Principal Components Analysis

Homework Week 10

The Solution

06 April 2019

```
pacman::p_load(vegan, tidyverse, broom, ggordiplots, RVAideMemoire, pander )
```

Data preparation, identification

```
 load("C:/Users/Devan.McGranahan/GoogleDrive/Teaching/Classes/Analysis of Ecosystems/compiled notes/AoE
str(mtcars2)
```

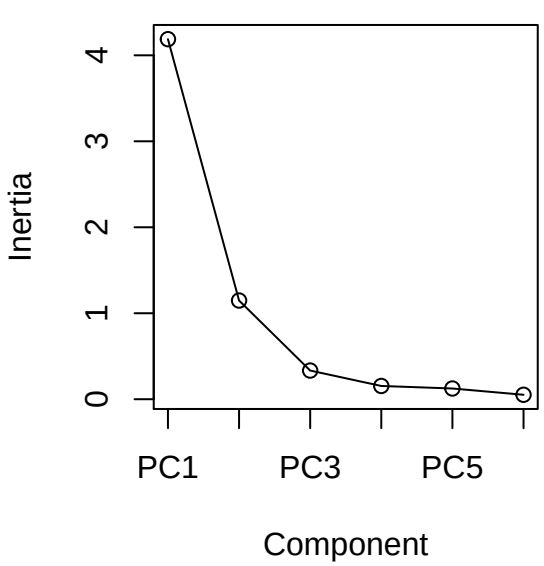
```
## 'data.frame':    32 obs. of  16 variables:
##  $ make.model: Factor w/ 32 levels "AMC Javelin",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ make      : Factor w/ 20 levels "AMC","Cadillac",..: 1 2 3 4 5 6 16 7 8 8 ...
##  $ origin    : Factor w/ 2 levels "domestic","foreign": 1 1 1 1 2 1 1 2 2 2 ...
##  $ country   : Factor w/ 6 levels "Germany","Italy",..: 6 6 6 6 3 6 6 2 2 2 ...
##  $ continent : Factor w/ 3 levels "Asia","Europe",..: 3 3 3 3 1 3 3 2 2 2 ...
##  $ am        : Factor w/ 2 levels "Automatic","Manual": 1 1 1 1 2 1 1 2 2 2 ...
##  $ vs        : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 2 ...
##  $ gear      : Factor w/ 3 levels "3","4","5": 1 1 1 1 2 1 1 3 2 2 ...
##  $ carb      : Factor w/ 6 levels "1","2","3","4",..: 2 4 4 4 1 2 4 5 1 1 ...
##  $ cyl       : Factor w/ 3 levels "4","6","8": 3 3 3 3 1 3 3 2 1 1 ...
##  $ disp      : num  304 472 350 440 108 318 360 145 78.7 79 ...
##  $ hp        : int  150 205 245 230 93 150 245 175 66 66 ...
##  $ drat      : num  3.15 2.93 3.73 3.23 3.85 2.76 3.21 3.62 4.08 4.08 ...
##  $ wt        : num  3.44 5.25 3.84 5.34 2.32 ...
##  $ qsec      : num  17.3 18 15.4 17.4 18.6 ...
##  $ mpg       : num  15.2 10.4 13.3 14.7 22.8 15.5 14.3 19.7 32.4 27.3 ...
```

- Which variables look appropriate for inclusion in the site x species matrix (ordination data)?

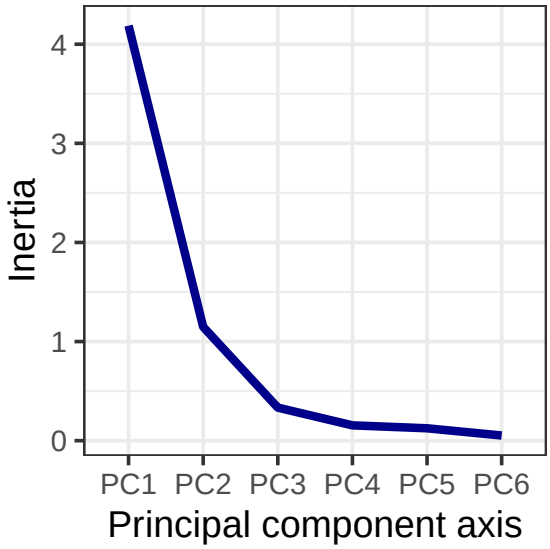
  The continuous variables `disp`–`mpg` are good variables for a PCA.

- Which might be "environmental" variables to test?

  Any of the multi-level factor variables that have several items/level; `make.model` and `make` do not have enough levels to test variabilty.



(a) Plain old `base` graphics.



(b) `ggplot` can handle ordination objects when `tibble::enframe` is applied to the correct part of the object (`eig`).

Figure 1: Two approaches to making a screeplot for the `mtcars2` PCA.

Analysis

Fit and assess ordination

Fit the PCA

```
cars.d <-
  mtcars2 %>%
    select(., .data$disp :.data$mpg )

cars.pca <- rda(cars.d ~ 1, scale=TRUE)
```

Assess the PCA

```
par(mar=c(4,4,1,1))
screeplot(cars.pca, type="l", main=" ")

enframe(cars.pca$CA$eig) %>%
  plyr::rename(c("value"="Inertia")) %>%
ggplot() + theme_bw(14) +
  geom_path(aes(x=name, y=Inertia, group=1),
            color="darkblue", size=1.5) +
  labs(x="Principal component axis")
```

The screeplot (Fig. 1) suggests marginal reductions in inertia (total variation explained) beyond PC2.

```
round(summary(eigenvals(cars.pca)), 2 ) %>%
  pander(caption="Eigenvalues and proportion explained by each axis in the PCA.")
```

Table 1: Eigenvalues and proportion explained by each axis in the PCA.

	PC1	PC2	PC3	PC4	PC5	PC6
Eigenvalue	4.19	1.15	0.33	0.15	0.12	0.05
Proportion Explained	0.7	0.19	0.06	0.03	0.02	0.01
Cumulative Proportion	0.7	0.89	0.94	0.97	0.99	1

```
round(scores(cars.pca, choices=c(1:6))$species, 2) %>%
  pander("Loadings for each variable along each PC, representing the relative influence
         of each variable on variation in each axis.")
```

Table 2: Loadings for each variable along each PC, representing the relative influence of each variable on variation in each axis.

	PC1	PC2	PC3	PC4	PC5	PC6
disp	-1.44	0.1	-0.08	-0.36	-0.16	0.2
hp	-1.31	-0.58	-0.13	-0.07	0.43	-0.02
drat	1.13	-0.71	-0.7	-0.01	-0.08	0.04
wt	-1.35	0.48	-0.36	-0.06	-0.12	-0.24
qsec	0.78	1.23	-0.3	-0.03	0.22	0.08
mpg	1.42	-0.09	0.17	-0.46	0.06	-0.12

- How much variation is explained by the first two axes (PCs) of the ordination? Would you consider this an acceptable amount?

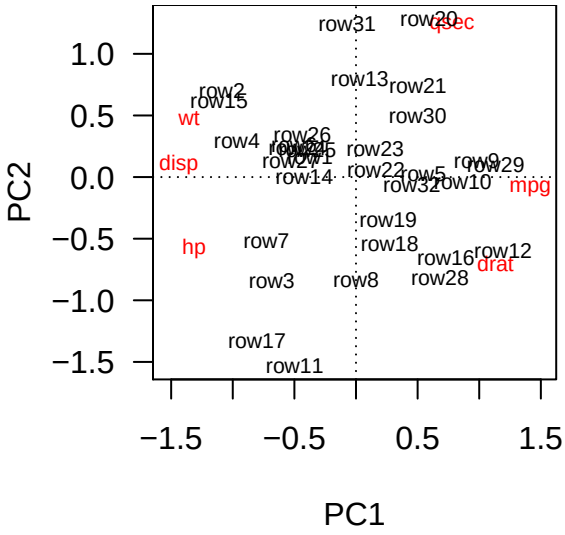
  The first two axes (PC1 and PC2) cumulatively explain 89% of the variation, well beyond the minimum acceptable 70%, so yes, this seems a good ordination.

- Which variable(s) contribute the most to PC1 and PC2? Along PC1, do you suspect they are negatively or positively correlated? How can you tell?

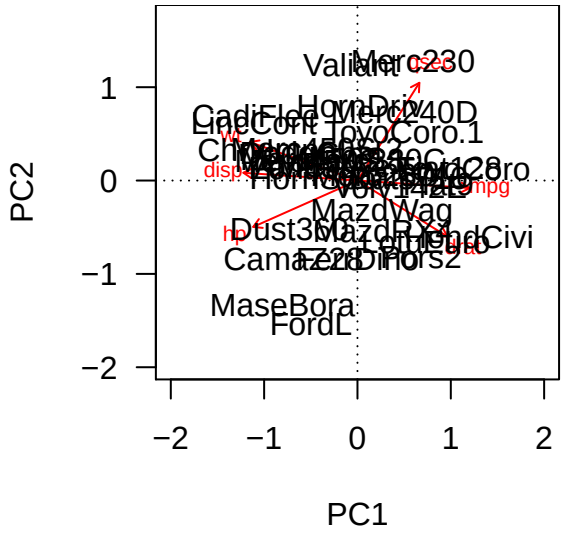
  `disp` and `mpg` contribute the most to PC1, while `qsec` and `drat` contribute the most to PC2. Along PC1, `disp` and `mpg` have a negative relationship because their loadings are negative and positive, respectively.

Plot the PCA

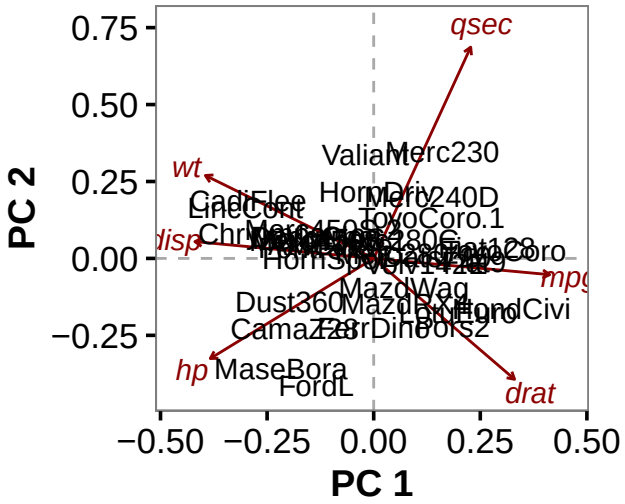
```
# a
par(mar=c(4,5,0,2))
    plot(cars.pca, las=1)
# b
par(mar=c(4,5,0,2))
    biplot(cars.pca, display="species", las=1,
           xlim=c(-2,2), ylim=c(-1.75,1.5))
    text(cars.pca, display = "sites",
         labels=make.cepnames(mtcars2$make.model))
# c
    ggplot() + theme_ord(14) +
      labs(x="PC 1", y="PC 2") +
      geom_vline(xintercept = 0, lty=2, color="darkgrey") +
      geom_hline(yintercept = 0, lty=2, color="darkgrey") +
      geom_segment(data=as_tibble(cars.pca$CA$v),
                   aes(x=0, y=0, xend=PC1*0.90, yend=PC2*0.90),
                   color="darkred",
                   arrow = arrow(length = unit(0.02, "npc")))+
      geom_text(data=as_tibble(cars.pca$CA$v),
                aes(x=PC1, y=PC2,
                    label=row.names(cars.pca$CA$v)),
                fontface="italic", color="darkred")  +
      geom_text(data=as_tibble(cars.pca$CA$u),
                aes(x=PC1, y=PC2,
                    label=make.cepnames(mtcars2$make.model)))
```



(a) Plain old base `plot`.



(b) With `vegan::biplot` using `base` graphics.



(c) Here with `ggplot`; calling the `u` and `v` objects require `tibble::as_tibble` to properly format the objects.

Figure 2: Three ways to plot the PCA. Note the use of `vegan::make.cepnames` to automatically abbreviate the `make.model` vector.

- True or False: The site scores are a mess.

  True!

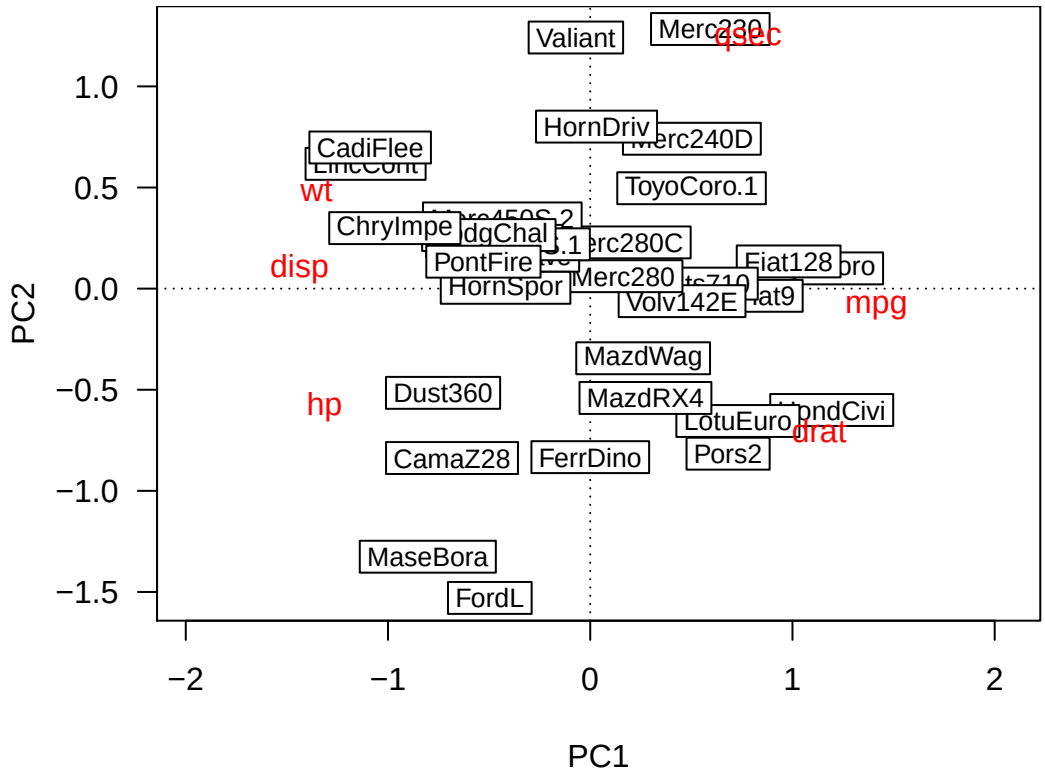
- What might be done to improve this?

  A plot that excludes less-important or overlapping site scores would be easier to read.

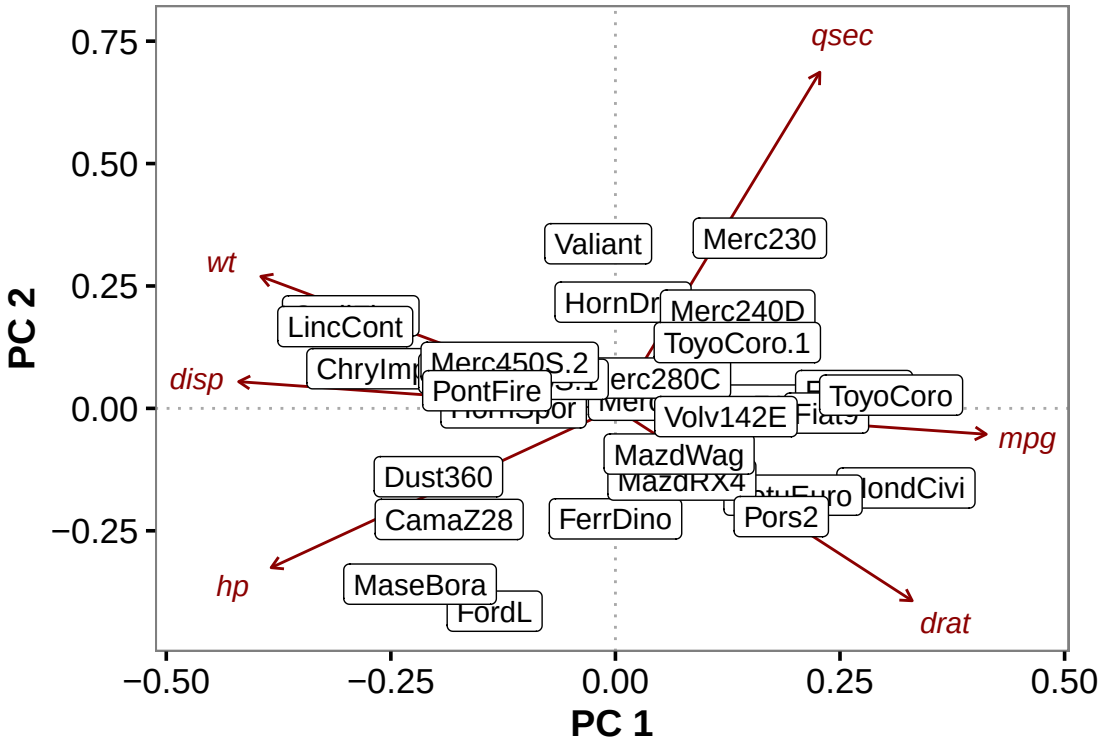
One option: The vegan tutorial provides some solutions to plot labels that give priority to more-important rows (Fig. 3a).

Another option: Using `geom_label` instead of `geom_text` helps with the readability of overlapping labels (Fig. 3b).

```
# (a) Modified from the vegan tutorial
shnam <- make.cepnames(mtcars2$make.model)
 stems <- rowSums(cars.d)
par(mar=c(4,5,0,2))
plot(cars.pca, dis="sites", type="n", las=1)
ordilabel(cars.pca, dis="sites",
          lab=shnam, priority = stems)
text(cars.pca, dis="sp", col="red")
# (b) Using geom_label instead of geom_text
    ggplot() + theme_ord(14) +
      labs(x="PC 1", y="PC 2") +
      geom_vline(xintercept = 0, lty=3, color="darkgrey") +
      geom_hline(yintercept = 0, lty=3, color="darkgrey") +
      geom_segment(data=as_tibble(cars.pca$CA$v),
                   aes(x=0, y=0, xend=PC1*0.90, yend=PC2*0.90),
                   color="darkred",
                   arrow = arrow(length = unit(0.02, "npc")))+
      geom_text(data=as_tibble(cars.pca$CA$v),
                aes(x=PC1, y=PC2,
                    label=row.names(cars.pca$CA$v)),
                fontface="italic", color="darkred")  +
      geom_label(data=as_tibble(cars.pca$CA$u),
                aes(x=PC1, y=PC2,
                    label=make.cepnames(mtcars2$make.model)))
```



(a) Using some tricks from the vegan tutorial.



(b) `ggplot` solution using `geom_label` to help readability of overlapping labels.

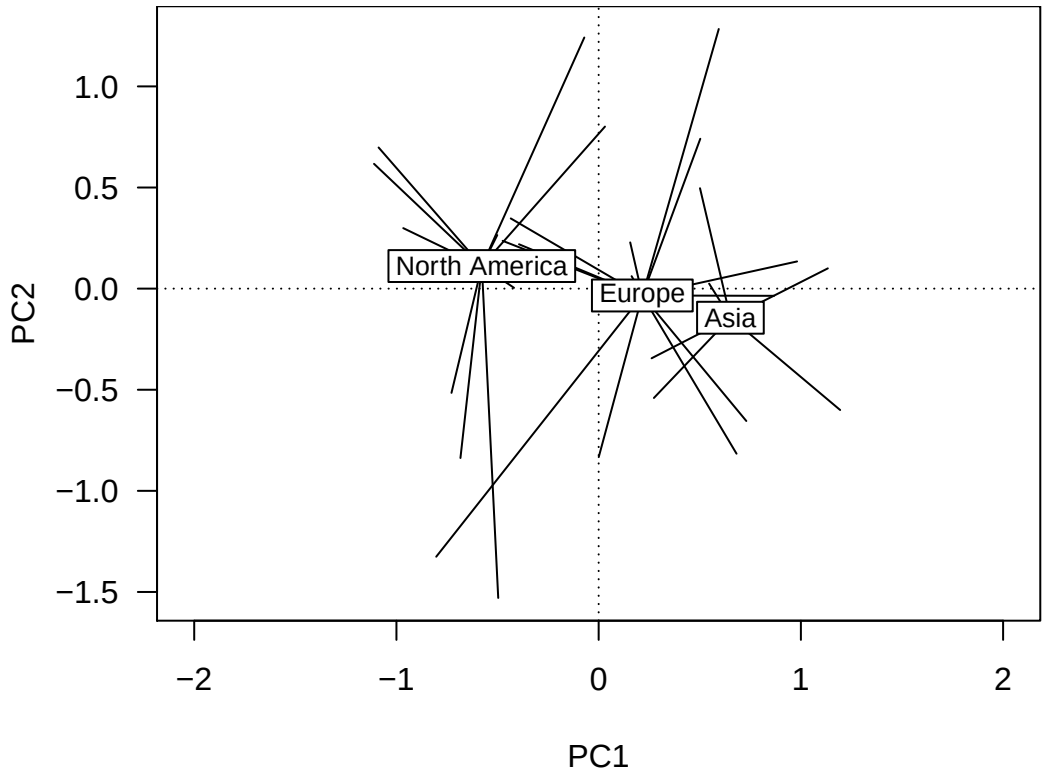
Figure 3: A couple of options for restricting the number of car types plotted.

Environmental variables

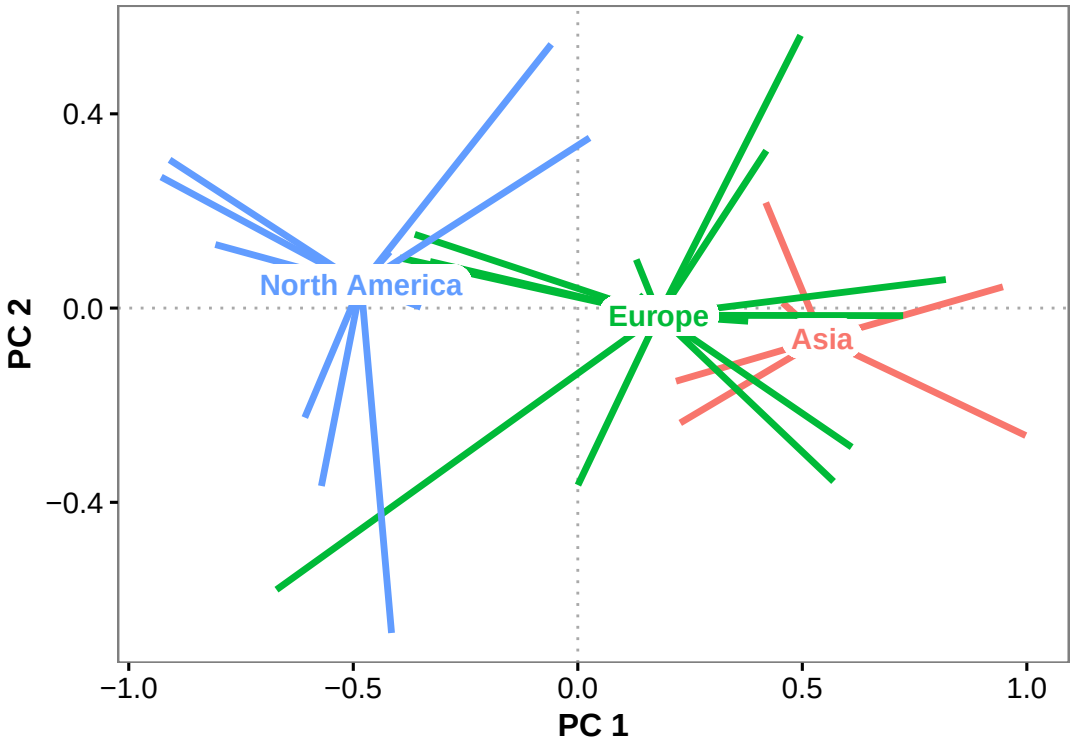
Plot

```
# (a)
par(mar=c(4,5,0,2))
plot(cars.pca, type="n", las=1,
     xlim=c(-1.5,1.5))
ordispider(cars.pca, groups=mtcars2$continent, label=T)
# (b)
# Problem: gg_ordiplot looks nice enough, but can't be customized
# Solution: HACK IT
cont.gg <- gg_ordiplot(cars.pca, groups = mtcars2$continent,
            spiders=TRUE, ellipse=FALSE, plot=FALSE)

cont.gg$df_spiders %>%
  plyr::rename(c("x"="X",
                 "y"="Y") )%>%
  ggplot() + theme_ord(12) +
  geom_vline(xintercept = 0, lty=3, color="darkgrey") +
  geom_hline(yintercept = 0, lty=3, color="darkgrey") +
  labs(x="PC 1", y="PC 2") +
  geom_segment(aes(x=cntr.x, y=cntr.y,
                   xend=X, yend=Y, color=Group),
               size=1.2, show.legend = FALSE) +
  geom_label(aes(x=cntr.x, y=cntr.y,
                 label=Group, color=Group),
             fontface="bold", size=4,
             label.size = 0,
             label.r = unit(0.5, "lines"),
             show.legend = FALSE)
```



(a) Plain old base `plot`.



(b) `ggplot` on the `gg_ordiplot` object.

Figure 4: Two ways to show groups in the PCA.

Test

```
f.fit <- envfit(cars.pca ~ continent, mtcars2)$factors
data.frame(Term=names(f.fit$r),
           R.squared=round(f.fit$r, 2),
           P=f.fit$pvals,
           row.names = NULL)   %>%
        pander("Results of envfit testing clusters by Continent in the PCA.")
```

Table 3: Results of envfit testing clusters by Continent in the PCA.

Term	R.squared	P
continent	0.27	0.001

```
tidy(pairwise.factorfit(cars.pca, mtcars2$continent,
                    nperm = 999, p.method = "fdr") ) %>%
  unite(comparison, c("group1", "group2"), sep = " vs. ", remove=TRUE) %>%
    pander("Results of post-hoc pairwise comparison of Continent clusters in the PCA.")
```

Table 4: Results of post-hoc pairwise comparison of Continent clusters in the PCA.

comparison	p.value
Europe vs. Asia	0.299
North America vs. Asia	0.003
North America vs. Europe	0.0045

Cars group by continent of origin, with significant differences between North American cars and those from both Asia and Europe. Most of this variability lies along the first axis (PC 1).

Here's the script used to make `theme_ord`:

```
# Defining custom theme options
theme_ord <- function (base_size = 12, base_family = "")
{
  theme_grey(base_size = base_size, base_family = base_family) %+replace%
    theme(axis.text = element_text(size = rel(0.9)),
          axis.title = element_text(face="bold"),
          axis.ticks = element_line(colour = "black"),
          strip.text = element_text(face="bold"),
          legend.key = element_rect(colour = "grey80"),
          panel.background = element_rect(fill = "white", colour = NA),
          panel.border = element_rect(fill = NA, colour = "grey50"),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.margin = unit(c(2,2,2,2), "mm"),
          strip.background = element_rect(fill = "lightgreen",
                                          colour = "grey50",
                                          size = 0.2))
}
```