Double-click (or enter) to edit

BI Lab 4

Task1: Descriptive Analysis, Unsupervised Learning – IKEA

Task: to discover some new places here in Sweden that may be suitable for IKEA department stores.

```
We will do this by using the k-means method on a text file,
*ikea_data.txt*, which contains important features for many of Sweden's municipalities.
```

PCA (Principal Components Analysis) is particularly handy when working with big data sets, where many variables are present. Because as such you cannot easily plot the data in its raw format, PCA allows you to see the overall shape of the data, identifying similaritites and differences between groups of samples.

In a nutshell,
source

We take a dataset with many variables, and you simplify that dataset by turning your original variables into a smaller number of "Principal Components".

Principal Components are the underlying structure in the data. They are the directions where there is the most variance, the directions where the data is most spread out. This means that we try to find the straight line that best spreads the data out when it is projected along it. This is the **first principal component**, the straight line that shows the most substantial variance in the data.

Eigenvectors, and **eigenvalues** come in pairs: every eigenvector has a corresponding eigenvalue. An eigenvector is basically a direction, such as "vertical" or "45 degrees", while an eigenvalue is a number telling you how much variance there is in the data in that direction. The *eigenvector with the highest eigenvalue* is the first principal component*.

```
install.packages("data.table")
install.packages("psych")
install.packages("dplyr")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘tmvnsim’, ‘mnormt’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(data.table)
```

```
# LOAD DATA ################################################
# creating a dataframe from the pricing dataset
#data_pricing <- read.csv("home_assignment_data_pricing.csv")
ikea <- fread("ikea_data.txt")
#data_pricing <- fread("home_assignment_data_pricing.csv")
```

```
ikea_df <- as.data.frame(ikea)
```

```
head(ikea,3)
```

A data.table: 3 × 12

Kommun_code	Year	Kommun_name	Revenue	Employee	Population	Population_University	Percent_University	Productivit

Explore the dataframe

1. Clean the dataset/Check data types.

1780 2010 Karlstad 4560 1910 85753 13308 0.15518991 119.647

```
# EXPLORE DATA ###################################################
# Using head or tail to look at part of it, 6 first lines by default
# here the first 3 lines

head(ikea,3)
#tail(data_pricing,3)
```

A data.table: 3 × 12

Kommun_code	Year	Kommun_name	Revenue	Employee	Population	Population_University	Percent_University	Productivit
<int>	<int>	<chr>	<int>	<int>	<int>	<int>	<dbl>	<dbl
2583	2010	Haparanda	1078	276	10059	719	0.07147828	195.741
880	2010	Kalmar	3790	1621	62815	8716	0.13875667	117.173
1780	2010	Karlstad	4560	1910	85753	13308	0.15518991	119.647

```
# Check the dimension, nrow x ncol
# 2612522 rows and 14 columns
dim(ikea)
```

207 · 12

```
summary(ikea)
```

```
 Kommun_code          Year      Kommun_name           Revenue
Min.   : 114.0   Min.   :2010   Length:207         Min.   :   11.0
1st Qu.: 582.5   1st Qu.:2010   Class :character   1st Qu.:  110.0
Median :1263.0   Median :2010   Mode  :character   Median :  252.0
Mean   :1049.7   Mean   :2010                      Mean   : 1031.8
3rd Qu.:1461.5   3rd Qu.:2010                      3rd Qu.:  825.5
Max.   :2583.0   Max.   :2010                      Max.   :32897.0
   Employee         Population     Population_University Percent_University
Min.   :    2.0   Min.   :  3672   Min.   :   174.0      Min.   :0.04614
1st Qu.:   64.5   1st Qu.: 10786   1st Qu.:   788.5      1st Qu.:0.06902
Median :  142.0   Median : 16515   Median :  1598.0      Median :0.08660
Mean   :  523.0   Mean   : 34543   Mean   :  4660.1      Mean   :0.09746
3rd Qu.:  435.0   3rd Qu.: 37922   3rd Qu.:  4073.5      3rd Qu.:0.11208
Max.   :18795.0   Max.   :847073   Max.   :191585.0      Max.   :0.26965
 Productivity      SalesIndex          Infrast            Border
Min.   : 19.90   Min.   :   1.133   Min.   :0.00000   Min.   :0.00000
1st Qu.: 74.70   1st Qu.:  11.334   1st Qu.:0.00000   1st Qu.:0.00000
Median : 90.46   Median :  25.966   Median :0.00000   Median :0.00000
Mean   : 91.45   Mean   : 106.316   Mean   :0.04831   Mean   :0.03865
3rd Qu.:104.01   3rd Qu.:  85.058   3rd Qu.:0.00000   3rd Qu.:0.00000
```

```
#clean data from NAs if any
ikea_clean = na.omit(ikea)


library(psych)

describe(ikea)
```

A psych: 12 × 13

	vars	n	mean	sd	median	trimmed	mad	min	
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
Kommun_code	1	207	1.049734e+03	5.410655e+02	1.263000e+03	1.061874e+03	6.360354e+02	1.140000e+02	2
Year	2	207	2.010000e+03	0.000000e+00	2.010000e+03	2.010000e+03	0.000000e+00	2.010000e+03	2
Kommun_name*	3	207	1.040000e+02	5.989992e+01	1.040000e+02	1.040000e+02	7.709520e+01	1.000000e+00	2
Revenue	4	207	1.031802e+03	2.694890e+03	2.520000e+02	5.187365e+02	2.920722e+02	1.100000e+01	3
Employee	5	207	5.230386e+02	1.463194e+03	1.420000e+02	2.737425e+02	1.541904e+02	2.000000e+00	1

```
ikea_clean = as.data.frame(gsub("[[:punct:]]", "", as.matrix(ikea)))

tail(ikea_clean)
```

```
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
ERROR while rich displaying an object: Error in gsub(chr, html_specials[[chr]], text, fixed = TRUE): input string 3 is

Traceback:
1. FUN(X[[i]], ...)
2. tryCatch(withCallingHandlers({
 .     if (!mime %in% names(repr::mime2repr))
 .         stop("No repr_* for mimetype ", mime, " in repr::mime2repr")
 .     rpr <- repr::mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler), error = outer_handler)
3. tryCatchList(expr, classes, parentenv, handlers)
4. tryCatchOne(expr, names, parentenv, handlers[[1L]])
5. doTryCatch(return(expr), name, parentenv, handler)
6. withCallingHandlers({
 .     if (!mime %in% names(repr::mime2repr))
 .         stop("No repr_* for mimetype ", mime, " in repr::mime2repr")
 .     rpr <- repr::mime2repr[[mime]](obj)
 .     if (is.null(rpr))
 .         return(NULL)
 .     prepare_content(is.raw(rpr), rpr)
 . }, error = error_handler)
7. repr::mime2repr[[mime]](obj)
8. repr_markdown.data.frame(obj)
9. repr_matrix_generic(obj, "\n%s\n\n%s%s\n", sprintf("|%%s\n|%s|\n",
 .     underline), NULL, " <!--/--> |", " %s |", "%s", "|%s\n",
 .     " %s |", " %s |", escape_fun = markdown_escape, rows = rows,
 .     cols = cols, ...)
10. lapply(seq_len(nrow(x)), function(r) {
  .     row <- escape_fun(slice_row(x, r))
```

```
    .     cells <- sprintf(cell, row)
    .     if (has_rownames) {
    .         row_head <- sprintf(row_head, escape_fun(rownames(x)[[r]]))
    .         cells <- c(row_head, cells)
    .     }
    .     sprintf(row_wrap, paste(cells, collapse = ""))
    . })
11. FUN(X[[i]], ...)
12. escape_fun(slice_row(x, r))
13. html_escape(values, do_spaces = FALSE)
14. gsub(chr, html_specials[[chr]], text, fixed = TRUE)
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
“input string 3 is invalid in this locale”
Warning message in FUN(X[[i]], ...):
```

```
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
Warning message in FUN(X[[i]], ...):
"input string 3 is invalid in this locale"
```

```
head(ikea_clean,3)
```

A data.frame: 3 × 12

	Kommun_code	Year	Kommun_name	Revenue	Employee	Population	Population_University	Percent_University	Producti
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<
1	2583	2010	Haparanda	1078	276	10059	719	007147828	1957
2	880	2010	Kalmar	3790	1621	62815	8716	013875667	1171
3	1780	2010	Karlstad	4560	1910	85753	13308	015518991	1196
206	1881	2010	Kumla	152	98	20456	1598	007811889	7

```
#save in a csv file
write.csv(ikea_clean, "ikea_clean.csv")
```

Double-click (or enter) to edit

2. Perform PCA for dimensionality reduction (explain 90-95% of total variance)

PCA is a type of **linear transformation** on a given data set that has values for a certain number of variables (coordinates) for a certain amount of spaces. This linear transformation fits this dataset to a new coordinate system in such a way that the most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance. In this way, you transform a set of x correlated variables over y samples to a set of p uncorrelated principal components over the same samples.

```
# convert specific columns with numbers into int type
#samp <- c("Revenue", "Employee", "Population", "Population_University", "Percent_University", "Productivity", "SalesIndex
#ikea_clean <- as.numeric(ikea_clean[, samp])

ikea_clean$Revenue <- as.numeric(ikea_clean$Revenue)
ikea_clean$Employee <- as.numeric(ikea_clean$Employee)
ikea_clean$Population <- as.numeric(ikea_clean$Population)
ikea_clean$Population_University <- as.numeric(ikea_clean$Population_University)
ikea_clean$Percent_University <- as.numeric(ikea_clean$Percent_University)
ikea_clean$Productivity <- as.numeric(ikea_clean$Productivity)
ikea_clean$SalesIndex <- as.numeric(ikea_clean$SalesIndex)


tail(ikea_clean, 3)
```

```
..):
is locale”
..):
is locale”
..):
is locale”
bject: Error in gsub(chr, html_specials[[chr]], text, fixed = TRUE): input string 3 is invalid in this locale



{
::mime2repr))
imetype ", mime, " in repr::mime2repr")
ime]](obj)


r), rpr)
ror = outer_handler)
arentenv, handlers)
ntenv, handlers[[1L]])
, parentenv, handler)

::mime2repr))
imetype ", mime, " in repr::mime2repr")
ime]](obj)


r), rpr)



s\n\n%s%s\n", sprintf("|%%s\n|%s|\n",
--> |", " %s |", "%s", "|%s\n",
fun = markdown_escape, rows = rows,

ction(r) {
row(x, r))
ow)

row_head, escape_fun(rownames(x)[[r]]))
 cells)

cells, collapse = ""))
```

```
s = FALSE)
]], text, fixed = TRUE)
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
..):
is locale”
```

Let's replace indexes with Kommun_name instead

```
#rownames(ikea_clean) <- ikea_clean$Kommun_name
```

```
de categorical data / textual
et the dataset keeping only numerical data

 c("Revenue",  "Employee", "Population", "Population_University", "Percent_University", "Productivity", "SalesIndex")
ea_clean[, samp]
```

```
lean.pca <- prcomp(ikea_clean[,c(4:10)], center = TRUE,scale. = TRUE)
```

```
- prcomp(z, center = TRUE,scale. = TRUE)


summary(ikea_clean.pca)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6       PC7
Standard deviation     2.278 0.9951 0.8662 0.23238 0.11852 0.06641 7.078e-10
Proportion of Variance 0.741 0.1415 0.1072 0.00771 0.00201 0.00063 0.000e+00
Cumulative Proportion  0.741 0.8825 0.9897 0.99736 0.99937 1.00000 1.000e+00
```

We obtain 7 principal components called PC1-7. Each of these explains a percentage of the total variation in the dataset.
PC1 explains 74% of the total variance, which means that nearly three-fourths of the information in this dataset subset made of 7 variables can be encapsulated by just that one Principal Component. PC2 explains 14% of the variance. So, by knowing the position of a sample in relation to just PC1 and PC2, we can get a very accurate view on where the sample stands in relation to other samples, considering that PC1 and PC2 alone can explain 88% of the variance.

```
# Let's have a look at our PCA object with a str() call
str(z.pca)
```

```
List of 5
 $ sdev    : num [1:7] 2.278 0.995 0.866 0.232 0.119 ...
 $ rotation: num [1:7, 1:7] -0.435 -0.434 -0.434 -0.43 -0.238 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:7] "Revenue" "Employee" "Population" "Population_University" ...
  .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
 $ center  : Named num [1:7] 1032 523 34543 4660 9746109 ...
  ..- attr(*, "names")= chr [1:7] "Revenue" "Employee" "Population" "Population_University" ...
 $ scale   : Named num [1:7] 2695 1463 66882 14403 4075996 ...
  ..- attr(*, "names")= chr [1:7] "Revenue" "Employee" "Population" "Population_University" ...
 $ x       : num [1:207, 1:7] 0.256 -1.818 -2.539 -0.285 0.291 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:207] "Haparanda" "Kalmar" "Karlstad" "Upplands V\xe4sby" ...
  .. ..$ : chr [1:7] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

Our PCA object contains the following information:

- The center point (*center*), scaling (*scale*), standard deviation(sdev) of each principal component
- The relationship (correlation or anticorrelation, etc) between the initial variables and the principal components (*rotation*)
- The values of each sample in terms of the principal components (*x*)

```
head(z,3)
```

A matrix: 3 × 7 of type dbl

	Revenue	Employee	Population	Population_University	Percent_University	Productivity	SalesIndex
Haparanda	0.01714284	-0.1688351	-0.3660808	-0.2736215	-0.6374592	3.7297749	0.01714284
Kalmar	1.02349206	0.7503865	0.4227056	0.2815909	1.0131408	0.9198444	1.02349206
Karlstad	1.30921806	0.9478996	0.7656652	0.6004024	1.4163119	1.0083356	1.30921806

Normalize or scale data

```
library(dplyr)
```

```
Attaching package: ‘dplyr’


The following objects are masked from ‘package:plyr’:

    arrange, count, desc, failwith, id, mutate, rename, summarise,
    summarize


The following objects are masked from ‘package:data.table’:

    between, first, last


The following objects are masked from ‘package:stats’:
```

```
    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

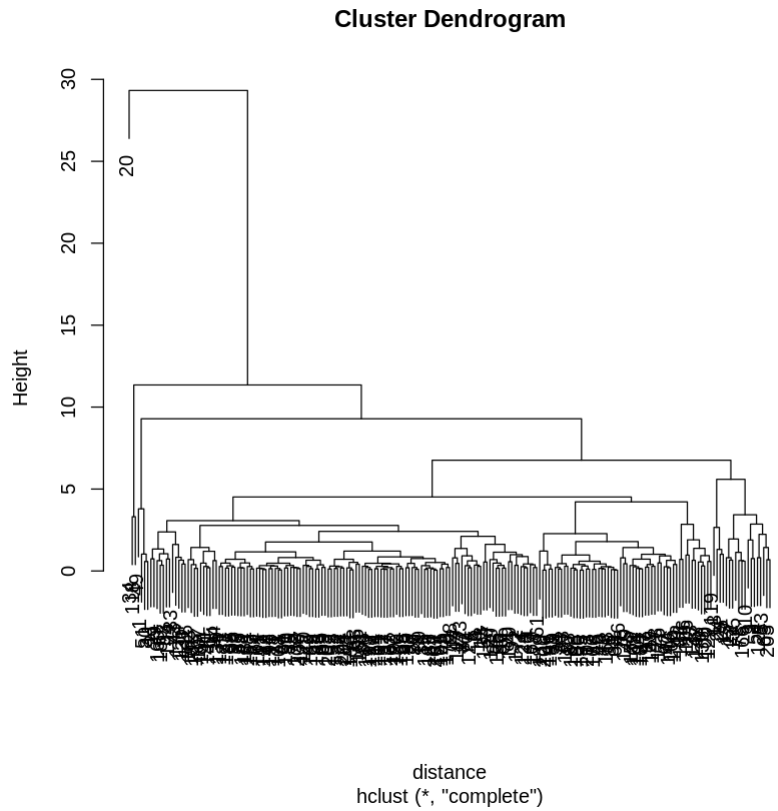
```
dim(z)
```

207 · 7

```
# Normalization
m <- apply(z, 2, mean)

s <- apply(z, 2, sd)

z <- scale(z, m, s)

# Calculate the Euclidean distance
distance <- dist(z)


# plot dendrogram

install.packages("data.tree")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(data.tree)

#png(file="dendogram2.png",
#width=900, height=600)

hc.l <- hclust(distance)
plot(hc.l)
```

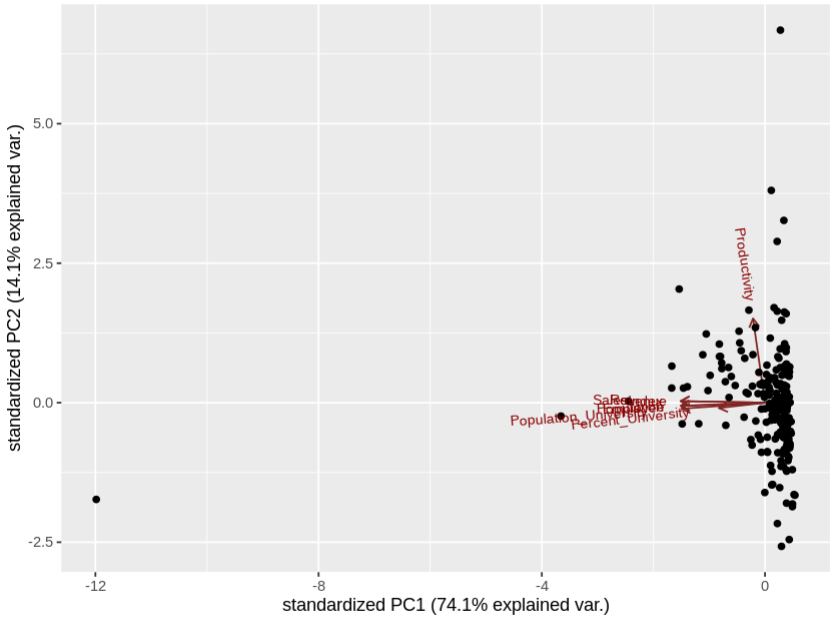



```
# Plotting PCA with ggbiplot
library(devtools)
install_github("vqv/ggbiplot")
```

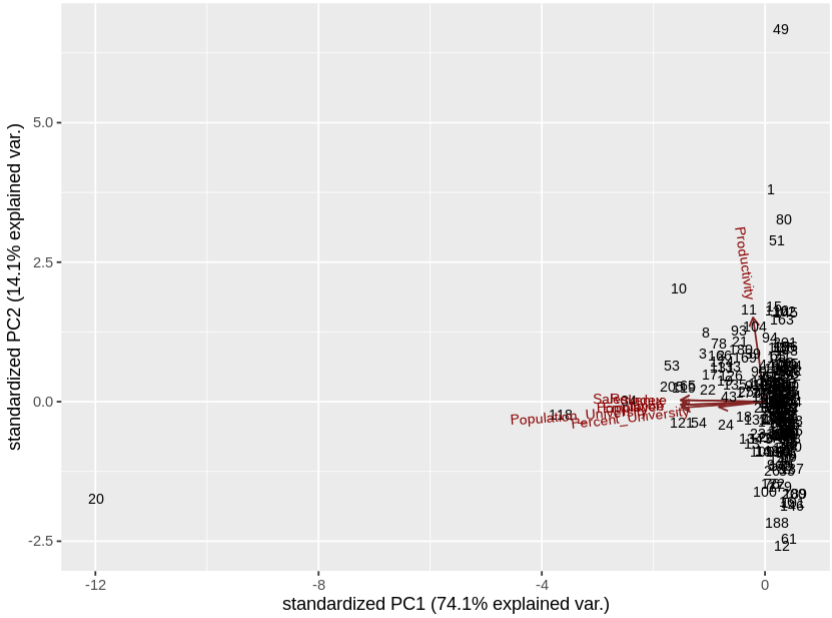
```
     Loading required package: usethis

     Downloading GitHub repo vqv/ggbiplot@HEAD

     plyr (NA -> 1.8.6) [CRAN]
     Installing 1 packages: plyr

     Installing package into ‘/usr/local/lib/R/site-library’
     (as ‘lib’ is unspecified)

     ✓  checking for file ‘/tmp/Rtmpgh6kUa/remotes3b4b782a1f/vqv-ggbiplot-7325e88/DESCRIPTION’
```

```
−  preparing ‘ggbiplot’:
✓  checking DESCRIPTION meta-information
−  checking for LF line-endings in source and make files and shell scripts
−  checking for empty or unneeded directories
−  looking to see if a ‘data/datalist’ file should be added
−  building ‘ggbiplot_0.55.tar.gz’

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(ggbiplot)

ggbiplot(z.pca)
```




```
ggbiplot(z.pca, labels=rownames(ikea_clean))
```



We can see that the variables Population, Percent_university, SalesIndexes contribute the least to PC1 (because placed in the lowest quadrant at bottom left) -- as opposed to the influence thatt variables in the upper right quadrant would have had with higher values in those variables moving the samples to the right on this plot.
This lets you see how the data points relate to the axes, but the plot is not very informative without knowing which point corresponds to which sample.

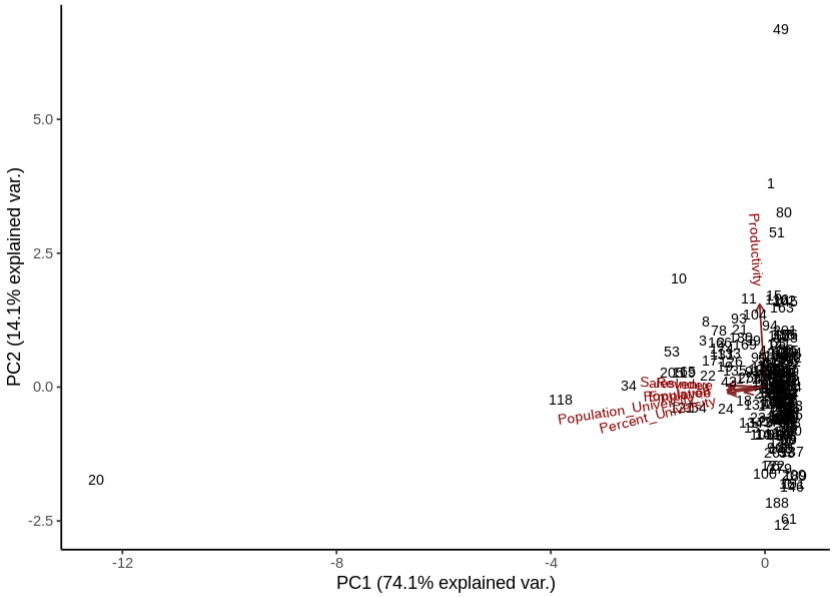
Thus what represents the data with index 20 for eg.?

```
ikea_clean[20,]

# Stockholm
```

A data.frame: 1 × 12

	Kommun_code	Year	Kommun_name	Revenue	Employee	Population	Population_University	Percent_University	Product
	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
20	180	2010	Stockholm	32897	18795	847073	191585	22617295	87

```
ggbiplot(ikea_clean.pca,ellipse=TRUE,obs.scale = 0.05, var.scale = 0.05,  labels=rownames(ikea_clean))+
  theme_classic()
```



3. Apply Elbow method to select optimum number of clusters - would turn out to be 3

```
install.packages("purrr")
```

```
library(purrr)
```

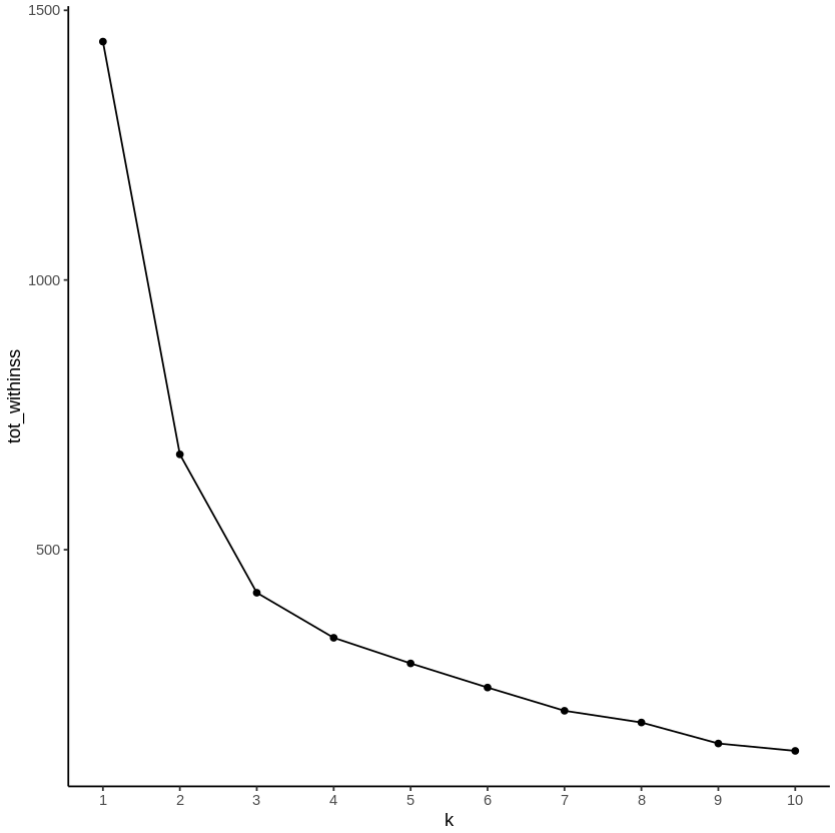
```
     Installing package into ‘/usr/local/lib/R/site-library’
     (as ‘lib’ is unspecified)


     Attaching package: ‘purrr’


     The following object is masked from ‘package:scales’:

         discard


     The following object is masked from ‘package:plyr’:

         compact


     The following object is masked from ‘package:data.table’:

         transpose
```

```
# Use map_dbl to run many models with varying value of k (centers)
tot_withinss <- map_dbl(1:10,  function(k){
  model <- kmeans(x = z, centers = k)
  model$tot.withinss
})
```

```
# Generate a data frame containing both k and tot_withinss
```

```
elbow_df <- data.frame(
  k = 1:10,
  tot_withinss = tot_withinss
)
```

```
# Plot the elbow plot
ggplot(elbow_df, aes(x = k, y = tot_withinss)) +
  geom_line() +
  geom_point()+
  theme_classic() +
  scale_x_continuous(breaks = 1:10)
```



Alternative vizualisations

```
install.packages("factoextra")
```

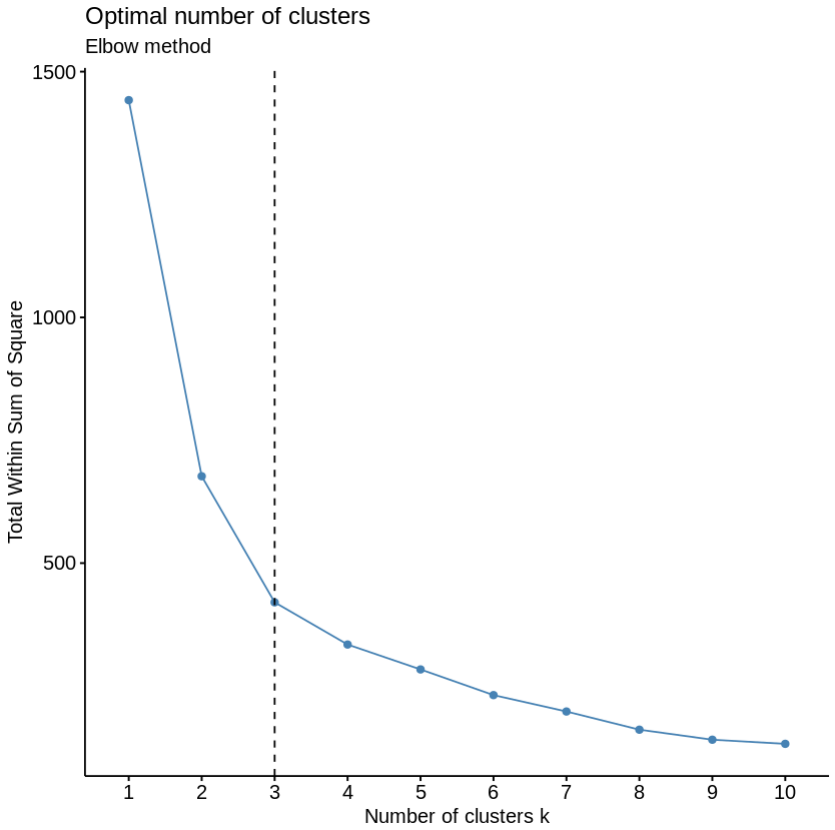
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘matrixStats’, ‘RcppArmadillo’, ‘numDeriv’, ‘SparseM’, ‘MatrixModels’, ‘conquer’, ‘sp


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘NBClust’ is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages”
Warning message:
“Perhaps you meant ‘NbClust’ ?”
```

```
install.packages("NbClust")
```

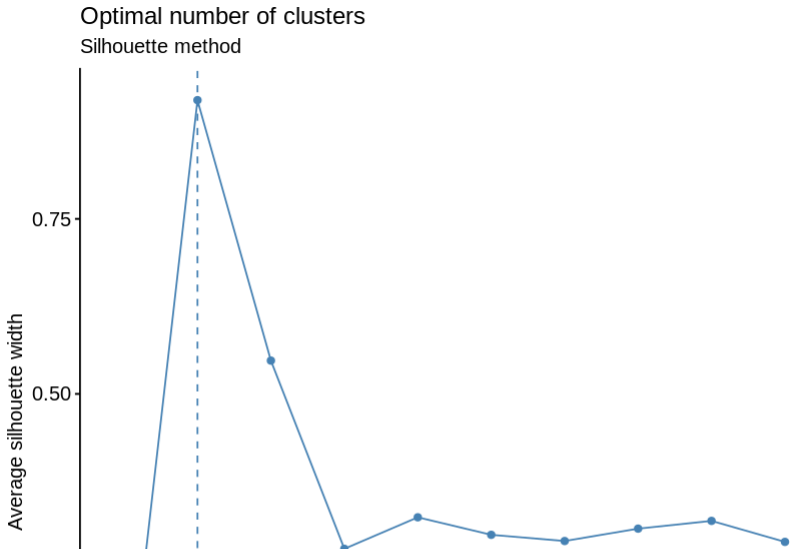
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(factoextra)
library(NbClust)
```

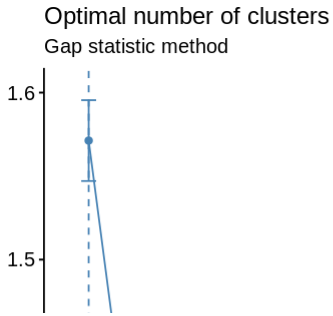
```
# z = scaled data
# Elbow method
fviz_nbclust(z, kmeans, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```
# Silhouette method
fviz_nbclust(z, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```



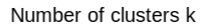

```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(z, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```



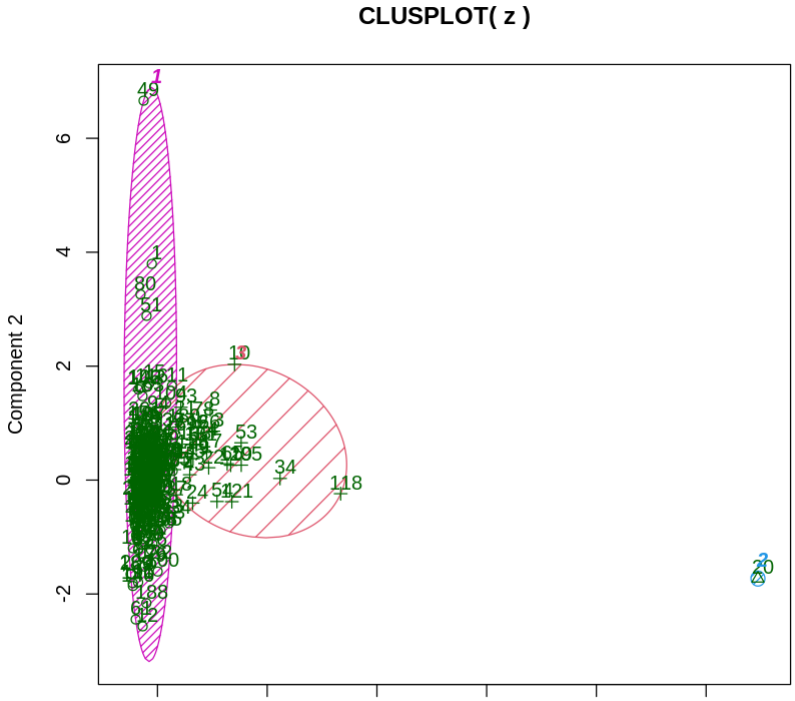
K-Means clustering


```
install.packages("fpc")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘modeltools’, ‘DEoptimR’, ‘mclust’, ‘flexmix’, ‘prabclus’, ‘diptest’, ‘robustbase’, ‘
```




```
# K-Means Clustering with 3 clusters
fit <- kmeans(z, 3)


# Cluster Plot against 1st 2 principal components

# vary parameters for most readable graph
library(cluster)

clusplot(z, fit$cluster, color=TRUE, shade=TRUE,
   labels=2, lines=0)
```




```
# Centroid Plot against 1st 2 discriminant functions
library(fpc)

plotcluster(z, fit$cluster)
```


```
tail(fit,3)
```

```
$size
[1] 177 · 1 · 29
$iter
[1] 3
$ifault
[1] 0
```

```
summary(fit)
```

```
             Length Class  Mode
cluster      207    -none- numeric
centers       21    -none- numeric
totss          1    -none- numeric
withinss       3    -none- numeric
tot.withinss   1    -none- numeric
betweenss      1    -none- numeric
size           3    -none- numeric
iter           1    -none- numeric
ifault         1    -none- numeric
```

Append column clusters to original df

```
# To demonstrate why we can remove this outliers
# Compute mahalonobis distance and flag outliers if any
# [source](https://www.youtube.com/watch?v=BdEOIQ2ozYM&t=223s)

# First, let's calculate mahalanobis distance with height and weight distribution
# let's select price and year columns
Sx <- cov(sub_xbox360[, c(4,8)])
MD <- mahalanobis(sub_xbox360[, c(4,8)], colMeans(sub_xbox360[, c(4,8)]), Sx)



Sx <- cov(xbox360_911420_sub[, c(4,8)])
MD <- mahalanobis(xbox360_911420_sub[, c(4,8)], colMeans(xbox360_911420_sub[, c(4,8)]), Sx)



# covariance matrix for price and year, with the variances in its diagonal
Sx
```

A matrix: 2 × 2 of type dbl

	price	year
price	8752.87861	-18.781293
year	-18.78129	1.272325

```
# covariance matrix for price and number of store per product per day
Sx1 <- cov(xbox360_911420_sub[, c(4,12)])
MD1 <- mahalanobis(xbox360_911420_sub[, c(4,12)], colMeans(xbox360_911420_sub[, c(4,12)]), Sx1)


Sx1
```

A matrix: 2 × 2 of type dbl

	price	number_of_store_per_product_and_day
price	8752.87861	-48.00986
number_of_store_per_product_and_day	-48.00986	23.32179

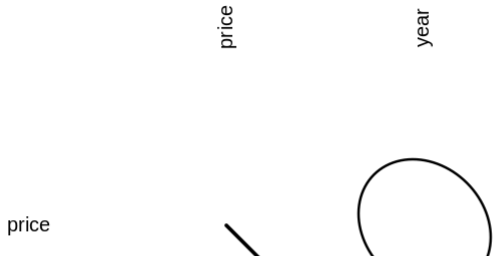
```
install.packages("pcaPP", repo="http://cran.r-project.org", dep=T)
library(pcaPP)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘mvtnorm’


Error in plotCov(Sx): could not find function "plotCov"
Traceback:
```

SEARCH STACK OVERFLOW

```
plotcov(Sx)
```

```
Warning message in if (class(cov1) == "matrix") cm1 = cov1 else if (is.null(cov1$cov)) stop("No appropriate covariance
“the condition has length > 1 and only the first element will be used”
Warning message in plot.xy(xy.coords(x, y), type = type, ...):
“supplied color is neither numeric nor character”
```




```
plotcov(Sx1)
```

```
Warning message in if (class(cov1) == "matrix") cm1 = cov1 else if (is.null(cov1$cov)) stop("No appropriate covariance
“the condition has length > 1 and only the first element will be used”
Warning message in plot.xy(xy.coords(x, y), type = type, ...):
“supplied color is neither numeric nor character”
```

Let's explore the first 100 MD data while rounding it up to 2 decimals

```
#head(MD)
```

1.46343045772125 · 0.602163062952911 · 0.194938685534366 · 5.00209748932361 · 1.46343045772125 · 1.91238182651806

```
MD[1:300] %>% round(2)
```

0.49 · 0.49 · 0.48 · 0.48 · 0.48 · 0.48 · 0.48 · 0.48 · 0.47 · 2.49 · 2.49 · 2.49 · 2.49 · 2.45 · 2.45 · 2.46 · 2.46 · 2.46 · 2.46 · 2.38 · 2.38 · 2.38 · 2.38 ·
2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.4 · 2.4 · 2.4 · 2.4 · 2.4 · 2.4 · 2.41 · 2.41 · 2.39 · 2.39 · 2.39 · 2.39 · 2.39 · 2.39 · 2.39 · 2.39 ·
2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.39 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.38 · 2.39 · 2.35 · 2.35 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 ·
0.11 · 0.11 ·
0.11 · 0.11 ·
0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.11 · 0.42 · 0.42 · 0.42 · 0.42 · 0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42 ·
0.42 · 0.42

```
xbox360_911420_sub$MD <- round(MD,3)
```

```
xbox360_911420_sub_MD = xbox360_911420_sub
```

```
head(xbox360_911420_sub_MD,3)
```

A data

product_id	date	category	price	weekday	week	store_id	year	month	cpi_adjusted_price	log_of_cpi_adjusted_p
<int>	<date>	<chr>	<int>	<chr>	<int>	<int>	<int>	<chr>	<dbl>	<
911420	2016-12-23	Xbox 360	231	Friday	51	20443	2016	Dec	231.0361	5.44
911420	2016-12-24	Xbox 360	231	Saturday	51	20443	2016	Dec	231.0361	5.44

We now want to flag outliers (price data points), and for that we need to get a sense of the data distribution of MDs in order to know what could be a reasonable threashold. Let's then do quick plots.

```
plot(xbox360_911420_sub_MD$MD, xbox360_911420_sub_MD$price)
```

```
# Let's create a new column where outliers will be flagged
# called outlier_MD
sub_xbox360$outlier_MD <- FALSE
sub_xbox360$outlier_MD[sub_xbox360$MD > 50] <- TRUE
```

```
# Let's create a new column where outliers will be flagged
# called outlier_MD
xbox360_911420_sub_MD$outlier_MD <- FALSE
xbox360_911420_sub_MD$outlier_MD[xbox360_911420_sub_MD$MD > 50] <- TRUE
```

```
tail(sub_xbox360)
```

A data.table

product_id	date	category	price	weekday	week	store_id	year	month	cpi_adjusted_price	log_of_cpi_adjusted
<int>	<date>	<chr>	<int>	<chr>	<int>	<fct>	<int>	<chr>	<dbl>	
3096808	2016-04-06	Xbox 360	849	Wednesday	14	428	2016	Apr	860.0012	6.
2010167	2016-01-22	Xbox 360	309	Friday	3	12377	2016	Jan	315.5129	5.
2678994	2015-01-28	Xbox 360	599	Wednesday	5	578	2015	Jan	616.3098	6.
1263856	2015-04-24	Xbox 360	249	Friday	17	13676	2015	Apr	254.2239	5.
3169815	2016-08-17	Xbox 360	249	Wednesday	33	428	2016	Aug	251.6365	5.
676491	2015-01-24	Xbox 360	399	Saturday	4	112	2015	Jan	410.5302	6.

```
xbox360_911420_sub_MD0 = xbox360_911420_sub_MD

tail(xbox360_911420_sub_MD0)
```

product_id <int>	date <date>	category <chr>	price <int>	weekday <chr>	week <int>	store_id <int>	year <int>	month <chr>	cpi_adjusted_price <dbl>	log_of_cpi_adjusted
911420	2017-02-20	Xbox 360	199	Monday	8	1595	2017	Feb	199	5.
911420	2017-02-21	Xbox 360	199	Tuesday	8	1595	2017	Feb	199	5.
911420	2017-02-22	Xbox 360	199	Wednesday	8	1595	2017	Feb	199	5.
911420	2017-02-23	Xbox 360	199	Thursday	8	1595	2017	Feb	199	5.
911420	2017-02-24	Xbox 360	199	Friday	8	1595	2017	Feb	199	5.
911420	2017-02-25	Xbox 360	199	Saturday	8	1595	2017	Feb	199	5.

```
tail(xboxb360_911420_sub_MD0,3)
```

product_id	date	category	price	weekday	week	store_id	year	month	cpi_adjusted_price	log_of_cpi_adjusted_p

```
# order in the MD column in ascending order
xboxb360_911420_sub_MD0 = xboxb360_911420_sub_MD0[order(xboxb360_911420_sub_MD0$MD, decreasing = FALSE),]
```

911420	02-23	Xbox 360	199	Thursday	8	1595	2017	Feb	199	5.29

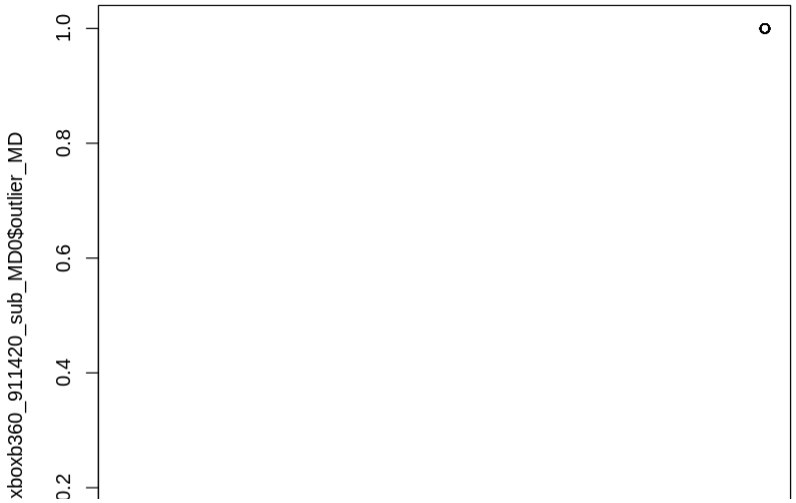
```
# display the points with highest MD (90.888)
# thus outliers
tail(xboxb360_911420_sub_MD0,3)
```

product_id	date	category	price	weekday	week	store_id	year	month	cpi_adjusted_price	log_of_cpi_adjusted
<int>	<date>	<chr>	<int>	<chr>	<int>	<int>	<int>	<chr>	<dbl>	
911420	2013-08-27	Xbox 360	1108	Tuesday	35	1260	2013	Aug	1128.794	7.
911420	2013-08-28	Xbox 360	1108	Wednesday	35	1260	2013	Aug	1128.794	7.
911420	2013-08-29	Xbox 360	1108	Thursday	35	1260	2013	Aug	1128.794	7.

```
dim(xboxb360_911420_sub_MD0)
```

10152 · 17

```
plot(xboxb360_911420_sub_MD0$MD, xboxb360_911420_sub_MD0$outlier_MD)
```




```
md_plot <- ggplot(xboxb360_911420_sub_MD0, aes(MD, outlier_MD), colour = MD)

md_plot + geom_point(outlier.colour = "red", outlier.shape = 1, shape = xboxb360_911420_sub_MD0$outlier_MD) +
theme_tufte() +
theme(axis.text.x = element_text(angle = 70)) +
labs(x = "mahalanobis distance (MD)", y = "outlier") +
#xlab("mahalanobis distance (MD)") + ylab("outlier") +
ggsave("md_plot.png",width=6, height=4,dpi=300)
```

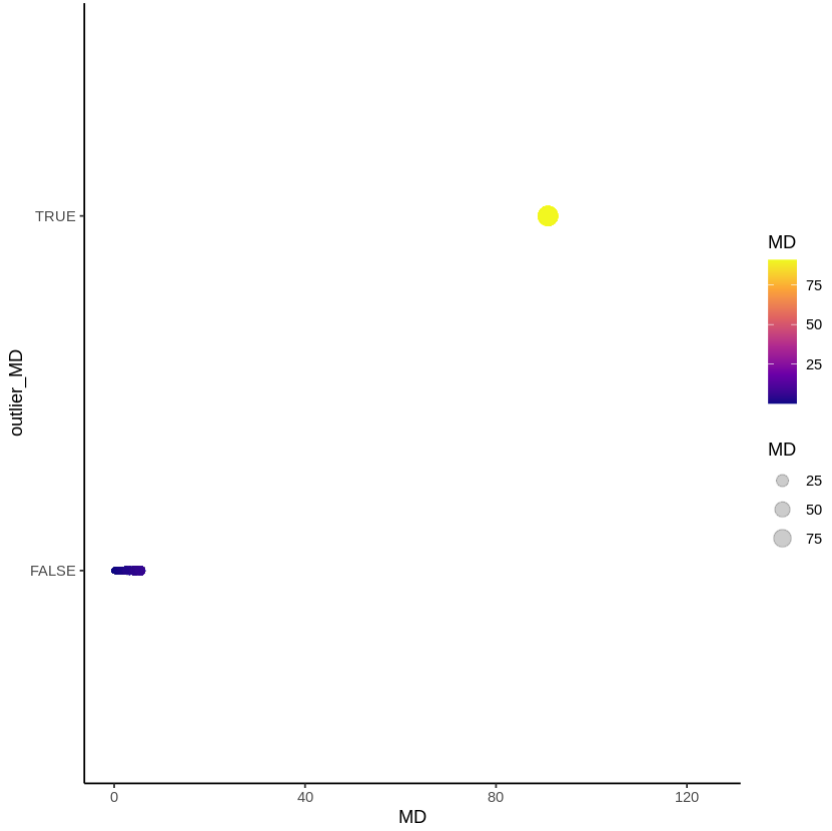
```
Warning message:
“Ignoring unknown parameters: outlier.colour, outlier.shape”
```

TRUE -

```
xboxb360_911420_sub_MD0$MD <- as.numeric(xboxb360_911420_sub_MD0$MD)


md_plot1 <- ggplot(xboxb360_911420_sub_MD0, aes(MD, outlier_MD, size = MD, colour = MD))

md_plot1 + geom_point(alpha=0.2) +
theme_tufte() +
#scale_fill_gradient("MD", low = "green", high = "red") +
#scale_colour_gradient2() +
scale_colour_viridis_c(option = "plasma") +
scale_size(range = c(1, 5))  +
scale_x_continuous(limits = c(0, 125)) + # Show dots
  geom_label(
    label=rownames(xboxb360_911420_sub_MD0$),
    nudge_x = 0.25, nudge_y = 0.25,
    check_overlap = T
  )
#geom_text(aes(label = outlier_MD), hjust = -0.5, , size = 3)
#ggsave("md_plot1.png",width=6, height=4,dpi=300)
```

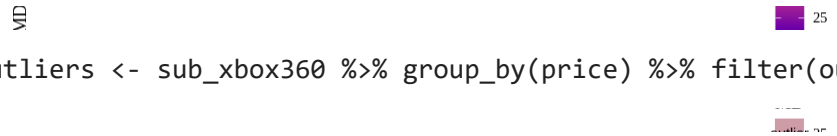
```
Error in parse(text = x, srcfile = src): <text>:11:44: unexpected ')'
10:   geom_label(
```

```
md_plot1 <- ggplot(xboxb360_911420_sub_MD0, aes(MD, outlier_MD, size = MD, colour = MD))

md_plot1 + geom_point(alpha=0.2) +
theme_classic() +
#scale_fill_gradient("MD", low = "green", high = "red") +
#scale_colour_gradient2() +
scale_colour_viridis_c(option = "plasma") +
scale_size(range = c(1, 5))  +
scale_x_continuous(limits = c(0, 125))
#geom_text(aes(label = outlier_MD), hjust = -0.5, , size = 3)
#ggsave("md_plot1.png",width=6, height=4,dpi=300)
```



```
md_plot11 <- ggplot(xboxb360_911420_sub_MD0, aes(MD, outlier_MD, size = MD, colour = MD))

md_plot11 + geom_point(alpha=0.2) +
theme_tufte() +
#scale_fill_gradient("MD", low = "green", high = "red") +
#scale_colour_gradient2() +
scale_colour_viridis_c(option = "plasma") +
scale_size(range = c(1, 5))  +
scale_x_continuous(limits = c(0, 125)) + # Show dots
  geom_label(
    label="outlier",
    x=90.888,
    y="TRUE",
    label.padding = unit(0.55, "lines"), # Rectangle size around label
    label.size = 0.35,
    color = "black",
    fill="#cd9ca7"
  )
#geom_text(aes(label = outlier_MD), hjust = -0.5, , size = 3)
#ggsave("md_plot1.png",width=6, height=4,dpi=300)
```

Identify the products that are outliers




```
df_outliers <- sub_xbox360 %>% group_by(price) %>% filter(outlier_MD == "TRUE")
```

```
df_outliers_MD <- xbox360_911420_sub_MD %>% group_by(price) %>% filter(outlier_MD == "TRUE")
```


```
dim(df_outliers_MD)
```

97 · 17

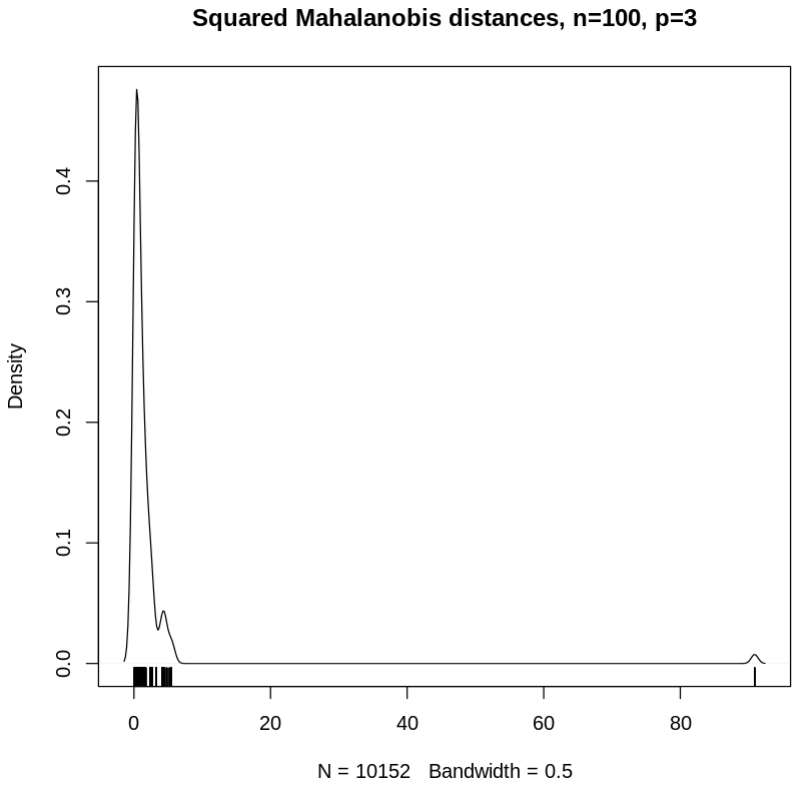
```
head(df_outliers_MD)
```

product_id	date	category	price	weekday	week	store_id	year	month	cpi_adjusted_price	log_of_cpi_adjusted
<int>	<date>	<chr>	<int>	<chr>	<int>	<int>	<int>	<chr>	<dbl>	

```
dim(df_outliers)
```

10 · 16

```
plot(density(MD, bw = 0.5),
     main="Squared Mahalanobis distances, n=100, p=3") ; rug(MD)
```




```
library(ggplot2)
```

```
Attaching package: ‘ggplot2’


The following objects are masked from ‘package:psych’:

    %+%, alpha
```

```
# ggplot dendogram
dg <- ggplot(distance, aes(distance))

dg + theme_classic()
```

```
Error: `data` must be a data frame, or other object coercible by `fortify()`, not an S3 object with class dist
Traceback:

1. ggplot(distance, aes(distance))
2. ggplot.default(distance, aes(distance))
3. fortify(data, ...)
4. fortify.default(data, ...)
5. abort(msg)
6. signal_abort(cnd)
```

SEARCH STACK OVERFLOW

Double-click (or enter) to edit

K-Means clustering

source

```
#sub_xbox360_clean <- na.omit(sub_xbox360)
```

This error occurs also due to non numeric values present in the table. Kmeans cannot handle data that has NA or NAN values, which is the case in our data frame. The mean and variance are then no longer well defined, and we don't know anymore which center is closest.

A work around could be to focus only on the numerical variables.

```
#
k2 <- kmeans(na.omit(xbox360_911420_sub_2014),4) # into 3 cluster

str(k2)
```

```
Warning message in storage.mode(x) <- "double":
“NAs introduced by coercion”
Error in do_one(nmeth): NA/NaN/Inf in foreign function call (arg 1)
Traceback:

1. kmeans(na.omit(xbox360_911420_sub_2014), 4)
2. do_one(nmeth)
```

SEARCH STACK OVERFLOW

```
print(k2)
```

```
 [1009] 2 1 1 3 2 2 2 1 2 2 3 2 2 2 2 2 2 1 1 2 2 2 2 1 1 1 1 2 3 2 1 1 2 2 1
 [1045] 1 1 1 2 1 1 2 1 3 1 2 1 3 1 3 1 1 1 2 3 1 1 1 1 2 1 1 3 1 1 1 2 2 3 1 2
```

```
[1081] 2 1 1 1 2 2 2 1 2 3 2 1 1 2 1 2 1 3 1 1 1 1 1 3 1 2 1 1 1 2 2 2 1 2 1 1
[1117] 2 3 1 2 1 2 2 1 1 1 1 1 2 1 2 2 1 2 3 1 1 1 2 1 3 1 2 1 1 2 1 2 1 1 1 1
[1153] 1 1 1 2 3 2 2 1 1 1 1 1 2 1 1 2 3 1 1 1 1 1 1 2 1 1 2 1 2 2 1 2 1 1 3 1
[1189] 2 1 1 1 2 3 1 1 1 2 2 2 2 1 1 2 1 1 1 1 1 2 3 2 1 2 3 1 2 1 1 2 1 1 1 2
[1225] 1 2 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1 1 2 3 2 2 2 1 1 2 2 1 2 1 1 2 1 2 2 1

[1261] 1 2 3 2 1 1 2 1 1 2 1 1 2 1 3 2 1 1 1 2 1 1 1 1 3 1 2 1 3 3 1 3 1 2 3 2
[1297] 2 2 1 2 1 2 1 2 1 2 2 1 1 1 3 2 3 3 1 2 2 2 1 1 1 2 2 1 1 2 1 2 1 1 1 2
[1333] 2 1 3 3 2 2 1 2 1 1 1 2 1 1 1 1 2 1 3 1 2 1 1 1 1 1 1 1 2 1 1 1 1 3 2 2
[1369] 1 1 1 1 2 2 2 1 2 1 1 2 1 1 1 1 1 1 3 1 1 1 3 3 1 3 3 1 3 1 2 2 1 1 1 1
[1405] 2 1 1 2 2 1 2 3 2 2 2 1 1 1 1 1 3 2 1 1 2 1 2 1 2 1 3 2 1 3 1 1 2 2 1 1
[1441] 2 1 1 1 2 3 2 2 1 2 2 2 1 3 1 1 1 2 2 1 1 1 1 1 1 2 2 2 1 1 2 2 2 1 1 2
[1477] 2 2 2 1 2 1 1 2 2 1 3 2 2 2 1 2 1 1 2 2 1 2 2 1 1 1 2 3 1 1 1 1 1 2 1 1
[1513] 2 1 2 2 1 1 1 2 1 2 2 1 1 1 1 1 1 2 1 1 1 2 1 3 1 1 1 2 1 1 1 3 1 1 1 1
[1549] 2 3 1 3 1 2 1 1 1 2 2 1 1 2 1 2 3 1 3 1 1 1 2 2 2 1 2 1 2 1 1 2 2 3 1 2
[1585] 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 2 1 1 1 1 2 1 2 2 1 3 1 1 2 2 2 2 2 1
[1621] 2 1 2 1 3 1 1 2 2 3 2 1 2 2 2 1 1 1 2 2 1 1 1 3 1 2 2 1 2 2 1 1 2 1 2 1
[1657] 3 3 1 2 2 1 2 2 3 3 3 1 2 3 1 2 1 2 2 2 1 2 1 1 1 1 2 2 2 1 1 3 2 1 1 1
[1693] 3 2 1 1 3 1 2 1 1 1 1 3 1 2 1 2 2 1 2 1 2 1 1 2 1 3 1 1 1 1 2 1 2 1 1 3
[1729] 1 2 1 1 1 1 2 1 2 1 2 3 2 2 2 1 2 1 2 1 1 2 1 1 2 1 2 3 1 3 1 2 1 2 1 1
[1765] 1 3 2 2 2 2 1 3 2 2 2 1 1 1 1 1 1 2 3 1 1 1 1 1 2 1 2 2 1 1 1 2 1 2 2 2
[1801] 2 2 2 3 1 1 1 2 1 2 2 3 2 1 3 1 2 1 2 1 1 3 2 1 3 1 1 1 1 1 1 1 1 1 3 1
[1837] 2 1 1 1 2 1 1 3 3 1 3 3 1 1 3 2 1 2 2 1 2 2 2 2 3 1 2 1 3 1 2 3 2 2 1 1
[1873] 1 2 1 2 1 1 1 2 3 3 2 2 2 2 1 1 2 1 1 1 2 1 1 1 1 1 1 1 1 2 1 3 1 1 2 2
[1909] 3 2 2 3 3 1 1 1 1 1 1 1 2 2 1 1 1 1 1 2 2 1 1 2 1 3 1 2 1 1 3 2 2 3 2 2
[1945] 1 1 1 2 2 1 1 2 1 1 2 2 2 3 2 3 3 2 1 1 1 1 2 1 1 1 2 1 1 2 1 1 3 2 2 1
[1981] 1 2 3 1 2 1 1 3 1 3 1 2 2 1 2 2 1 1 1 2 1 2 1 3 1 2 2 1 2 2 1 1 1 1 1 1
[2017] 1 2 2 2 2 2 1 3 1 1 2 1 1 1 1 3 1 2 2 1 1 1 1 2 1 2 1 1 2 1 1 1 1 1 1 1
[2053] 2 1 2 1 1 1 2 1 2 3 2 1 1 1 3 1 1 3 2 1 1 2 1 2 1 1 1 3 2 3 1 1 1 1 2 1
[2089] 2 1 2 3 2 1 1 2 2 1 1 2 1 3 2 3 1 1 1 3 1 2 2 2 2 1 1 1 1 1 1 1 2 1 1
[2125] 2 2 1 1 1 2 1 1 1 1 1 2 1 1 2 2 1 1 3 3 2 2 1 2 2 1 1 1 1 2 2 1 1 2 3 2
[2161] 1 2 1 2 1 2 1 1 2 1 3 1 2 2 1 1 1 3 2 2 1 1 1 1 3 2 3 3 1 1 1 1 1 2 1 1
[2197] 1 1 2 1 3 1 2 1 2 3 1 1 1 1 1 1 1 2 1 1 2 2 1 1 1 2 2 1 2 3 1 3 1 1 1 1
[2233] 2 1 1 3 1 1 1 1 1 3 1 1 1 1 2 3 2 1 2 2 1 1 1 1 2 2 2 2 1 2 1 1 1 1 1 1
[2269] 1 1 2 1 2 1 1 3 2 1 1 2 1 3 3 1 1 1 2 2 1 1 1 2 1 1 2 2 1 3 1 2 2 1 2 2
[2305] 1 1 2 1 2 2 2 1 1 1 1 3 2 1 1 2 1 3 1 1 2 1 3 2 1 2 2 1 2 1 3 2 1 2 1 1
[2341] 3 3 2 3 1 2 2 2 1 2 1 1 3 3 2 3 1 3 2 1 1 1 2 1 2 3 1 1 3 2 1 1 3 2 3 2
[2377] 3 3 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 2 1 2 2 1 2 3 3 1 1 1 1 1 1 2 2 3 2 1
[2413] 1 2 1 2 1 1 2 3 2 1 1 1 1 1 2 3 1 1 3 1 2 1 3 1 1 2 2 1 2 1 2 3 2 1 3 1
[2449] 2 1 1 1 2 2 3 1 1 2 3 2 2 1 2 2 1 3 1 1 2 1 2 2 1 3 2 2 1 1 1 1 1 1 2 1
[2485] 1 1 1 1 2 1 2 2 2 1 2 2 1 3 1 1 2 1 1 2 1 1 3 3 3 1 2 2 1 1 1 2 1 1 2 2
[2521] 1 1 2 1 3 2 1 3 2 1 3 1 1 1 2 1 1 1 3 1 2 2 1 1 1 2 2 2 1 1 2 1 3 1 1 1
[2557] 1 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 2 2 1 2 1 2 3 3 1 1 2 1 1 1 1 2 1 2 2 1
[2593] 1 1 2 3 2 1 2 1 2 1 1 1 2 2 1 1 1 1 1 2 2 1 3 2 1 2 3 1 1 1 1 1 1 1 1 1
```

```
 [2629] 2 1 3 1 3 2 1 1 1 2 1 1 1 2 3 2 1 2 2 1 2 3 2 1 1 2 1 2 2 1 3 1 1 2 1 2
 [2665] 1 1 2 1 2 2 1 2 1 1 2 1 2 3 1 2 1 1 1 2 2 2 2 1 1 1 1 2 2 1 2 1 2 2 1 1
 [2701] 1 1 1 1 1 1 2 1 2 2 1 3 2 1 1 2 3 1 1 1 2 3 1 1 2 1 1 1 1 3 3 1 3 1 3 1
 [2737] 1 1 2 1 1 2 1 1 1 3 3 2 1 2 1 1 3 2 1 3 1 2 3 2 1 2 1 1 3 1 1 1 2 2 2 3
 [2773] 1 2 1 2 1 1 1 2 2 1 2 2 1 2 2 2 1 1 1 1 1 3 2 2 1 1 2 1 2 1 1 1 1 1 2 1

 [2809] 3 1 1 3 1 1 2 1 2 1 2 1 1 1 1 2 3 2 1 3 1 2 1 2 1 3 1 1 2 2 2 1 2 1 3 3
 [2845] 1 1 2 1 3 1 3 1 1 1 2 2 1 1 1 1 2 1 2 1 1 2 2 2 1 1 3 1 2 2 1 1 1 1 1 2
 [2881] 2 1 1 2 2 2 1 3 3 1 1 1 2 1 1 2 1 1 3 2 2 1 1 2 2 1 3 1 1 1 2 1 2 1 1 2
 [2917] 2 1 1 3 2 2 1 1 1 1 2 2 1 3 1 2 2 1 1 1 1 1 3 1 1 2 1 1 1 3 3 2 3 2 1 3
 [2953] 2 1 1 1 2 1 1 2 1 2 3 1 2 1 2 2 2 1 2 2 2 2 2 1 3 2 3 2 1 3 1 1 1 2 1 2
 [2989] 3 2 1 2 1 1 1 1 1 1 1 1 1 2 2 2 3 2 1 2 1 1 1 2 1 1 1 1 1 2 2 2 3 2 1 1
 [3025] 1 1 1 1 1 2 2 1 1 1 3 1 1 3 3 3 1 2 1 1 3 1 1 1 2 2 2 1 3 1 3 1 1 1 1 1
 [3061] 1 2 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1 2 2 2 1 2 2 2 2 3 2 1 2
 [3097] 1 1 1 2 1 2 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 2 2 1 1 1 2 2 2 1 1 2 1 1 1
```

```
install.packages("corrplot")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

```
library(corrplot)
```

```
corrplot 0.84 loaded
```

```
num.col <- sapply(sub_xbox360, is.numeric)
```

```
tail(num.col,3)
```

```
number_of_store_per_product_and_day:  TRUE total_days_for_product:  TRUE total_products_in_category:  TRUE
```

```
cor.data <- cor(sub_xbox360[, num.col])
```

```
Error in `[.data.table`(sub_xbox360, , num.col): j (the 2nd argument inside [...]) is a single symbol but column name
'num.col' is not found. Perhaps you intended DT[, ..num.col]. This difference to data.frame is deliberate and
explained in FAQ 1.1.
Traceback:

1. cor(sub_xbox360[, num.col])
2. is.data.frame(x)
3. sub_xbox360[, num.col]
4. `[.data.table`(sub_xbox360, , num.col)
5. stop("j (the 2nd argument inside [...]) is a single symbol but column name '",
 .     jsubChar, "' is not found. Perhaps you intended DT[, ..",
```

```
#fviz_cluster(k2, data = sub_xbox360)
```

SEARCH STACK OVERFLOW

Elbow method [source]

```
# Determine and plot the optimal number of clusters

# function to compute total within-cluster sum of square
wss <- (nrow(xbox360_911420_sub)-1)*sum(apply(xbox360_911420_sub,2,var))

# Compute and plot wss for k = 1 to k = 15
# extract wss for 2-15 clusters
for (i in 2:15) wss[i] <- sum(kmeans(xbox360_911420_sub,
   centers=i)$withinss)

plot(1:15, wss, type="b", xlab="Number of Clusters K",
  frame = FALSE, ylab="Within-clusters sum of squares")
```

```
Warning message in FUN(newX[, i], ...):
“NAs introduced by coercion”
Warning message in FUN(newX[, i], ...):
“NAs introduced by coercion”
Warning message in FUN(newX[, i], ...):
“NAs introduced by coercion”
Warning message in FUN(newX[, i], ...):
“NAs introduced by coercion”
Warning message in storage.mode(x) <- "double":
“NAs introduced by coercion”
Error in do_one(nmeth): NA/NaN/Inf in foreign function call (arg 1)
Traceback:

1. kmeans(xbox360_911420_sub, centers = i)
2. do_one(nmeth)
```

SEARCH STACK OVERFLOW

```
# K-Means Cluster Analysis
fit <- kmeans(xbox360_911420_sub$price, 5) # 5 cluster solution
# get cluster means
aggregate(xbox360_911420_sub$price,by=list(fit$cluster),FUN=mean)
# append cluster assignment
xbox360_911420_sub$price <- data.frame(xbox360_911420_sub$price, fit$cluster)
```

A data.frame: 5 × 2

Group.1	x
<int>	<dbl>
1	274.5092
2	188.0031
3	1108.0000
4	171.7117
5	206.9420

```
Error in set(x, j = name, value = value): Supplied 2 items to be assigned to 10152 items of column 'price'. If you
wish to 'recycle' the RHS please use rep() to make this intent clear to readers of your code.
Traceback:

1. `$<-`(`*tmp*`, price, value = structure(list(xbox360_911420_sub.price = c(231L,
 . 231L, 229L, 229L, 229L, 229L, 229L, 229L, 227L, 227L, 227L, 227L,
 . 227L, 221L, 221L, 222L, 222L, 222L, 222L, 210L, 210L, 210L, 210L,
 . 210L, 208L, 208L, 208L, 208L, 208L, 208L, 210L, 214L, 214L, 214L,
 . 214L, 214L, 214L, 215L, 215L, 172L, 172L, 172L, 172L, 172L, 172L,
 . 172L, 172L, 173L, 173L, 173L, 173L, 173L, 173L, 173L, 172L, 173L,
 . 173L, 173L, 173L, 173L, 173L, 173L, 171L, 184L, 184L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
 . 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
```

```
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 186L, 186L, 186L, 186L, 186L, 188L, 188L,
. 188L, 188L, 188L, 188L, 188L, 188L, 188L, 188L, 188L, 188L, 188L,
. 190L, 189L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 215L, 215L, 215L, 225L, 225L, 225L, 188L, 188L, 188L, 188L, 188L,
. 188L, 186L, 186L, 185L, 185L, 186L, 186L, 186L, 225L, 186L, 186L,
. 186L, 184L, 184L, 185L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 205L, 205L, 205L, 205L, 208L, 205L, 204L, 204L, 204L, 204L,
. 204L, 204L, 203L, 203L, 203L, 203L, 203L, 203L, 203L, 201L, 203L,
. 204L, 204L, 204L, 204L, 205L, 205L, 205L, 205L, 205L, 205L, 204L,
. 205L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 203L, 203L, 203L, 203L, 203L, 203L, 204L, 203L,
. 203L, 203L, 203L, 203L, 203L, 203L, 201L, 201L, 203L, 201L, 201L,
. 201L, 201L, 201L, 201L, 200L, 200L, 201L, 201L, 201L, 201L, 201L,
. 201L, 201L, 201L, 201L, 204L, 204L, 204L, 204L, 201L, 201L, 201L,
. 203L, 203L, 179L, 179L, 178L, 178L, 178L, 179L, 178L, 176L, 178L,
. 178L, 178L, 178L, 178L, 178L, 178L, 179L, 179L, 179L, 179L, 203L,
. 201L, 201L, 201L, 201L, 203L, 203L, 203L, 204L, 203L, 203L, 203L,
. 203L, 203L, 203L, 201L, 203L, 201L, 203L, 203L, 203L, 203L, 203L,
```

```
. 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 204L, 204L, 203L, 204L, 204L, 204L, 204L, 204L,
. 204L, 203L, 203L, 204L, 204L, 204L, 204L, 203L, 201L, 201L, 200L,
. 201L, 201L, 201L, 203L, 203L, 201L, 201L, 201L, 201L, 201L, 201L,
. 201L, 201L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 226L, 226L, 199L,
. 199L, 198L, 198L, 198L, 198L, 198L, 198L, 198L, 198L, 198L, 198L,
. 198L, 198L, 198L, 199L, 200L, 199L, 200L, 200L, 200L, 200L, 200L,
. 200L, 201L, 201L, 201L, 201L, 201L, 201L, 201L, 201L, 201L, 201L,
. 201L, 201L, 200L, 199L, 199L, 200L, 200L, 200L, 200L, 200L, 200L,
. 200L, 200L, 200L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 201L, 201L, 201L, 203L, 203L, 203L, 203L, 201L, 200L, 201L, 201L,
. 205L, 205L, 205L, 203L, 203L, 204L, 203L, 203L, 203L, 203L, 203L,
. 204L, 204L, 205L, 205L, 205L, 205L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 204L, 204L, 204L, 205L, 205L, 205L, 205L, 205L,
. 205L, 205L, 205L, 206L, 206L, 206L, 206L, 206L, 204L, 205L, 205L,
. 205L, 205L, 204L, 204L, 204L, 203L, 203L, 203L, 203L, 204L, 205L,
. 204L, 204L, 204L, 205L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 205L, 205L, 205L, 205L, 206L, 206L, 206L, 206L, 204L,
. 205L, 205L, 205L, 205L, 205L, 205L, 205L, 205L, 205L, 206L, 206L,
. 206L, 206L, 206L, 206L, 206L, 206L, 206L, 206L, 208L, 208L, 208L,
. 208L, 208L, 208L, 208L, 208L, 208L, 208L, 208L, 208L, 208L, 210L,
. 210L, 210L, 210L, 210L, 209L, 209L, 209L, 209L, 209L, 209L, 209L,
. 209L, 209L, 209L, 210L, 213L, 213L, 213L, 213L, 213L, 214L, 214L,
. 215L, 215L, 215L, 215L, 215L, 215L, 214L, 214L, 213L, 213L, 213L,
. 211L, 211L, 211L, 211L, 211L, 211L, 211L, 211L, 211L, 210L, 210L,
. 211L, 211L, 211L, 210L, 210L, 210L, 211L, 211L, 211L, 211L, 211L,
. 211L, 211L, 209L, 209L, 209L, 210L, 210L, 210L, 210L, 211L, 210L,
. 210L, 210L, 210L, 209L, 209L, 208L, 208L, 208L, 208L, 208L, 208L,
. 206L, 206L, 206L, 206L, 206L, 205L, 205L, 205L, 205L, 205L, 205L,
. 205L, 205L, 206L, 205L, 205L, 204L, 204L, 204L, 204L, 205L, 205L,
. 205L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 203L, 204L, 203L, 203L, 203L, 203L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L, 204L,
. 204L, 204L, 204L, 205L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
```

```
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L, 279L,
. 279L, 279L, 279L, 279L, 279L, 279L, 279L, 246L, 245L, 245L, 245L,
. 245L, 245L, 244L, 244L, 244L, 244L, 244L, 245L, 245L, 246L, 245L,
. 243L, 243L, 243L, 243L, 243L, 244L, 244L, 245L, 243L, 243L, 243L,
. 243L, 243L, 243L, 242L, 242L, 242L, 243L, 242L, 242L, 242L, 242L,
. 241L, 241L, 242L, 242L, 243L, 243L, 243L, 243L, 243L, 243L, 243L,
. 242L, 242L, 242L, 243L, 243L, 243L, 243L, 243L, 243L, 243L, 244L,
. 274L, 274L, 274L, 274L, 274L, 274L, 274L, 274L, 274L, 274L, 274L,
. 274L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
```

```
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
```

```
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L, 225L,
. 225L, 225L, 225L, 225L, 225L, 225L, 225L, 229L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 219L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 239L, 239L, 239L,
. 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L,
. 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L,
. 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L,
. 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L, 239L,
. 239L, 239L, 239L, 239L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 219L, 219L, 219L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
```

```
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 169L, 169L, 169L, 169L, 169L, 169L, 169L,
. 169L, 169L, 169L, 169L, 169L, 169L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
```

```
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L,
. 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L,
. 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L,
. 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L, 179L,
. 179L, 179L, 179L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
```

```
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L,
. 229L, 229L, 229L, 229L, 229L, 229L, 229L, 229L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L, 219L,
. 219L, 219L, 219L, 219L, 219L, 219L, 199L, 199L, 199L, 199L, 199L,
. 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 199L, 1108L,
. 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L,
. 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L,
. 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L, 1108L
```