Project Part B: Transformation

In [73]:
```
analyst = "Khoa Nguyen" # Replace this with your name
```

In [74]:
```
f = "setup.R"; for (i in 1:10) { if (file.exists(f)) break else f = paste0("../", f) }; source(f)
options(repr.matrix.max.rows=674)
update_geom_defaults("point", list(size=1))
update_geom_defaults("col", list(fill=PALETTE[1]))
```

1 Introduction

1.1 Objective

Transform the representation of public company fundamentals. Later, use the transformed dataset along with additional analysis to recommend a portfolio of 12 company investments that maximizes 12-month return of an overall $1,000,000 investment.

1.2 Approach

Retrieve a public company fundamentals dataset comprising thousands of US companies from quarters 1, 2, 3, and 4 of year 2017 + company stock price data for those companies from quarter 4 of year 2018. The dataset has been prepared such that each company and its associated information is represented as a single observation.

Transform the dataset using variable filtration, imputation, principal component analysis, and other methods to ready it for predictive model construction.

2 Data

In [75]:
```
# Retrieve prepared data.
# How many observations and variables?
# Present the first few observations and first few variables.

data.raw = read.csv("My Prepared Data.csv", header=TRUE, na.strings=c("NA", ""), stringsAsFactors=FALSE)
data.raw$big_growth = factor(data.raw$big_growth, levels=c("YES","NO"))

size(data.raw)
fmtx(data.raw[1:6, 1:13], "data.raw (first few observations, first few variables variables)")
# fmtx(data.raw[1:3,], FFO)  # takes about 2 minutes to display all variables
```

A data.frame: 1 × 2

observations	variables
<int>	<int>
4305	2714

data.raw (first few observations, first few variables variables)

big_growth	growth	prccq	gvkey	tic	conm	datadate.q1	fyearq.q1	fqtr.q1	fyr.q1	indfmt.q1	consol.q1	popsrc.q1
NO	0.0507	43.69	1,004	AIR	AAR CORP	02/28/2017	2,016	3	5	INDL	C	D
NO	-0.3829	32.11	1,045	AAL	AMERICAN AIRLINES GROUP INC	03/31/2017	2,017	1	12	INDL	C	D
YES	0.3158	6.75	1,050	CECE	CECO ENVIRONMENTAL CORP	03/31/2017	2,017	1	12	INDL	C	D
NO	-0.2165	8.66	1,062	ASA	ASA GOLD AND PRECIOUS METALS	02/28/2017	2,017	1	11	INDL	C	D
NO	-0.1185	15.25	1,072	AVX	AVX CORP	03/31/2017	2,016	4	3	INDL	C	D
NO	0.0002	85.20	1,075	PNW	PINNACLE WEST CAPITAL CORP	03/31/2017	2,017	1	12	INDL	C	D

In [76]:
```
# Specify which variables can later be used as outcome, identifier, and predictor variables.
outvars = colnames(data.raw[1:3])
idvars = colnames(data.raw[4:6])
prevars = colnames(data.raw[4:ncol(data.raw)])
fmtsx(fmt(outvars), fmt(idvars),fmt(prevars[1:10], FFO))
```

outvars
big_growth
growth
prccq

idvars
gvkey
tic
conm

prevars (first few observations)

prevars[1:10]
gvkey
tic
conm
datadate.q1
fyearq.q1
fqtr.q1
fyr.q1
indfmt.q1
consol.q1
popsrc.q1

3 Transform

3.1 Filter Out Sparse Variables

In [77]:
```
# Filter the data to include only predictor variables with at least 95% non-missing values.
# Keep the variable names from the resulting dataset for later use.
# How many observations and variables in the resulting dataset?
#
# You can use select_if(..., ~mean(is.na(.))<...)
# You can use colnames(...)

data.filter = select_if(data.raw[, prevars], ~mean(is.na(.)) < 0.05)
fmtx(size(data.filter), "data after variable filtration")
```

data after variable filtration

observations	variables
4,305	200

3.2 Impute

In [78]:
```
# Impute missing data:
# for each numerical variable, use the mean of non-missing values;
# for each non-numerical variable, use the mode of non-missing values.
# Keep the imputed values used for each variable for future use.
#
# You can use get_impute(...)
# You can use impute(...)
#
# get_impute(data) provides a list of means and modes, one element for each variable of data.
# impute(data) provides a table like data, but imputed.

imputed_data = impute(data.filter)
fmtx(size(imputed_data), "data after imputation")
```

data after imputation

observations	variables
4,305	200

3.3 Principal Component Analysis

In [79]:
```
# For principal component analysis, filter the data to include only numerical variables with non-zero variance
# How many observations and variables?
#
# You can use select_if(..., ...)
# You can use is.numeric(...)
# You can use var(..., na.rm=TRUE)!=0

data.var = select_if(imputed_data,~is.numeric(.))
data.var = select_if(data.var, ~var(., na.rm=TRUE) != 0)
fmtx(size(data.var), "data after keeping numerical variables and removing zero-variance variables")
```

data after keeping numerical variables and removing zero-variance variables

observations	variables
4,305	151

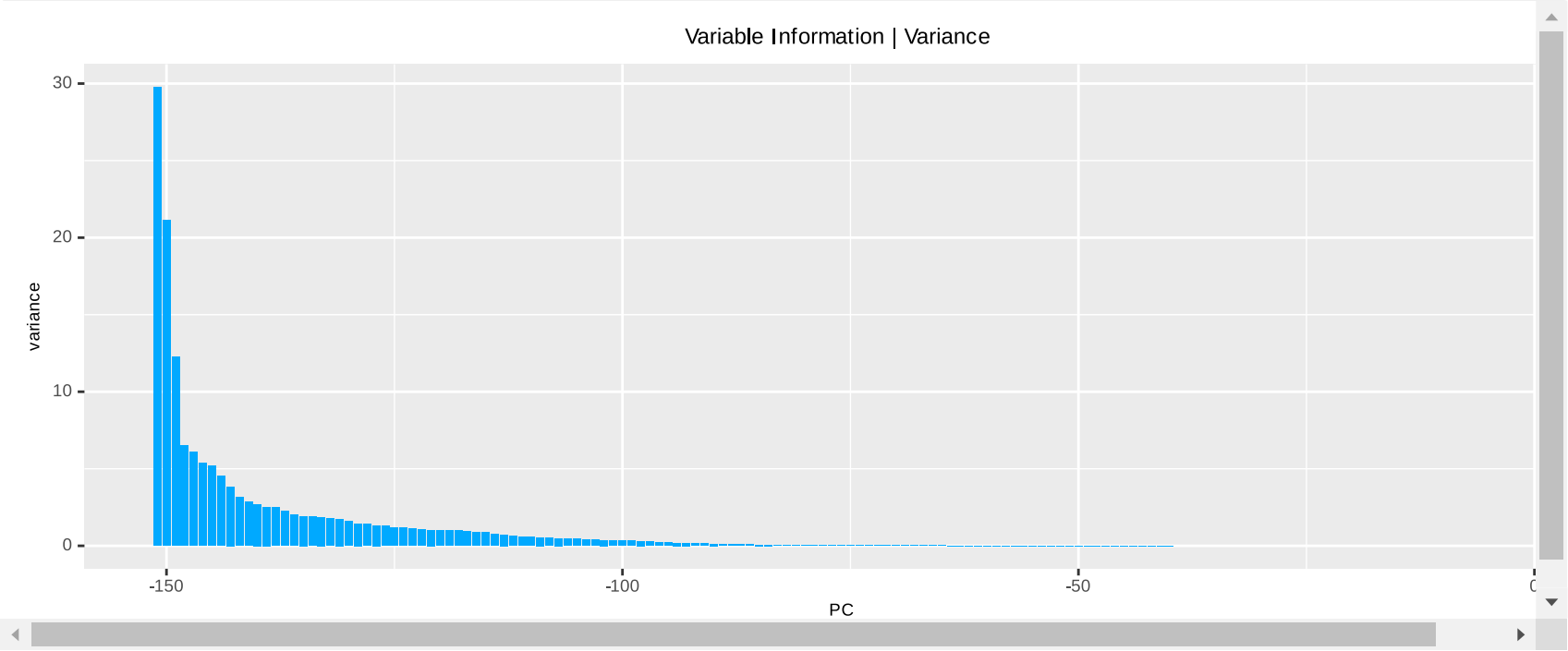
In [80]:
```
# Perform principal component analysis on the data (use scale=TRUE).
# Keep the centroid and weight matrix information (calculated by prcomp) for later use.
# Present the first few observations represented as principal components.
# For the first three principal components, +/- may be different than as shown in the expected output.
# If so, then multiply by -1 to adjust.
#
# You can use pc = prcomp(..., ..., ...)
# You can use as.data.frame(pc$x)

pc = prcomp(data.var, scale = TRUE, retx=TRUE)
centroid = pc$center
weight_matrix = pc$rotation
data.pc = as.data.frame(pc$x)
fmtx(size(data.pc))
fmtx(data.pc[1:5,], FFO)
```

size(data.pc)

observations	variables
4,305	151

PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16	
1.410	0.2125	-0.1874	-1.3446	-0.0365	-5.0250	0.8071	0.2309	0.4172	-1.4908	0.3994	0.5379	-0.5058	-0.2376	-0.8975	0.4677	-
-2.809	0.2246	1.4366	-3.2326	0.3686	1.4262	2.0046	-1.6757	1.5734	-0.3330	-1.3086	0.5723	-0.8130	-1.0506	-0.0734	-0.3888	-
1.525	0.4396	-0.1679	-0.4734	0.6653	0.3661	2.2428	1.3493	0.0613	0.0645	-0.0753	0.1220	-0.4036	-0.2399	0.4368	-0.2697	-
1.574	0.6384	0.0123	1.2309	0.0510	0.6110	-0.7640	-0.7708	0.1304	-1.1817	0.3875	0.3708	0.2988	-0.6562	-0.8140	0.2308	-
1.281	0.4529	0.0929	-0.8542	-0.3221	-4.7347	-1.0983	-0.7556	-0.1833	-1.5743	0.3494	-0.3257	1.6853	-0.2920	0.5216	-0.1302	

In [81]:
```
out(9,3.5)
var.pc = as.numeric(summarize_all(data.pc, var))
scree_data = data.frame(Principal_Component = 1:length(var.pc), Variance = var.pc)
ggplot(scree_data)  +
 geom_col(aes(x = -rank(Variance, ties.method = "first"),
              y = Variance), fill = PALETTE[1], width = 0.9) +
 xlab("PC") + ylab("variance") +
 ggtitle("Variable Information | Variance ") +
 theme.no_legend
```



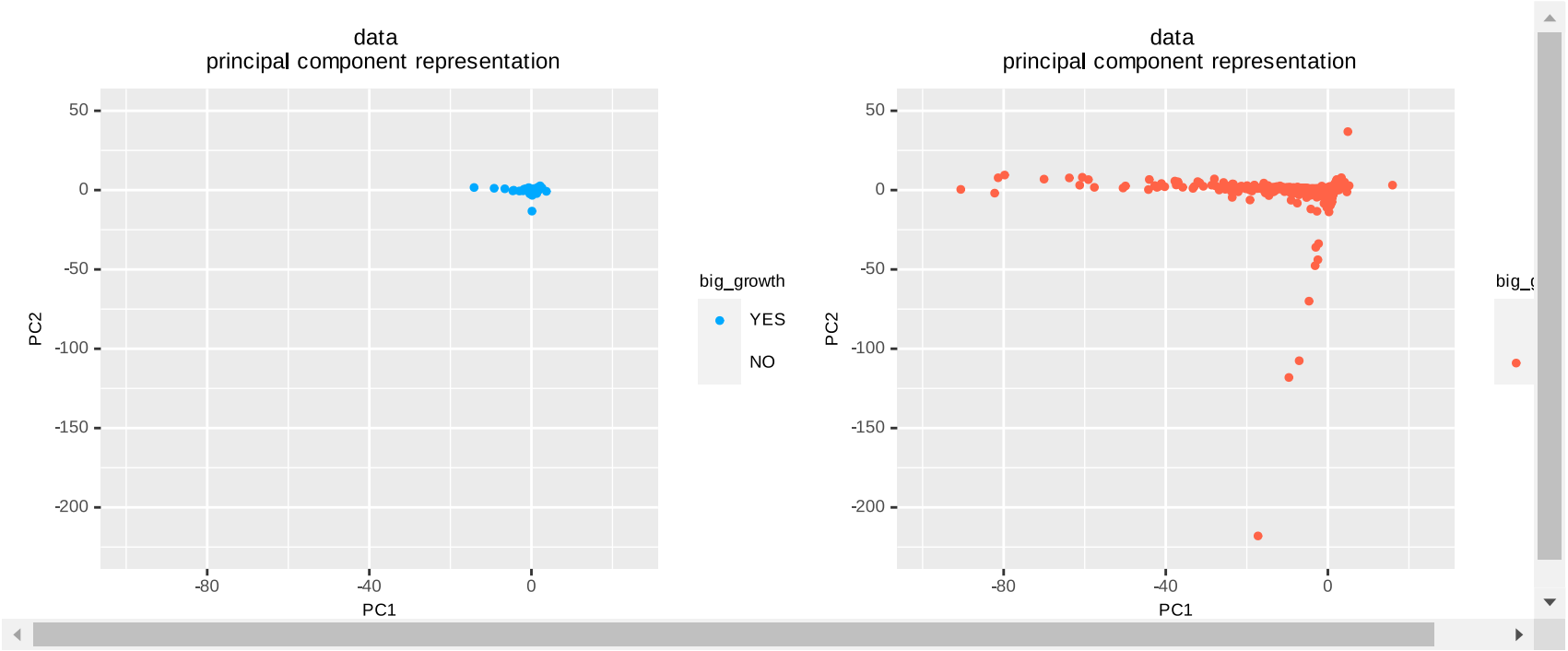
3.4 Restore Outcome and Identifier Variables

In [82]:
```
# Restore the outcome and indentifier variables to the data.
# How many observations and variables?
# Present the first few observations of the resulting dataset.
data.pcv = cbind(data.raw[, 1:6], data.pc)
fmtx(size(data.pcv))
fmtx(data.pcv[1:5,], FFO)
```

4,305 157

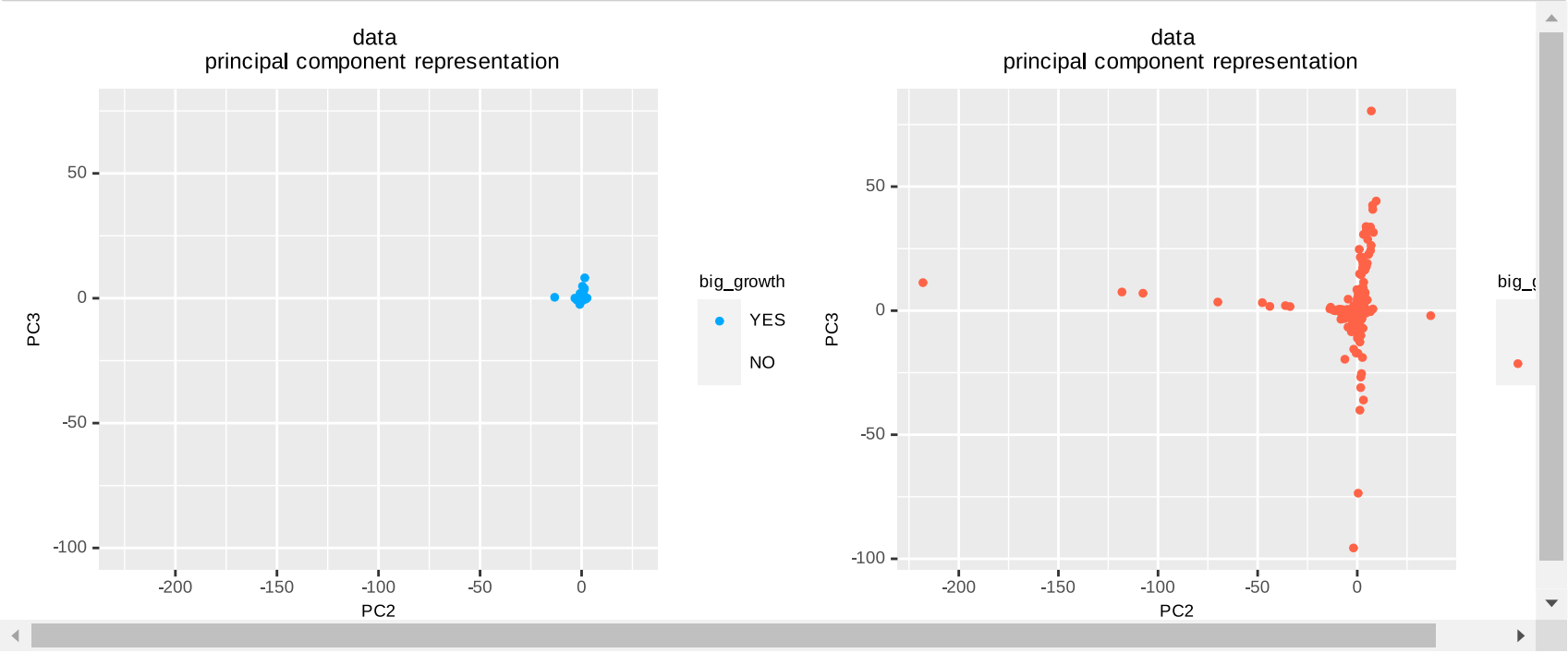
big_growth	growth	prccq	gvkey	tic	conm	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC
NO	0.0507	43.69	1,004	AIR	AAR CORP	1.410	0.2125	-0.1874	-1.3446	-0.0365	-5.0250	0.8071	0.2309	0.41
NO	-0.3829	32.11	1,045	AAL	AMERICAN AIRLINES GROUP INC	-2.809	0.2246	1.4366	-3.2326	0.3686	1.4262	2.0046	-1.6757	1.57
YES	0.3158	6.75	1,050	CECE	CECO ENVIRONMENTAL CORP	1.525	0.4396	-0.1679	-0.4734	0.6653	0.3661	2.2428	1.3493	0.06
NO	-0.2165	8.66	1,062	ASA	ASA GOLD AND PRECIOUS METALS	1.574	0.6384	0.0123	1.2309	0.0510	0.6110	-0.7640	-0.7708	0.13
NO	-0.1185	15.25	1,072	AVX	AVX CORP	1.281	0.4529	0.0929	-0.8542	-0.3221	-4.7347	-1.0983	-0.7556	-0.18

In [83]:
```
# Present 2D scatterplots to visualize PC1 vs PC2 vs big_growth
# (PC1 on horizontal axis, PC2 on vertical axis, and big_growth color-coded).
# You can use ... + geom_point(aes(..., color=big_growth, alpha=big_growth) + scale_alpha_manual(values=c(1,0)
# You can use ... + geom_point(aes(..., color=big_growth, alpha=big_growth) + scale_alpha_manual(values=c(0,1)
# You can use ... + theme.legend_title to show legend title

yes1.2 = ggplot(data.pcv) + geom_point(aes(x = PC1, y = PC2, color = big_growth, alpha = big_growth)) +
 xlim(-100, 25) +
 ylim(-225, 50) +
 scale_alpha_manual(values = c(1, 0)) +
xlab("PC1") + ylab("PC2") + ggtitle("data \n principal component representation") +
 theme.legend_title


no1.2 = ggplot(data.pcv) + geom_point(aes(x = PC1, y = PC2, color = big_growth, alpha = big_growth)) +
 xlim(-100, 25) +
 ylim(-225, 50) +
 scale_alpha_manual(values = c(0, 1)) +
xlab("PC1") + ylab("PC2") + ggtitle("data \n principal component representation") +
 theme.legend_title

grid.arrange(yes1.2, no1.2 , ncol = 2)
```



In [84]:
```
# Present 2D scatterplots to visualize PC1 vs PC3 vs big_growth
# (PC1 on horizontal axis, PC3 on vertical axis, and big_growth color-coded).
```

In [85]:
```
# Present 2D scatterplots to visualize PC2 vs PC3 vs big_growth
# (PC2 on horizontal axis, PC3 on vertical axis, and big_growth color-coded).

yes2.3 = ggplot(data.pcv) + geom_point(aes(x = PC2, y = PC3, color = big_growth, alpha = big_growth)) +
 xlim(-225, 25) +
 ylim(-100, 75) +
 scale_alpha_manual(values = c(1, 0)) +
xlab("PC2") + ylab("PC3") + ggtitle("data \n principal component representation") +
 theme.legend_title


no2.3 = ggplot(data.pcv) + geom_point(aes(x = PC2, y = PC3, color = big_growth, alpha = big_growth)) +

 scale_alpha_manual(values = c(0, 1)) +
xlab("PC2") + ylab("PC3") + ggtitle("data \n principal component representation") +
 theme.legend_title

grid.arrange(yes2.3, no2.3 , ncol = 2)
```



3.5 Filter Out Low-Variance Variables

In [86]:
```
# Filter the data to include only the outcome variables, identifier variables, and first three principal compo
# Update the specification of predictor variables.
#
# Present the outcome variables and predictor variables.
# How many observations and variables in the resulting dataset?
# Present the first few observations of the resulting dataset.
prevars = colnames(data.pcv[4:9])
fmtsx(fmt(outvars), fmt(prevars))
data = data.pcv[, 1:9]
fmtx(size(data))
fmtx(data[1:6,], FFO)
```

outvars	prevars
big_growth	gvkey
growth	tic
prccq	conm
	PC1
	PC2
	PC3

size(data)

observations	variables
4,305	9

data (first few observations)

big_growth	growth	prccq	gvkey	tic	conm	PC1	PC2	PC3
NO	0.0507	43.69	1,004	AIR	AAR CORP	1.4098	0.2125	-0.1874
NO	-0.3829	32.11	1,045	AAL	AMERICAN AIRLINES GROUP INC	-2.8093	0.2246	1.4366
YES	0.3158	6.75	1,050	CECE	CECO ENVIRONMENTAL CORP	1.5247	0.4396	-0.1679
NO	-0.2165	8.66	1,062	ASA	ASA GOLD AND PRECIOUS METALS	1.5737	0.6384	0.0123
NO	-0.1185	15.25	1,072	AVX	AVX CORP	1.2813	0.4529	0.0929
NO	0.0002	85.20	1,075	PNW	PINNACLE WEST CAPITAL CORP	0.3698	-0.4861	-0.0128

3.6 Store Transformed Data & Transformation Meta-Data

In [89]:
```
# Store the variable names of the filtered data (200 variable names)
cn = data.filter
saveRDS(cn, "My Filter.rds")
```

In [90]:
```
# Store the imputation values (200 means and modes)
ml = get_impute(cn)
saveRDS(ml, "My Imputation.rds")
```

In [91]:
```
# Store the PC information (one data object produced by prcomp that contains centroids and weight matrix)
saveRDS(pc, "My PC.rds")
```

In [92]:
```
# Store the predictor variable names of the transformed data (6 variable names)
saveRDS(prevars, "My Predictors.rds")
```

In [93]:
```
# Store the transformed data (4305 observations, 9 variables)
write.csv(data, "My Data.csv", row.names=FALSE)
```


Document revised May 6, 2023