Homework 2. PCA. (60 Points)

Shri Harsha

2023-09-27

Part 1. PCA vs Linear Regression (6 points).

Let's say we have two 'features': let one be x and another y. Recall that in linear regression, we are looking to get a model like:

$$y_i = \beta_0 + \beta_1 * x_i + \varepsilon_i$$

after the fitting, for each data point we would have:

$$y_i = \hat{\beta_0} + \hat{\beta_1} * x_i + r_i$$

where r_i is residual. It can be rewritten as:

$$\hat{\beta_0} + r_i = y_i - \hat{\beta_1} * x_i \qquad (1)$$

The first principal component z_1 calculated on (x, y) is

$$z_{i1} = \phi_{i1} y_i + \phi_{i2} x_i$$

Dividing it by ϕ_{i1}:

$$\frac{z_{i1}}{\phi_{i1}} = y_i + \frac{\phi_{i2}}{\phi_{i1}} x_i \qquad (2)$$

There is a functional resemblance between equations (1) and (2) (described linear relationship between y and x). Is the following true:

$$\hat{\beta_0} + r_i = \frac{z_{i1}}{\phi_{i1}}$$

$$\frac{\phi_{i2}}{\phi_{i1}} = -\hat{\beta_1}$$

Answer: *(just yes or no)* : YES

What is the difference between linear regression coefficients optimization and first PCA calculations?

Answer:

Linear Regression Coefficients Optimization:

Linear regression minimizes the vertical distances between the data points and the hyperplane defined by the regression equation.

The coefficients in linear regression represent the weights assigned to each feature to create a linear combination that best fits the data.

Principal Component Analysis (PCA) Calculations:

PCA seeks to minimize the sum of squared perpendicular distances from the data points to the subspace formed by the principal components.

In PCA, the coefficients represent how much each original feature contributes to each principal component.

(here should be the answer. help yourself with a plot)

Part 2. PCA Exercise (27 points).

In this exercise we will study UK Smoking Data (`smoking.R` , `smoking.rda` or `smoking.csv`):

Description

Survey data on smoking habits from the UK. The data set can be used for analyzing the demographic characteristics of smokers and types of tobacco consumed.

Format

A data frame with 1691 observations on the following 12 variables.

`gender` - Gender with levels Female and Male.

`age` - Age.

`marital_status` - Marital status with levels Divorced, Married, Separated, Single and Widowed.

`highest_qualification` - Highest education level with levels A Levels, Degree, GCSE/CSE, GCSE/O Level, Higher/Sub Degree, No Qualification, ONC/BTEC and Other/Sub Degree

`nationality` - Nationality with levels British, English, Irish, Scottish, Welsh, Other, Refused and Unknown.

`ethnicity` - Ethnicity with levels Asian, Black, Chinese, Mixed, White and Refused Unknown.

`gross_income` - Gross income with levels Under 2,600, 2,600 to 5,200, 5,200 to 10,400, 10,400 to 15,600, 15,600 to 20,800, 20,800 to 28,600, 28,600 to 36,400, Above 36,400, Refused and Unknown.

`region` - Region with levels London, Midlands & East Anglia, Scotland, South East, South West, The North and Wales

`smoke` - Smoking status with levels No and Yes

`amt_weekends` - Number of cigarettes smoked per day on weekends.

`amt_weekdays` - Number of cigarettes smoked per day on weekdays.

`type` - Type of cigarettes smoked with levels Packets, Hand-Rolled, Both/Mainly Packets and Both/Mainly Hand-Rolled

Source National STEM Centre, Large Datasets from stats4schools, https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools (https://www.stem.org.uk/resources/elibrary/resource/28452/large-datasets-stats4schools).

Obtained from https://www.openintro.org/data/index.php?data=smoking (https://www.openintro.org/data/index.php?data=smoking)

Read and Clean the Data

2.1 Read the data from smoking.R or smoking.rda (3 points) > hint: take a look at source or load functions > there is also smoking.csv file for a refference

```
# load libraries
library(tibble)
library(readr)
library(dplyr)
library(data.table)

library(broom)
library(cowplot)

library(ggplot2)
library(ggbiplot)
library(fastDummies)

library(plotly)
```

```
# Load data
smoke_data <- fread("smoking.csv")
```

Take a look into data

```
# place holder
head(smoke_data)
```

gen… <chr>	a.. <int>	marital_status <chr>	highest_qualification <chr>	nationality <chr>	ethnicity <chr>	gross_income <chr>
Male	38	Divorced	No Qualification	British	White	2,600 to 5,200
Female	42	Single	No Qualification	British	White	Under 2,600
Male	40	Married	Degree	English	White	28,600 to 36,40
Female	40	Married	Degree	English	White	10,400 to 15,60
Female	39	Married	GCSE/O Level	British	White	2,600 to 5,200
Female	37	Married	GCSE/O Level	British	White	15,600 to 20,80

6 rows | 1-8 of 12 columns

There are many fields there so for this exercise lets only concentrate on smoke, gender, age, marital_status, highest_qualification and gross_income.

Create new data.frame with only these columns.

```
# place holder
smoke_data_new <- smoke_data[ ,c("smoke", "gender", "age", "marital_status", "highest
_qualification", "gross_income")]

head(smoke_data_new)
```

sm…	gender	a…	marital_status	highest_qualification	gross_income
<chr>	<chr>	<int>	<chr>	<chr>	<chr>
No	Male	38	Divorced	No Qualification	2,600 to 5,200
Yes	Female	42	Single	No Qualification	Under 2,600
No	Male	40	Married	Degree	28,600 to 36,400
No	Female	40	Married	Degree	10,400 to 15,600
No	Female	39	Married	GCSE/O Level	2,600 to 5,200
No	Female	37	Married	GCSE/O Level	15,600 to 20,800

6 rows

2.2 Omit all incomplete records.(3 points)

```
# place holder
smoke_data_new$gross_income[smoke_data_new$gross_income=='Unknown'] <- NA
smoke_data_new$gross_income[smoke_data_new$gross_income=='Refused'] <- NA

smoke_data_new <- na.omit(smoke_data_new)
```

2.3 For PCA feature should be numeric. Some of fields are binary (`gender` and `smoke`) and can easily be converted to numeric type (with one and zero). Other fields like `marital_status` has more than two categories, convert them to binary (e.g. is_married, is_devorced). Several features in the data set are ordinal (`gross_income` and `highest_qualification`), convert them to some king of sensible level (note that levels in factors are not in order). (3 points)

```
# place holder
smoke_data_new$gender <- as.integer(smoke_data_new$gender == 'Male')

smoke_data_new <- dummy_cols(smoke_data_new, select_columns = c( "marital_status", "h
ighest_qualification", "gross_income"))%>%
  select(-marital_status, -highest_qualification, -gross_income)


head(smoke_data_new)
```

sm…	gen…	a..	marital_status_Divorced	marital_status_Married	marital_status_S
<chr>	<int>	<int>	<int>	<int>	
No	1	38	1	0	
Yes	0	42	0	0	
No	1	40	0	1	
No	0	40	0	1	
No	0	39	0	1	
No	0	37	0	1	

6 rows | 1-6 of 24 columns

2.4. Do PCA on all columns except smoking status. (3 points)

```
# place holder
pca_fit <- smoke_data_new %>%
  select(where(is.numeric)) %>% # retain only numeric columns
  prcomp(scale = TRUE)
```

2.5 Make a scree plot (3 points)

```
# place holder
PVE <- tibble(
  PC=1:length(pca_fit$sdev),
  Var=pca_fit$sdev^2,
  PVE=Var/sum(Var),
  CumPVE=cumsum(PVE)
)
PVE
```

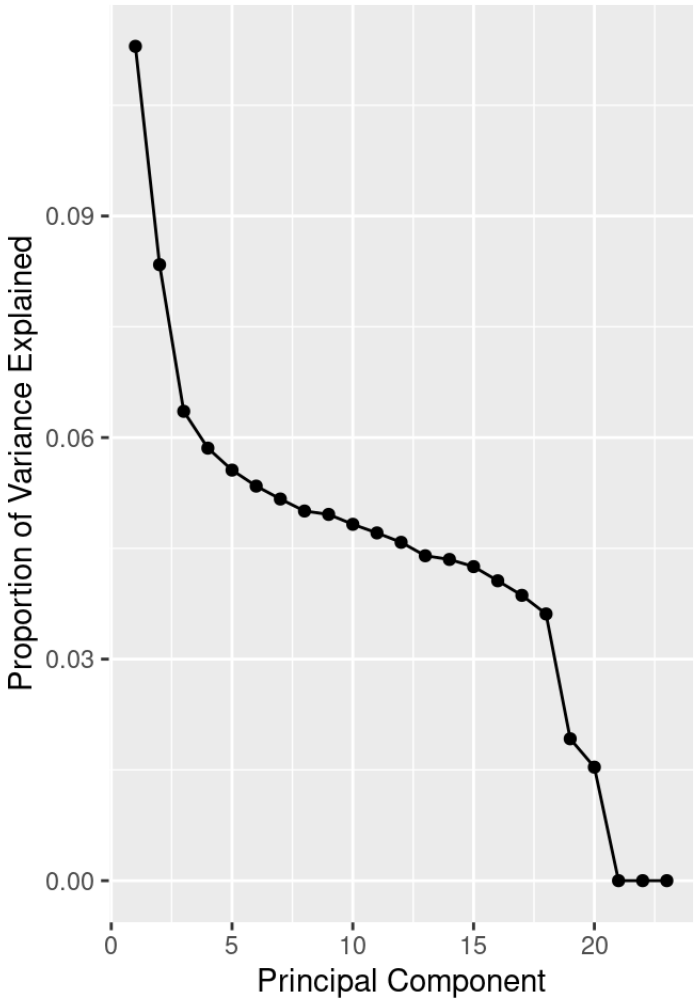
PC	Var	PVE	CumPVE
<int>	<dbl>	<dbl>	<dbl>
1	2.598587e+00	1.129820e-01	0.1129820

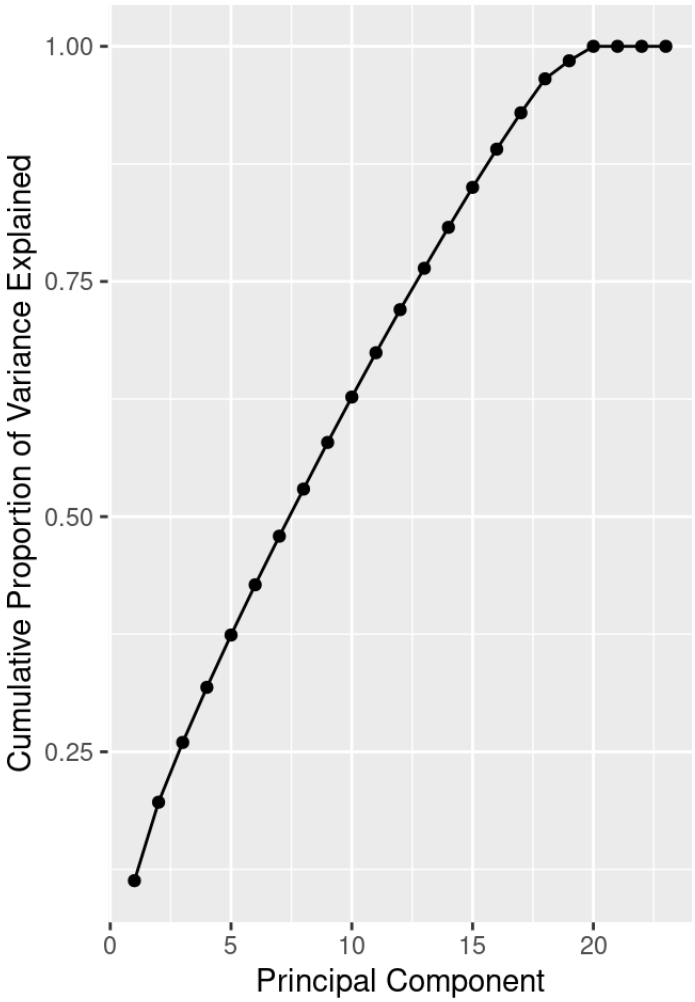
2	1.918515e+00	8.341368e-02	0.1963957
3	1.461777e+00	6.355551e-02	0.2599512
4	1.347095e+00	5.856933e-02	0.3185206
5	1.278676e+00	5.559459e-02	0.3741152
6	1.228877e+00	5.342942e-02	0.4275446
7	1.188660e+00	5.168088e-02	0.4792255
8	1.151251e+00	5.005441e-02	0.5292799
9	1.140671e+00	4.959440e-02	0.5788743
10	1.109874e+00	4.825539e-02	0.6271297

1-10 of 23 rows Previous **1** 2 3 Next

```
cowplot::plot_grid(
qplot(data=PVE,x=PC,y=PVE,geom=c("point","line"),
      xlab = "Principal Component",
      ylab = "Proportion of Variance Explained"),
qplot(data=PVE,x=PC,y=CumPVE,geom=c("point","line"),
      xlab = "Principal Component",
      ylab = "Cumulative Proportion of Variance Explained")
)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



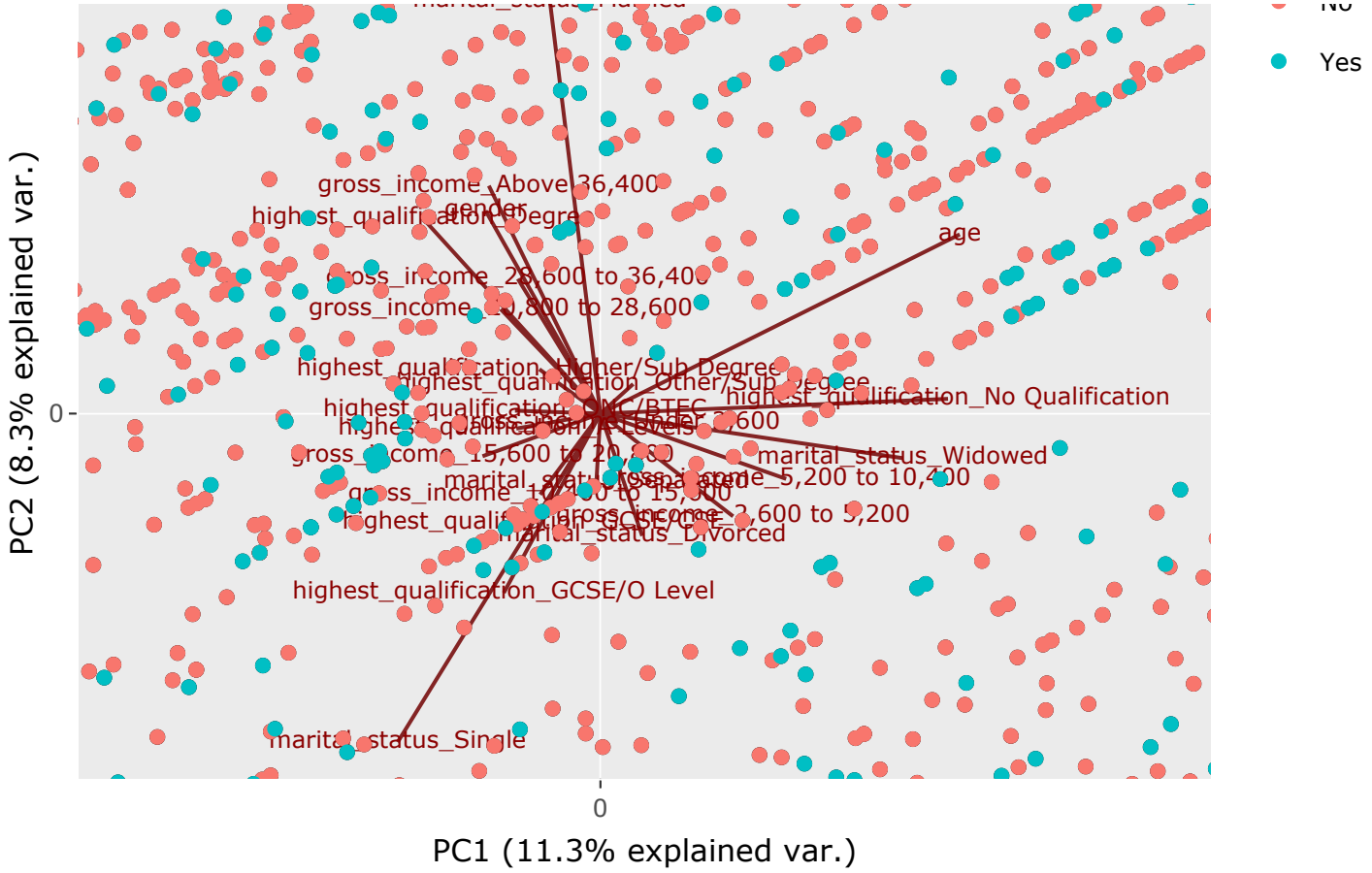


Comment on the shape, if you need to reduce dimensions home many would you choose

```
After first two principal components there is a sharpe decrease in the varaince expla
ined by the next set of principal components. And the last three principal components
are explaining nothing showing those there are highly correlated with other principal
components.

The shape of the curve is still elbow for scree plot. If we draw cummulative varaince
exlained as function of principal components we can see a linear increase in the expl
ainability later got flattened out.

If I have to reduce the dimensions I will take 19 Principal Components.
```

2.6 Make a biplot color points by smoking field. (3 points)

```
# place holder
smoke <- smoke_data_new$smoke
ggplotly(ggbiplot(pca_fit, scale = 0,labels = pca_fit$x %>% rownames()) + geom_point(
aes(color = smoke)))
```



Comment on observed biplot.

```
We can clearly see that Qualification status has more effect on 1st principal compone
nt and Martial status has more effect on the 2nd principal component.
```

Can we use first two PC to discriminate smoking?

```
No. Because even after plotting points in frist two principal components we can see t
hat data is cloudy no real seperation between smoking and not smoking.
```

2.7 Based on the loading vector can we name PC with some descriptive name? (3 points)

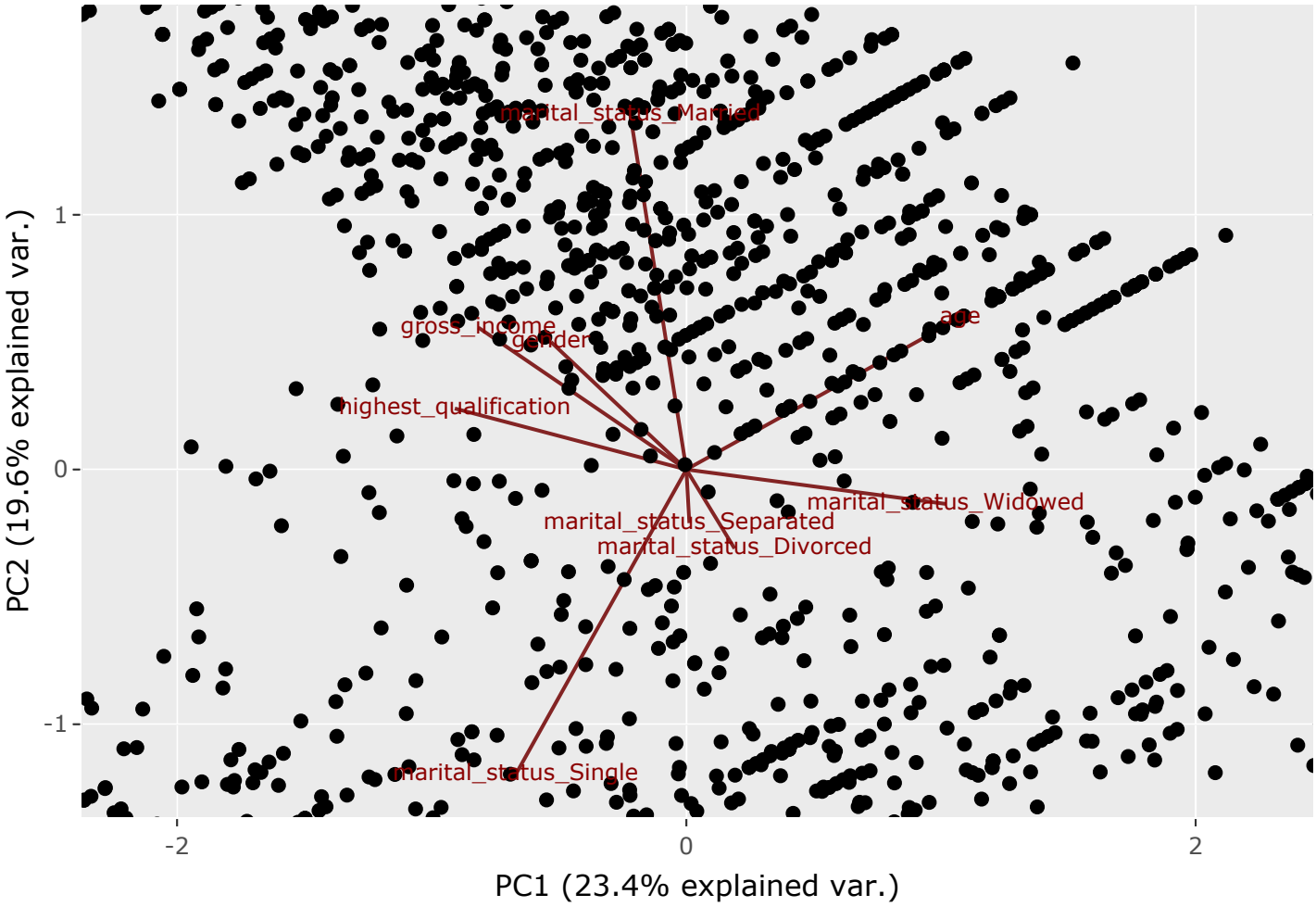
```
1st Principal Component - Education qualification.
2nd Principal Component - Martial Status.
```

2.8 May be some of splits between categories or mapping to numerics should be revisited, if so what will you do differently? (3 points)

```
May be we can change the encoding of highest_qualification and gross_income from dumm
y varibles to ordinal encoding.
```

2.9 Follow your suggestion in 2.10 and redo PCA and biplot (3 points)

```
smoke_data_alter <- smoke_data[ ,c("smoke", "gender", "age", "marital_status", "highe
st_qualification", "gross_income")]

smoke_data_alter$gross_income[smoke_data_alter$gross_income=='Unknown'] <- NA
smoke_data_alter$gross_income[smoke_data_alter$gross_income=='Refused'] <- NA

smoke_data_alter <- na.omit(smoke_data_alter)
smoke_data_alter$gender <- as.integer(smoke_data_alter$gender == 'Male')

# Define the ordinal mapping
qualification_ordinal_mapping <- c("No Qualification" = 1,
                      "GCSE/CSE" = 2,
                      "GCSE/O Level" = 3,
                      "ONC/BTEC" = 4,
                      "A Levels" = 5,
                      "Other/Sub Degree" = 6,
                      "Degree" = 7,
                      "Higher/Sub Degree" = 8)

# Perform ordinal encoding
smoke_data_alter$highest_qualification <- as.integer(factor(smoke_data_alter$highest_
qualification, levels = names(qualification_ordinal_mapping), ordered = TRUE))

# Define the ordinal mapping
income_ordinal_mapping <- c("Under 2,600" = 1,
                      "2,600 to 5,200" = 2,
                      "5,200 to 10,400" = 3,
                      "10,400 to 15,600" = 4,
                      "15,600 to 20,800" = 5,
                      "20,800 to 28,600" = 6,
                      "28,600 to 36,400" = 7,
                      "Above 36,400" = 8)

# Perform ordinal encoding
smoke_data_alter$gross_income <- as.integer(factor(smoke_data_alter$gross_income, lev
els = names(income_ordinal_mapping), ordered = TRUE))



smoke_data_alter <- dummy_cols(smoke_data_alter, select_columns = c( "marital_status"
))%>%
  select(-marital_status)

pca_fit <- smoke_data_alter %>%
  select(where(is.numeric)) %>% # retain only numeric columns
  prcomp(scale = TRUE)

ggplotly(ggbiplot(pca_fit, scale = 0,labels = pca_fit$x %>% rownames()))
```



Part 3. Freestyle. (27 points).

Get the data set from your final project (or find something suitable). The data set should have at least four variables and it shouldn't be used in class PCA examples: iris, mpg, diamonds and so on).

- Convert a columns to proper format (9 points)
- Perform PCA (3 points)
- Make a skree plot (3 points)
- Make a biplot (3 points)
- Discuss your observations (9 points)

```
whl_df <- read_csv("Wholesale customers data.csv")
```

```
## Rows: 440 Columns: 8
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## dbl (8): Channel, Region, Fresh, Milk, Grocery, Frozen, Detergents_Paper, De...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(whl_df)
```

Channel <dbl>	Region <dbl>	Fresh <dbl>	Milk <dbl>	Grocery <dbl>	Frozen <dbl>	Detergents_Paper <dbl>	Delicassen <dbl>
2	3	12669	9656	7561	214	2674	1338
2	3	7057	9810	9568	1762	3293	1776
2	3	6353	8808	7684	2405	3516	7844
1	3	13265	1196	4221	6404	507	1788
2	3	22615	5410	7198	3915	1777	5185
2	3	9413	8259	5126	666	1795	1451

6 rows

```
sum(is.na(whl_df))
```

```
## [1] 0
```

```
whl_pca_fit <- whl_df %>%
  select(where(is.numeric)) %>% # retain only numeric columns
  prcomp(scale = TRUE)
whl_pca_fit
```

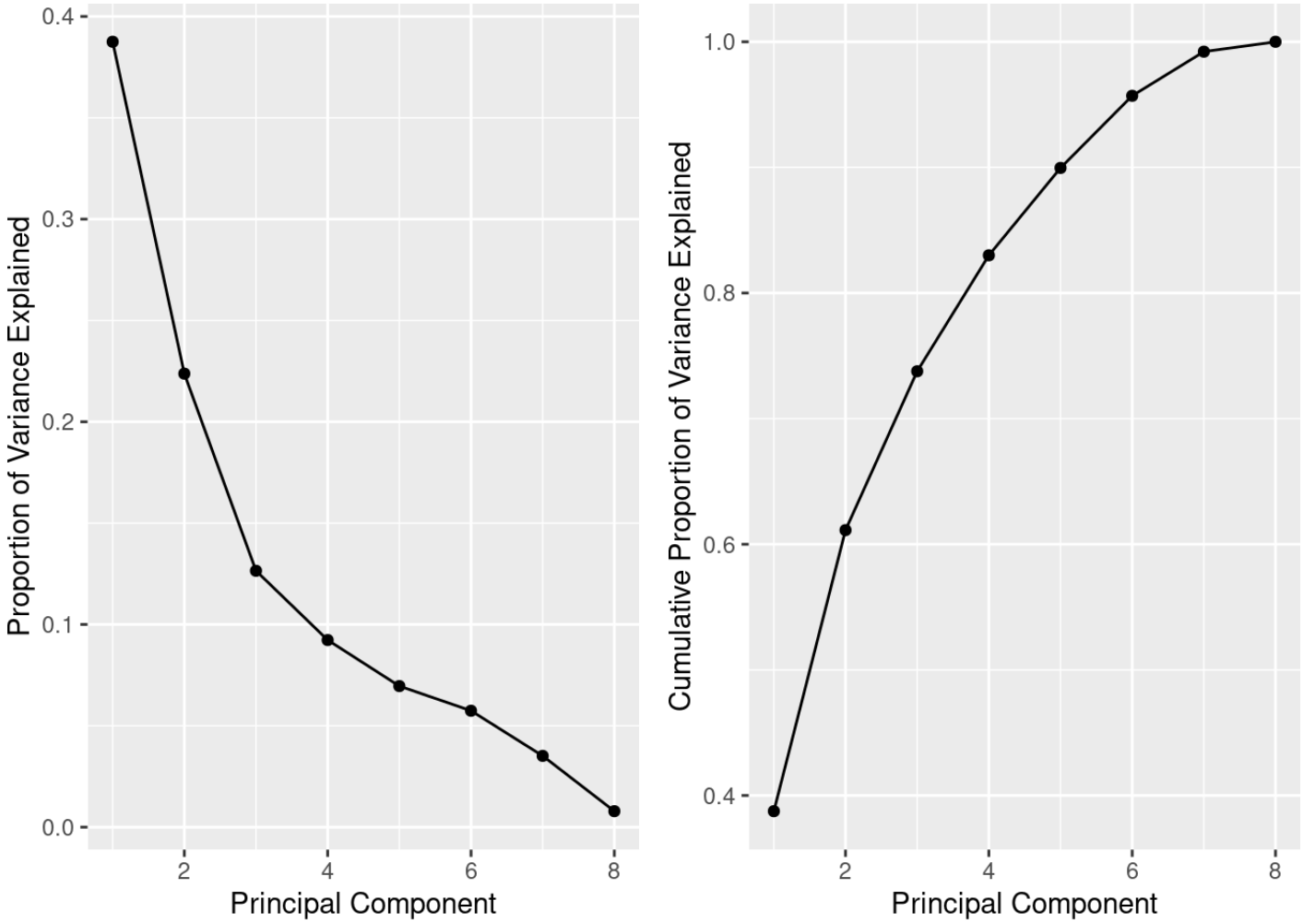
```
## Standard deviations (1, .., p=8):
## [1] 1.7606845 1.3378965 1.0058697 0.8592976 0.7460780 0.6777229 0.5302132
## [8] 0.2505796
##
## Rotation (n x k) = (8 x 8):
##                          PC1         PC2           PC3         PC4          PC5
## Channel          -0.42829156  0.20469886  0.0829798863 -0.02964416  0.03620585
## Region           -0.02472603 -0.04312964  0.9825008891 -0.07784462 -0.13250892
## Fresh             0.02531946 -0.51344468  0.0889509074  0.79847592  0.25811686
## Milk             -0.47440995 -0.20554061 -0.0257510842 -0.05402202  0.07208576
## Grocery          -0.53632914  0.00871762 -0.0453143572  0.12158624 -0.11172990
## Frozen            0.02997456 -0.59274525 -0.1221565222 -0.16131688 -0.75421244
## Detergents_Paper -0.52390630  0.12108309 -0.0474814388  0.15101211 -0.17650264
## Delicassen       -0.16499653 -0.53318082  0.0009301994 -0.53755767  0.54482721
##                          PC6          PC7          PC8
## Channel          -0.86350670  0.139899044  0.019335373
## Region            0.08976479 -0.023279938 -0.001545045
## Fresh            -0.14747474 -0.027173693 -0.033851114
## Milk              0.31593256  0.789020414 -0.039291347
## Grocery           0.21369889 -0.353064294  0.715984124
## Frozen           -0.19435993 -0.005336793 -0.012983225
## Detergents_Paper  0.19575356 -0.371374310 -0.691672189
## Delicassen       -0.05453289 -0.306582655 -0.075642587
```

```
PVE <- tibble(
  PC=1:length(whl_pca_fit$sdev),
  Var=whl_pca_fit$sdev^2,
  PVE=Var/sum(Var),
  CumPVE=cumsum(PVE)
)
PVE
```

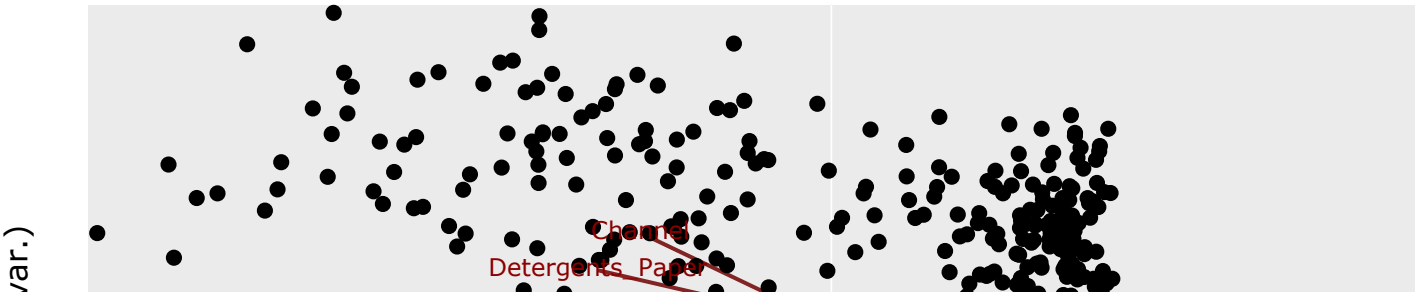
PC <int>	Var <dbl>	PVE <dbl>	CumPVE <dbl>
1	3.10000983	0.387501229	0.3875012
2	1.78996704	0.223745880	0.6112471
3	1.01177388	0.126471735	0.7377188
4	0.73839230	0.092299037	0.8300179
5	0.55663240	0.069579050	0.8995969
6	0.45930835	0.057413544	0.9570105
7	0.28112605	0.035140757	0.9921512
8	0.06279015	0.007848769	1.0000000

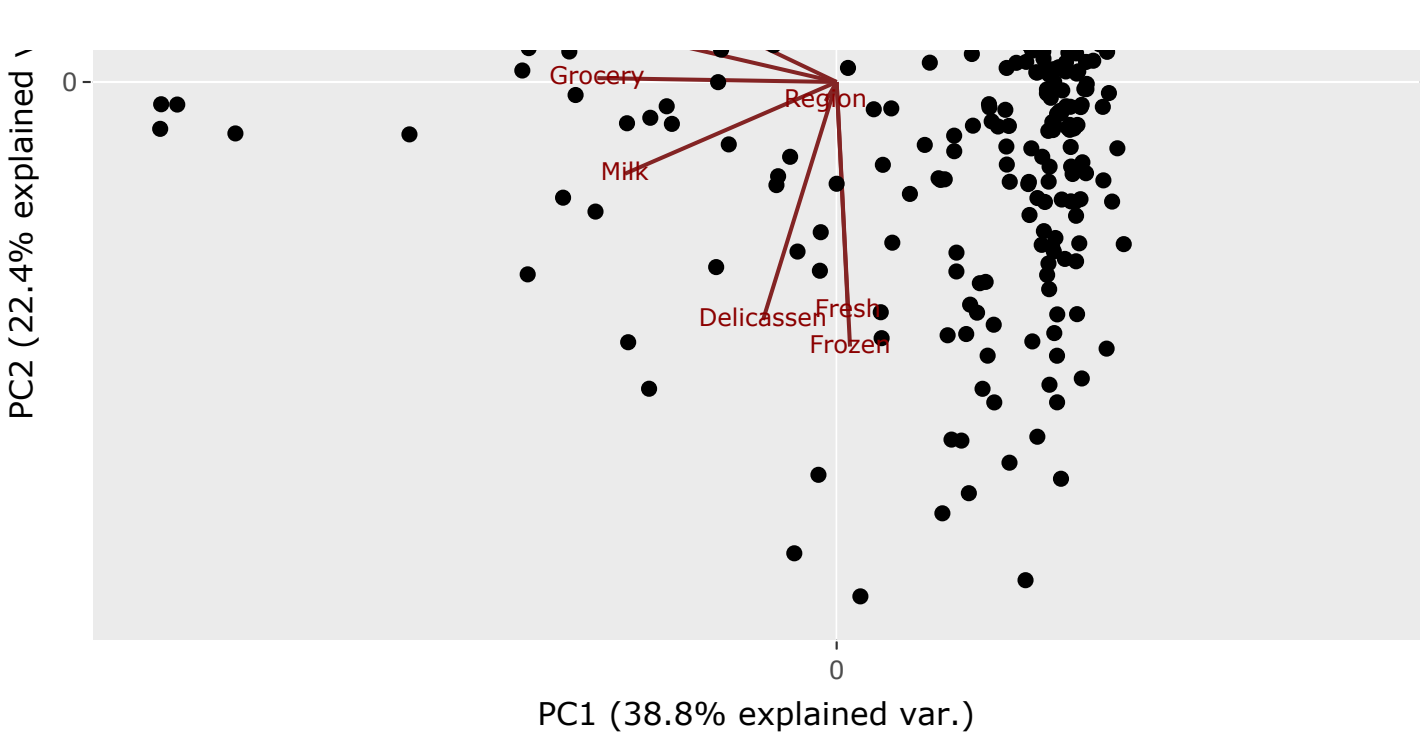
8 rows

```
cowplot::plot_grid(
qplot(data=PVE,x=PC,y=PVE,geom=c("point","line"),
      xlab = "Principal Component",
      ylab = "Proportion of Variance Explained"),
qplot(data=PVE,x=PC,y=CumPVE,geom=c("point","line"),
      xlab = "Principal Component",
      ylab = "Cumulative Proportion of Variance Explained")
)
```




```
ggplotly(ggbiplot(whl_pca_fit, scale = 0,labels = whl_pca_fit$x %>% rownames()))
```





Observations:

1. First two principal components expalain more than 60% of variance in the data.
2. First 5 principal components expalin almost 90% of variance.
3. Upon examining the biplot, it becomes evident that the type of product, such as whether it falls into categories like Grocery, Milk, or Detergent, significantly influences the first principal component. On the other hand, the characteristics of the product itself, such as whether it is categorized as Fresh or Frozen, notably impact the second principal component.