0.073456320
## 24 -0.073839196 -0.006219367
## 25 -0.121751409 -0.128711196
## 26 -0.193948821  0.009960242
## 27 -0.105923009 -0.056941569
## 28 -0.069390870  0.030387159
```

```
## 29  0.178095640 -0.076343166
## 30  0.221986541 -0.083274913
## 31  0.096971074 -0.004254988
## 32  0.011032171  0.110784138
##
## with conditional variances for "person"
```

Person number 26 has the lowest initial length with a predicted initial length of 1.7861111 + (-0.193948821) = 1.59 mm.

4.

Person number 32 has the highest growth rate with a predicted growth rate of 0.5822917 + 0.110784138 = 0.69 mm.

Question 2

a. Read in the abalone.csv data and name your data abalone_df. Inspect the abalone_df data using the head().

```
abalone_df <- read.csv("abalone.csv")
head(abalone_df, 20)
```

```
##     X Type LongestShell Diameter Height WholeWeight ShuckedWeight
VisceraWeight
## 1   1    M        0.455    0.365  0.095      0.5140        0.2245
0.1010
## 2   2    M        0.350    0.265  0.090      0.2255        0.0995
0.0485
## 3   3    F        0.530    0.420  0.135      0.6770        0.2565
0.1415
## 4   4    M        0.440    0.365  0.125      0.5160        0.2155
0.1140
## 5   5    I        0.330    0.255  0.080      0.2050        0.0895
0.0395
## 6   6    I        0.425    0.300  0.095      0.3515        0.1410
0.0775
## 7   7    F        0.530    0.415  0.150      0.7775        0.2370
0.1415
## 8   8    F        0.545    0.425  0.125      0.7680        0.2940
0.1495
## 9   9    M        0.475    0.370  0.125      0.5095        0.2165
0.1125
## 10 10    F        0.550    0.440  0.150      0.8945        0.3145
0.1510
## 11 11    F        0.525    0.380  0.140      0.6065        0.1940
0.1475
```

```
## 12 12    M        0.430    0.350  0.110      0.4060        0.1675
0.0810
## 13 13    M        0.490    0.380  0.135      0.5415        0.2175
0.0950
## 14 14    F        0.535    0.405  0.145      0.6845        0.2725
0.1710
## 15 15    F        0.470    0.355  0.100      0.4755        0.1675
0.0805
## 16 16    M        0.500    0.400  0.130      0.6645        0.2580
0.1330
## 17 17    I        0.355    0.280  0.085      0.2905        0.0950
0.0395
## 18 18    F        0.440    0.340  0.100      0.4510        0.1880
0.0870
## 19 19    M        0.365    0.295  0.080      0.2555        0.0970
0.0430
## 20 20    M        0.450    0.320  0.100      0.3810        0.1705
0.0750
##    ShellWeight Rings
## 1        0.150    15
## 2        0.070     7
## 3        0.210     9
## 4        0.155    10
## 5        0.055     7
## 6        0.120     8
## 7        0.330    20
## 8        0.260    16
## 9        0.165     9
## 10       0.320    19
## 11       0.210    14
## 12       0.135    10
## 13       0.190    11
## 14       0.205    10
## 15       0.185    10
## 16       0.240    12
## 17       0.115     7
## 18       0.130    10
## 19       0.100     7
## 20       0.115     9
```

I would not include the X column variable in my multivariate analysis because it is a categorical variable. I'm only needing the measurements, not the order of the mollusks i.e. if it's the first mollusk or the second one. Hence, I'll remove it.

```
abalone_active_df <- abalone_df %>% dplyr::select(-c(X, Type))
str(abalone_active_df)
```

```
## 'data.frame':    4177 obs. of  8 variables:
##  $ LongestShell : num  0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475
0.55 ...
```

```
##  $ Diameter     : num  0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37
0.44 ...
##  $ Height       : num  0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125
0.15 ...
##  $ WholeWeight  : num  0.514 0.226 0.677 0.516 0.205 ...
##  $ ShuckedWeight: num  0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ VisceraWeight: num  0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ ShellWeight  : num  0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165
0.32 ...
##  $ Rings        : int  15 7 9 10 7 8 20 16 9 19 ...
```

```
head(abalone_active_df)
```

```
##   LongestShell Diameter Height WholeWeight ShuckedWeight VisceraWeight
## 1        0.455    0.365  0.095      0.5140        0.2245        0.1010
## 2        0.350    0.265  0.090      0.2255        0.0995        0.0485
## 3        0.530    0.420  0.135      0.6770        0.2565        0.1415
## 4        0.440    0.365  0.125      0.5160        0.2155        0.1140
## 5        0.330    0.255  0.080      0.2050        0.0895        0.0395
## 6        0.425    0.300  0.095      0.3515        0.1410        0.0775
##   ShellWeight Rings
## 1       0.150    15
## 2       0.070     7
## 3       0.210     9
## 4       0.155    10
## 5       0.055     7
## 6       0.120     8
```

b. Scale all the variables in the abalone_active_df data set using the scale() function. Name the scaled data abalone_active_scaled_df. Explain in a couple of sentences what the scaling does and why it is important.

```
abalone_active_scaled_df <- apply(abalone_active_df, 2, scale)
head(abalone_active_scaled_df)
```

```
##      LongestShell   Diameter     Height WholeWeight ShuckedWeight
VisceraWeight
## [1,]   -0.5744894 -0.4320971 -1.0642967  -0.6418214    -0.6076126    -
0.7261246
## [2,]   -1.4488124 -1.4397566 -1.1838366  -1.2301298    -1.1707697    -
1.2050770
## [3,]    0.0500271  0.1221157 -0.1079779  -0.3094322    -0.4634444    -
0.3566471
## [4,]   -0.6993926 -0.4320971 -0.3470576  -0.6377430    -0.6481599    -
0.6075269
## [5,]   -1.6153501 -1.5405226 -1.4229163  -1.2719334    -1.2158222    -
1.2871831
## [6,]   -0.8242959 -1.0870758 -1.0642967  -0.9731910    -0.9838015    -
0.9405128
##      ShellWeight       Rings
```

```
## [1,]  -0.6381405  1.57135544
## [2,]  -1.2128421 -0.90990405
## [3,]  -0.2071143 -0.28958918
## [4,]  -0.6022216  0.02056826
## [5,]  -1.3205987 -0.90990405
## [6,]  -0.8536536 -0.59974661
```

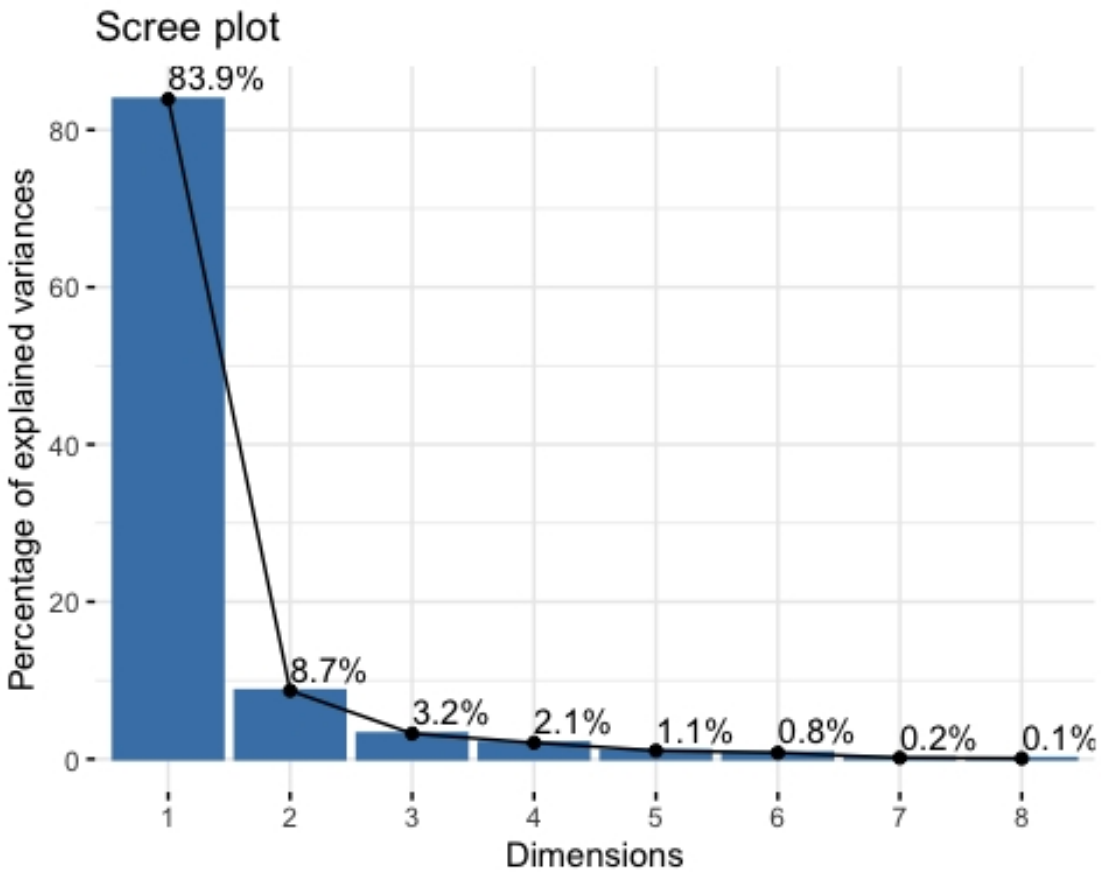
Scaling standardises each variable to a mean of zero and a variance of 1. This is important because it makes comparing each principal component (PCA analysis) to the mean straightforward, and removes potential problems with the scale of each variable.

c. Perform a PCA analyses on all the variables in the abalone_active_scaled_df data and name your model 'my_pca'

```
my_pca <- prcomp(abalone_active_scaled_df, scale = TRUE)
```

d. Use the code below to make a scree plot of the PCA model. Install the factoextra package if necessary.

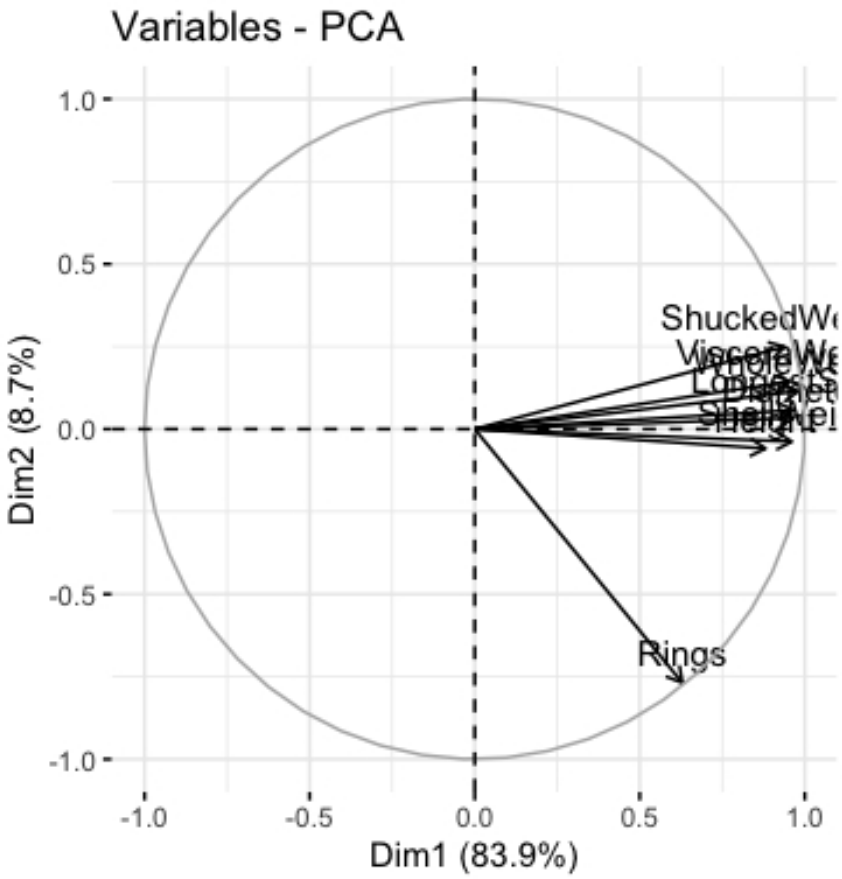
```
library("factoextra")

## Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa

fviz_eig(my_pca,addlabels = TRUE)

## Registered S3 methods overwritten by 'car':
##   method                           from
##   influence.merMod                 lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod          lme4
##   dfbetas.influence.merMod         lme4
```



The first two principal components combined explain 92.6% variance in the data.

Therefore, I would recommend using 2 principal components based on this scree plot as the elbow break point happens at the second principal component (where it explains a relatively large proportion of the variation in the data).

```
fviz_pca_var(my_pca)
```



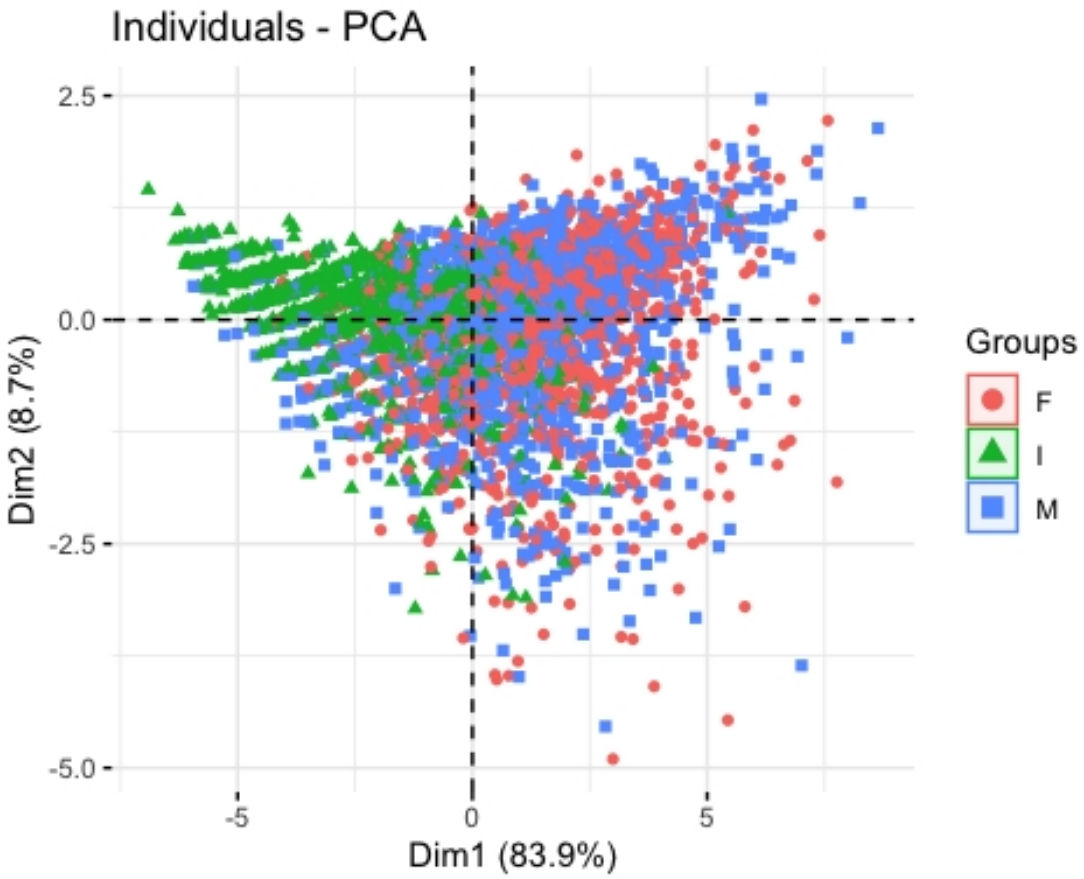

```
my_pca$rotation

##                     PC1         PC2         PC3         PC4         PC5
## LongestShell  0.3721385  0.06828270 -0.03107032  0.60405352 -0.01112485
## Diameter      0.3730941  0.04004804 -0.04100507  0.58759470  0.05791102
## Height        0.3400268 -0.07046315 -0.89970562 -0.25677704 -0.05672927
## WholeWeight   0.3783075  0.13734619  0.20619361 -0.24184895  0.01565561
## ShuckedWeight 0.3624545  0.29883992  0.20828630 -0.18324566 -0.39852530
## VisceraWeight 0.3685578  0.17297852  0.19737954 -0.26522056 -0.30982149
## ShellWeight   0.3707578 -0.04540040  0.16157408 -0.24419207  0.83056377
## Rings         0.2427128 -0.92120385  0.19214359 -0.04331013 -0.22002569
##                     PC6          PC7           PC8
## LongestShell   0.04749683 -0.698825733  0.0163485531
## Diameter       0.02337540  0.712985166 -0.0002192549
## Height        -0.02669146 -0.008614452  0.0026881714
## WholeWeight   -0.11725505  0.008331288  0.8502643707
## ShuckedWeight -0.62489286  0.009281766 -0.3911005419
## VisceraWeight  0.76584381  0.027345539 -0.2041790306
## ShellWeight   -0.03283235 -0.047395080 -0.2856239917
## Rings         -0.06819579 -0.008420573 -0.0233703940
```

The variable Rings has a high loading on the second principal component as it is further down based on the biplot. Also, it has the highest PC2 value in the loadings table (ignoring the negative sign).

I think the reason behind why the all other variables have a similar loading on the first principal component is that the value in these variables are similar. First principal component explains as much variation in the data as possible, thus similar first principal component means similar values in all the other variables.

f. Run the code below to map the abalone data on the new coordinate systems obtained by the performed PCA. The points are color coded by Type of abalone. Which principal component differentiates most between adults and infant molusks? Why do you think this is the case?

```
groups <- as.factor(abalone_df$Type[1:4177])
fviz_pca_ind(my_pca,
             geom=c("point"),
             col.ind = groups, # color by groups
             #palette = c("#00AFBB",  "#FC4E07"),
             addEllipses = TRUE, # Concentration ellipses
             ellipse.type = "confidence",
             legend.title = "Groups",
             repel = FALSE
             )
```



First principal component differentiates most between adults and infant mollusks. This is because most infant mollusks are on the left side of the graph whereas most adults mollusks are on the right side of the graph.

Based on the graph, we can't use the second principal component to differentiate between the different types of mollusks because infant, female and male mollusks are scattered almost evenly at the top and bottom of the graph. Also, the second principal component explains 8.7% variability which is little.

The loading of rings on the first and second principal component explains its impact on its principal components. The rings loading on the first principal component is 0.2427128 which is a low loading meaning that the first principal component is not focused on the rings. This indicates that the first principal component for rings does not tell much about the rings therefore not a good candidate for determining the age of an abalone. However, the rings loading on the second principal component is -0.92120385 which is a (strong) high loading meaning that the second principal component is focused on the rings and that it tells more about the rings. Therefore, it is a better candidate for determining the age of an abalone.