PCA

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4  
## ✔ tibble 3.1.8 ✔ dplyr 1.0.9  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(flextable)

##   
## Attaching package: 'flextable'  
##   
## The following object is masked from 'package:purrr':  
##   
## compose

library(readxl)  
library(extrafont)

## Registering fonts with R

#library(forcats)  
library(writexl)  
library(ggfortify) #autoplot  
library(scales) #percent()

##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

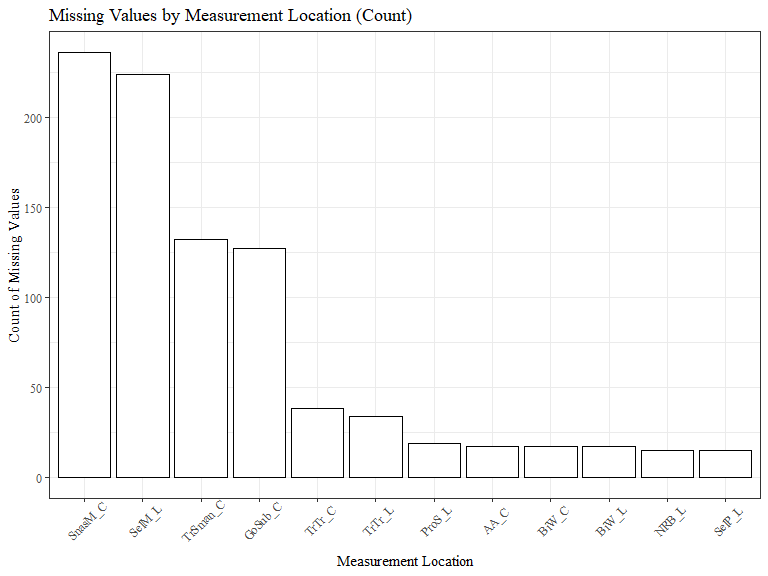
#times new roman tables  
my\_ft\_theme <- function(ft, ...) {  
 # Remove vertical cell padding  
 ft <- padding(ft, padding.top = 0, padding.bottom = 0, part = "all")  
   
 # Change font to TNR 11  
 ft <- font(ft, fontname = "Times New Roman", part = "all")  
 ft <- fontsize(ft, part = "all", size = 12)  
 ft  
}

measureNAs <- read\_excel("C:\\Users\\19177\\OneDrive - Colostate\\Desktop\\Dissertation\\headscan\_dissertation\\measureNAs.xlsx")  
  
PCA\_NAs <- slice(measureNAs, c(1,3,4,6,7,11,13,15,16,24,26,27))  
  
#Size 12 Table TNR  
flextable(PCA\_NAs) %>%  
 my\_ft\_theme()%>%   
 bold(part = "header") %>%   
 set\_caption("NA values for PCA Measurement Locations, total 891 NA values") %>%   
 autofit() %>%   
 set\_header\_labels(values = list(measure\_name = "Measurement Location",  
 measureNAprops = "Proportion of NA values",  
 measureNAsums = "Count of NA values"))

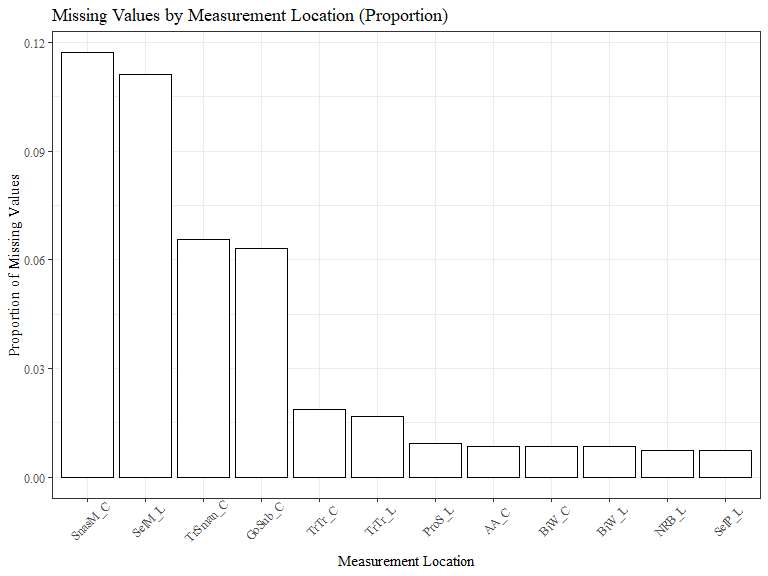
**Table** : NA values for PCA Measurement Locations, total 891 NA values

| **Measurement Location** | **Proportion of NA values** | **Count of NA values** |
| --- | --- | --- |
| AA\_C | 0.0084 | 17 |
| BiW\_C | 0.0084 | 17 |
| BiW\_L | 0.0084 | 17 |
| GoSub\_C | 0.0630 | 127 |
| NRB\_L | 0.0074 | 15 |
| ProS\_L | 0.0094 | 19 |
| SelP\_L | 0.0074 | 15 |
| SelM\_L | 0.1111 | 224 |
| SnasM\_C | 0.1171 | 236 |
| TrSman\_C | 0.0655 | 132 |
| TrTr\_C | 0.0188 | 38 |
| TrTr\_L | 0.0169 | 34 |

PCA\_NAs$measure\_name <- fct\_reorder(PCA\_NAs$measure\_name, PCA\_NAs$measureNAsums, .desc=TRUE)  
  
#bar chart with counts  
ggplot(data=PCA\_NAs, aes(x=measure\_name, y=measureNAsums))+  
 geom\_bar(stat= "identity", color= "black", fill = "white")+  
 theme\_bw()+theme(text=element\_text(family= "Times New Roman"))+  
 theme(axis.text.x = element\_text(angle = 45, vjust=0.7))+  
 labs(title="Missing Values by Measurement Location (Count)",  
 y="Count of Missing Values",  
 x="Measurement Location")



#bar chart with proportions  
ggplot(data=PCA\_NAs, aes(x=measure\_name, y=measureNAprops))+  
 geom\_bar(stat= "identity", color= "black", fill = "white")+  
 theme\_bw()+theme(text=element\_text(family= "Times New Roman"))+  
 theme(axis.text.x = element\_text(angle = 45, vjust=0.7))+  
 labs(title="Missing Values by Measurement Location (Proportion)",  
 y="Proportion of Missing Values",  
 x="Measurement Location")



#PCAdata\_num <- PCAdata\_num %>% drop\_na()  
PCAdata\_full<-read\_excel("C:\\Users\\19177\\OneDrive - Colostate\\Desktop\\Dissertation\\headscan\_dissertation\\chosen\_nona.xlsx")  
  
  
  
PCAdata\_num <- select\_if(PCAdata\_full, is.numeric)  
  
str(PCAdata\_num)

## tibble [1,677 × 12] (S3: tbl\_df/tbl/data.frame)  
## $ AA\_C : num [1:1677] 65 55 70 58 67 60 59 59 65 65 ...  
## $ BiW\_C : num [1:1677] 130 127 143 140 137 130 141 138 143 150 ...  
## $ BiW\_L : num [1:1677] 115 108 121 109 104 106 109 111 113 116 ...  
## $ GoSub\_C : num [1:1677] 93 93 115 93 103 100 79 106 85 102 ...  
## $ NRB\_L : num [1:1677] 17 18 19 21 19 14 17 18 16 17 ...  
## $ ProS\_L : num [1:1677] 17 18 14 13 20 20 18 12 24 22 ...  
## $ SelP\_L : num [1:1677] 42 41 51 44 47 48 46 41 46 44 ...  
## $ SelM\_L : num [1:1677] 122 99 130 115 119 126 117 112 117 117 ...  
## $ SnasM\_C : num [1:1677] 82 55 84 74 73 80 78 76 64 75 ...  
## $ TrSman\_C: num [1:1677] 177 145 178 147 157 164 149 159 151 160 ...  
## $ TrTr\_C : num [1:1677] 296 276 292 273 279 300 283 275 307 286 ...  
## $ TrTr\_L : num [1:1677] 155 141 156 149 146 146 147 151 157 144 ...

#https://www.statology.org/principal-components-analysis-in-r/  
  
#calculate principal components  
pca\_res <- prcomp(PCAdata\_num, scale=TRUE)  
  
#reverse the signs (R calculates eigenvectors in negative direction)  
pca\_res$rotation <- -1\*pca\_res$rotation  
  
#display PCs  
pca\_res$rotation

## PC1 PC2 PC3 PC4 PC5 PC6  
## AA\_C 0.22610676 -0.3779251 -0.27191984 -0.02721567 0.3556110 -0.12090072  
## BiW\_C 0.31862749 0.3092589 -0.14925876 0.35514129 0.2108243 0.24615880  
## BiW\_L 0.31985590 0.3603274 0.06247600 0.32013256 0.1828813 0.20770972  
## GoSub\_C 0.26391016 -0.3194544 0.39647678 -0.15385828 -0.3479355 0.06760927  
## NRB\_L 0.13011314 0.2687254 0.20783339 -0.62739232 0.5704433 -0.29655591  
## ProS\_L 0.09639314 -0.4223793 -0.26287412 -0.25265740 0.2689233 0.67499183  
## SelP\_L 0.19424050 -0.1133301 -0.63600471 0.01513560 -0.1689219 -0.47381013  
## SelM\_L 0.35269639 0.1874002 -0.28461427 -0.32224625 -0.3309942 0.07666820  
## SnasM\_C 0.29601013 0.3659376 -0.06456278 -0.29055474 -0.2928247 0.16642690  
## TrSman\_C 0.36861199 -0.2192867 0.30097637 -0.06486586 -0.1644443 -0.01555157  
## TrTr\_C 0.37142151 -0.1152908 0.15450138 0.18924213 0.1257169 -0.21189804  
## TrTr\_L 0.35440086 -0.1891442 0.16203007 0.24699373 0.1008921 -0.17407053  
## PC7 PC8 PC9 PC10 PC11  
## AA\_C 0.4862279 -0.58217734 0.019752828 -0.01922694 0.04225697  
## BiW\_C -0.2398765 -0.23360217 -0.006951676 0.03215282 -0.62595677  
## BiW\_L -0.1748471 -0.12365882 0.016096432 -0.14704919 0.66078776  
## GoSub\_C -0.2771165 -0.31614736 0.036009203 -0.57814816 -0.09591045  
## NRB\_L -0.2261964 0.02151877 -0.056541435 -0.05598251 -0.04673018  
## ProS\_L -0.1716974 0.31391995 0.014048789 -0.02036862 0.10095828  
## SelP\_L -0.4443011 0.04196601 -0.007291606 -0.07718999 0.18700636  
## SelM\_L 0.1099394 0.14528113 -0.012688564 0.04320080 -0.24290730  
## SnasM\_C 0.4378817 -0.02520448 -0.042472041 -0.05004832 0.18605244  
## TrSman\_C -0.2119913 -0.15522542 0.005263963 0.78090703 0.10807178  
## TrTr\_C 0.1981064 0.43534824 0.695647070 -0.08789820 -0.07075662  
## TrTr\_L 0.1805959 0.39385796 -0.713190608 -0.10628978 -0.05385218  
## PC12  
## AA\_C 0.11134458  
## BiW\_C -0.21065658  
## BiW\_L 0.28371343  
## GoSub\_C -0.01762083  
## NRB\_L -0.02340568  
## ProS\_L -0.11225570  
## SelP\_L -0.22697521  
## SelM\_L 0.66781644  
## SnasM\_C -0.58748037  
## TrSman\_C -0.05832774  
## TrTr\_C -0.05416325  
## TrTr\_L -0.01754129

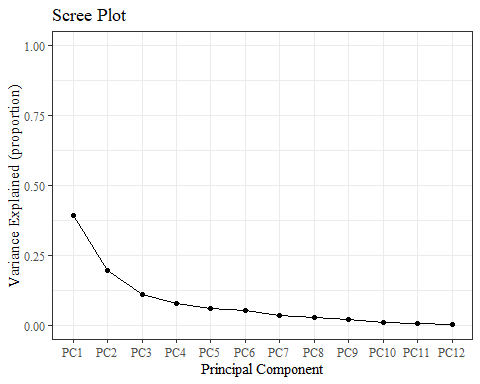
pcloadings <- as.data.frame(pca\_res$rotation)  
  
pcloadings <- rownames\_to\_column(pcloadings)  
  
pcloadings <- pcloadings %>%   
 rename(measure = "rowname")  
  
flextable(pcloadings) %>%  
 my\_ft\_theme()%>%   
 bold(part = "header") %>%   
 set\_caption("Principal Component Loadings") %>%   
 autofit()

**Table** : Principal Component Loadings

| **measure** | **PC1** | **PC2** | **PC3** | **PC4** | **PC5** | **PC6** | **PC7** | **PC8** | **PC9** | **PC10** | **PC11** | **PC12** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AA\_C | 0.22610676 | -0.3779251 | -0.27191984 | -0.02721567 | 0.3556110 | -0.12090072 | 0.4862279 | -0.58217734 | 0.019752828 | -0.01922694 | 0.04225697 | 0.11134458 |
| BiW\_C | 0.31862749 | 0.3092589 | -0.14925876 | 0.35514129 | 0.2108243 | 0.24615880 | -0.2398765 | -0.23360217 | -0.006951676 | 0.03215282 | -0.62595677 | -0.21065658 |
| BiW\_L | 0.31985590 | 0.3603274 | 0.06247600 | 0.32013256 | 0.1828813 | 0.20770972 | -0.1748471 | -0.12365882 | 0.016096432 | -0.14704919 | 0.66078776 | 0.28371343 |
| GoSub\_C | 0.26391016 | -0.3194544 | 0.39647678 | -0.15385828 | -0.3479355 | 0.06760927 | -0.2771165 | -0.31614736 | 0.036009203 | -0.57814816 | -0.09591045 | -0.01762083 |
| NRB\_L | 0.13011314 | 0.2687254 | 0.20783339 | -0.62739232 | 0.5704433 | -0.29655591 | -0.2261964 | 0.02151877 | -0.056541435 | -0.05598251 | -0.04673018 | -0.02340568 |
| ProS\_L | 0.09639314 | -0.4223793 | -0.26287412 | -0.25265740 | 0.2689233 | 0.67499183 | -0.1716974 | 0.31391995 | 0.014048789 | -0.02036862 | 0.10095828 | -0.11225570 |
| SelP\_L | 0.19424050 | -0.1133301 | -0.63600471 | 0.01513560 | -0.1689219 | -0.47381013 | -0.4443011 | 0.04196601 | -0.007291606 | -0.07718999 | 0.18700636 | -0.22697521 |
| SelM\_L | 0.35269639 | 0.1874002 | -0.28461427 | -0.32224625 | -0.3309942 | 0.07666820 | 0.1099394 | 0.14528113 | -0.012688564 | 0.04320080 | -0.24290730 | 0.66781644 |
| SnasM\_C | 0.29601013 | 0.3659376 | -0.06456278 | -0.29055474 | -0.2928247 | 0.16642690 | 0.4378817 | -0.02520448 | -0.042472041 | -0.05004832 | 0.18605244 | -0.58748037 |
| TrSman\_C | 0.36861199 | -0.2192867 | 0.30097637 | -0.06486586 | -0.1644443 | -0.01555157 | -0.2119913 | -0.15522542 | 0.005263963 | 0.78090703 | 0.10807178 | -0.05832774 |
| TrTr\_C | 0.37142151 | -0.1152908 | 0.15450138 | 0.18924213 | 0.1257169 | -0.21189804 | 0.1981064 | 0.43534824 | 0.695647070 | -0.08789820 | -0.07075662 | -0.05416325 |
| TrTr\_L | 0.35440086 | -0.1891442 | 0.16203007 | 0.24699373 | 0.1008921 | -0.17407053 | 0.1805959 | 0.39385796 | -0.713190608 | -0.10628978 | -0.05385218 | -0.01754129 |

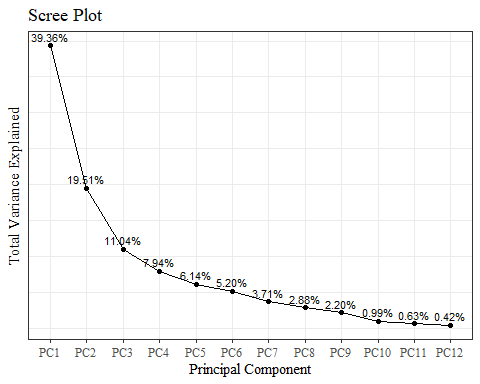
var\_explained = pca\_res$sdev^2 / sum(pca\_res$sdev^2)  
  
var\_explained\_total <- var\_explained

qplot(c(1:12), var\_explained) +   
 geom\_line() +   
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 xlab("Principal Component") +   
 ylab("Variance Explained (proportion)") +  
 ggtitle("Scree Plot") +  
 ylim(0, 1) +  
 scale\_x\_discrete(limits=c("PC1","PC2","PC3","PC4","PC5","PC6","PC7","PC8","PC9","PC10","PC11","PC12"))



var\_explained\_data <- data.frame(var\_explained)  
  
var\_explained\_data <- var\_explained\_data %>%   
 rename(v\_e = "var\_explained")  
  
  
var\_explained\_data$vep <- var\_explained\_data$v\_e  
  
var\_explained\_data <- rownames\_to\_column(var\_explained\_data, "PC\_num")  
  
var\_explained\_data$vep <- percent(var\_explained\_data$vep, accuracy=0.01)  
  
var\_explained\_data$PC\_num <- as.factor(var\_explained\_data$PC\_num)  
  
var\_explained\_data$PC\_num <-   
 recode\_factor(var\_explained\_data$PC\_num,   
 '1'= "PC1",  
 '2'= "PC2",  
 '3'= "PC3",  
 '4'= "PC4",  
 '5'= "PC5",  
 '6'= "PC6",  
 '7'= "PC7",  
 '8'= "PC8",  
 '9'= "PC9",  
 '10'= "PC10",  
 '11'= "PC11",  
 '12'= "PC12")

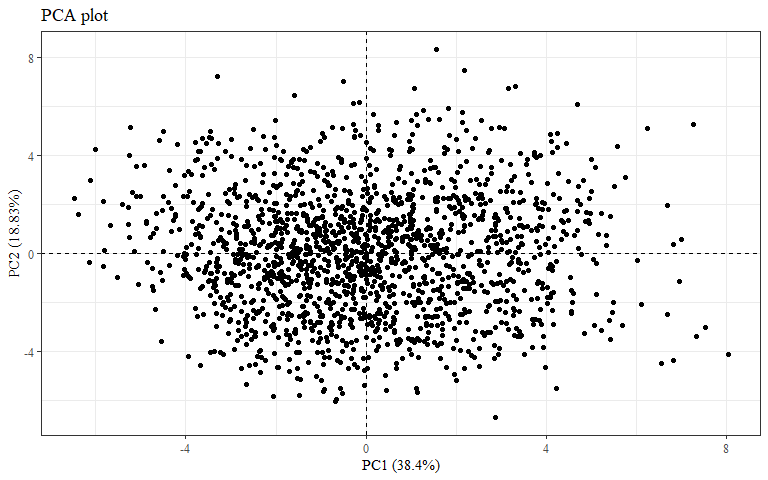
ggplot(data=var\_explained\_data, aes(x=PC\_num, y=v\_e, group=1)) +   
 geom\_line() +  
 geom\_point() +   
 geom\_text(aes(label=vep),  
 position= position\_dodge(0.9),  
 vjust = -0.5,   
 size = 3)+  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 theme(axis.text.y=element\_blank(),  
 axis.ticks.y=element\_blank()   
 )+  
 labs(title="Scree Plot",  
 x="Principal Component",  
 y="Total Variance Explained")



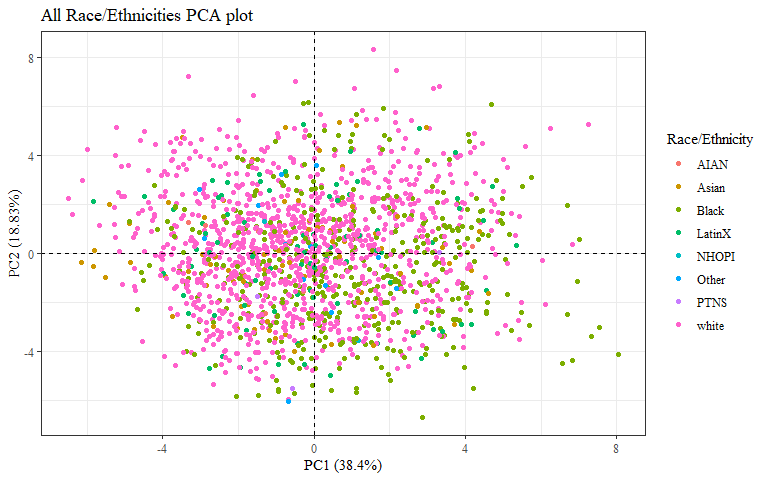
#making plot axis match the ggplot versions in PCA2.Rmd!!!   
#compare to PCAdata\_full1 pc1 and pc2 columns (should be same sign, here times 100)  
pca\_res$x <- pca\_res$x \* 100  
pca\_res$x[,1] <- pca\_res$x[,1] \* -1  
#pca\_res$x <- pca\_res$x[,2] \* 100  
  
#pca\_res$x

#<https://cran.r-project.org/web/packages/ggfortify/vignettes/plot_pca.html>

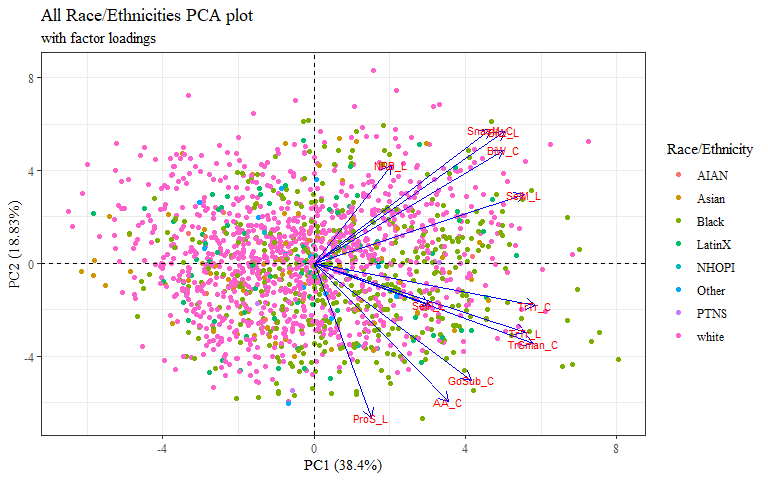
autoplot(pca\_res, data=PCAdata\_full)+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+   
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title="PCA plot")



autoplot(pca\_res, data=PCAdata\_full, colour="race\_eth")+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman")) +  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Race/Ethnicities PCA plot",  
 color = "Race/Ethnicity")

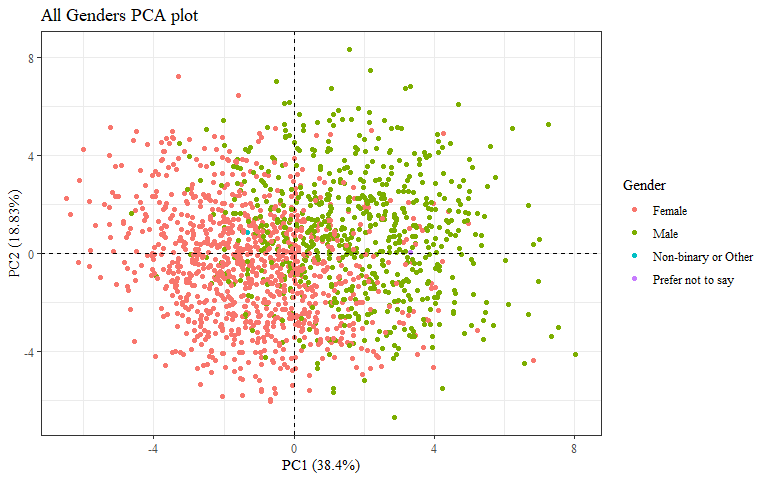


autoplot(pca\_res, data=PCAdata\_full, colour="race\_eth",  
 loadings = TRUE, loadings.colour = 'blue',  
 loadings.label = TRUE, loadings.label.size = 3)+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Race/Ethnicities PCA plot",  
 subtitle = "with factor loadings",  
 color = "Race/Ethnicity")

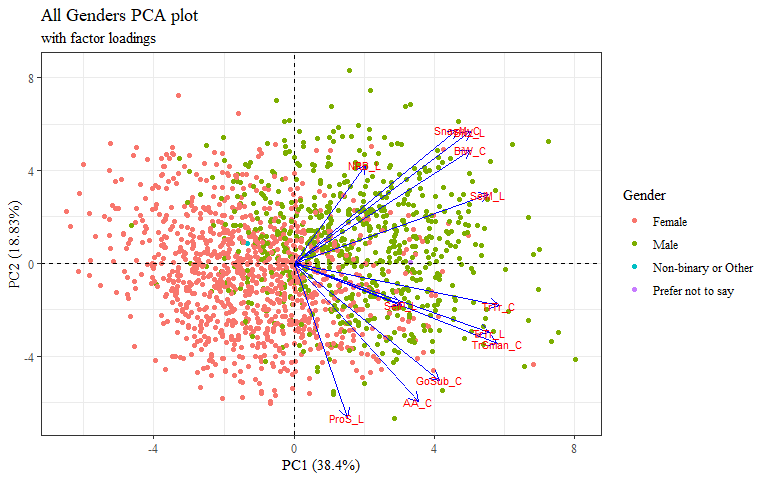
 arrows close together=higher correlation

COULD add ellipses (like in PCA2) to this graph with factor loadings.. but cannot add loadings to plot in PCA2 with ellipses

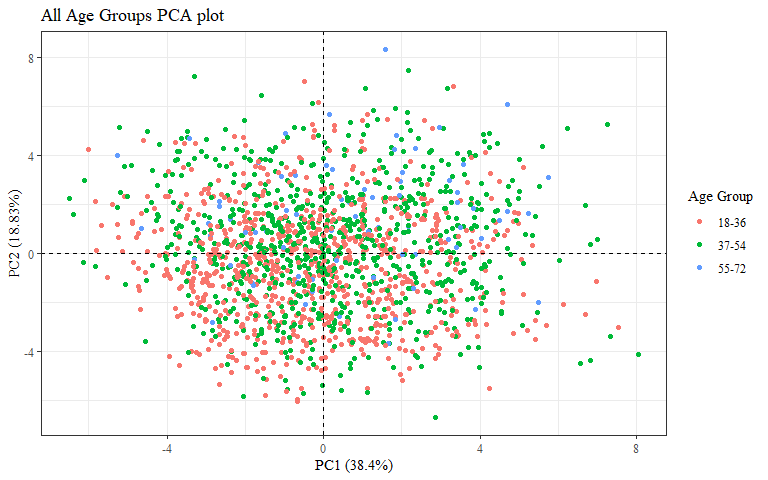
autoplot(pca\_res, data=PCAdata\_full, colour="gender")+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Genders PCA plot",  
 color = "Gender")



autoplot(pca\_res, data=PCAdata\_full, colour="gender",  
 loadings = TRUE, loadings.colour = 'blue',  
 loadings.label = TRUE, loadings.label.size = 3)+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Genders PCA plot",  
 subtitle = "with factor loadings",  
 color = "Gender")



autoplot(pca\_res, data=PCAdata\_full, colour="age\_group")+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Age Groups PCA plot",  
 color = "Age Group")



autoplot(pca\_res, data=PCAdata\_full, colour="age\_group",  
 loadings = TRUE, loadings.colour = 'blue',  
 loadings.label = TRUE, loadings.label.size = 3)+  
 geom\_hline(yintercept = 0, lty = 2) +  
 geom\_vline(xintercept = 0, lty = 2) +  
 theme\_bw() + theme(text=element\_text(family= "Times New Roman"))+  
 labs(y="PC2 (18.83%)",  
 x="PC1 (38.4%)",  
 title = "All Age Groups PCA plot",  
 subtitle = "with factor loadings",  
 color = "Age Group")

