Unsupervised Learning

Data Sets

Attrition

```
attrition <- attrition %>% mutate_if(is.ordered, factor, order = F)
attrition_h2o <- as.h2o(attrition)

churn <- initial_split(attrition, prop = .7, strata = "Attrition")

churn_train <- training(churn)
churn_test <- testing(churn)

rm(churn)
```

Ames, Iowa housing data.

```
ames <- AmesHousing::make_ames()
ames_h2o <- as.h2o(ames)

set.seed(123)

ames_split <- initial_split(ames, prop =.7, strata = "Sale_Price")

ames_train <- training(ames_split)
ames_test <- testing(ames_split)

rm(ames_split)

h2o.init(max_mem_size = "10g", strict_version_check = F)
```

```
 Connection successful!

R is connected to the H2O cluster:
    H2O cluster uptime:         30 minutes 13 seconds
    H2O cluster timezone:       America/New_York
    H2O data parsing timezone:  UTC
    H2O cluster version:        3.28.0.4
    H2O cluster version age:    11 days
    H2O cluster name:           H2O_started_from_R_bmore_vjw758
    H2O cluster total nodes:    1
    H2O cluster total memory:   15.98 GB
    H2O cluster total cores:    16
    H2O cluster allowed cores:  16
    H2O cluster healthy:        TRUE
```

```
    H2O Connection ip:          localhost
    H2O Connection port:        54321
    H2O Connection proxy:       NA
    H2O Internal Security:      FALSE
    H2O API Extensions:         Amazon S3, Algos, AutoML, Core V3, TargetEncoder, Core V4
    R Version:                  R version 3.6.2 (2019-12-12)
```

```
train_h2o <- as.h2o(ames_train)
response <- "Sale_Price"
predictors <- setdiff(colnames(ames_train), response)
```

Market Basket:

```
url <- "https://koalaverse.github.io/homlr/data/my_basket.csv"

my_basket <- readr::read_csv(url)
```

```
Parsed with column specification:
cols(
  .default = col_double()
)
```

```
See spec(...) for full column specifications.
```

```
dim(my_basket)
```

```
[1] 2000   42
```

Principal Components Analysis

h2o PCA

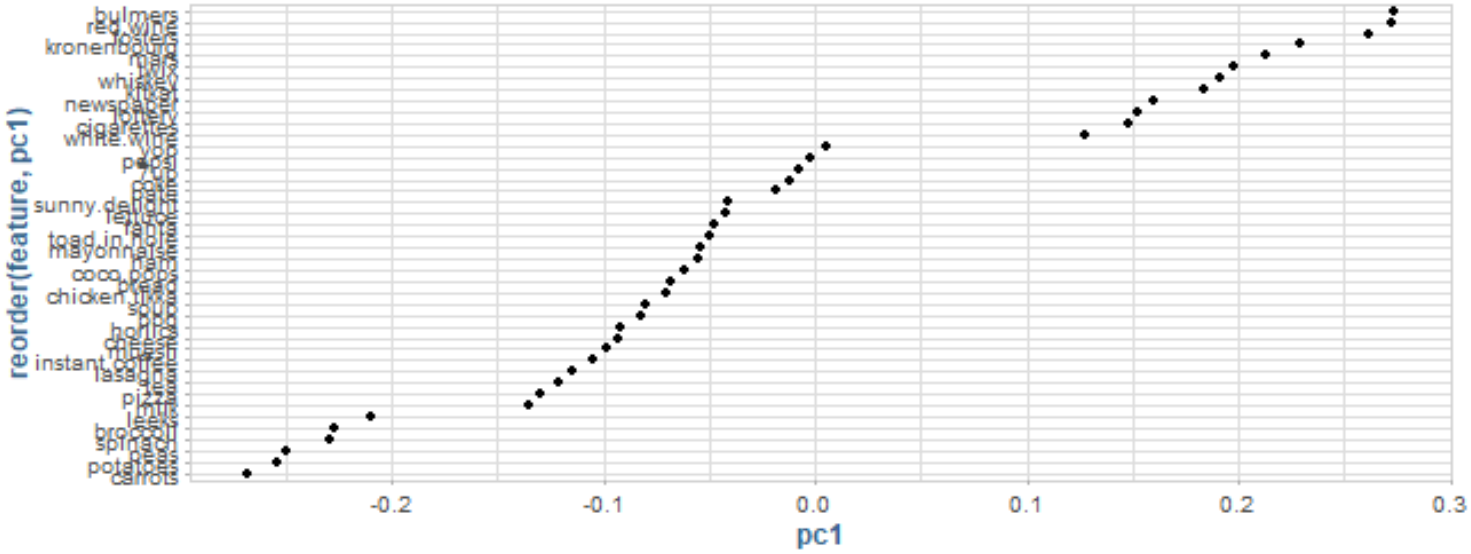
```
my_basket_h2o <- as.h2o(my_basket)

# PCA
my_pca <- h2o.prcomp(
  training_frame = my_basket_h2o,
  pca_method = "GramSVD",
  k = ncol(my_basket_h2o),
  transform = "STANDARDIZE",
  impute_missing = T,
  max_runtime_secs = 1000
)
```

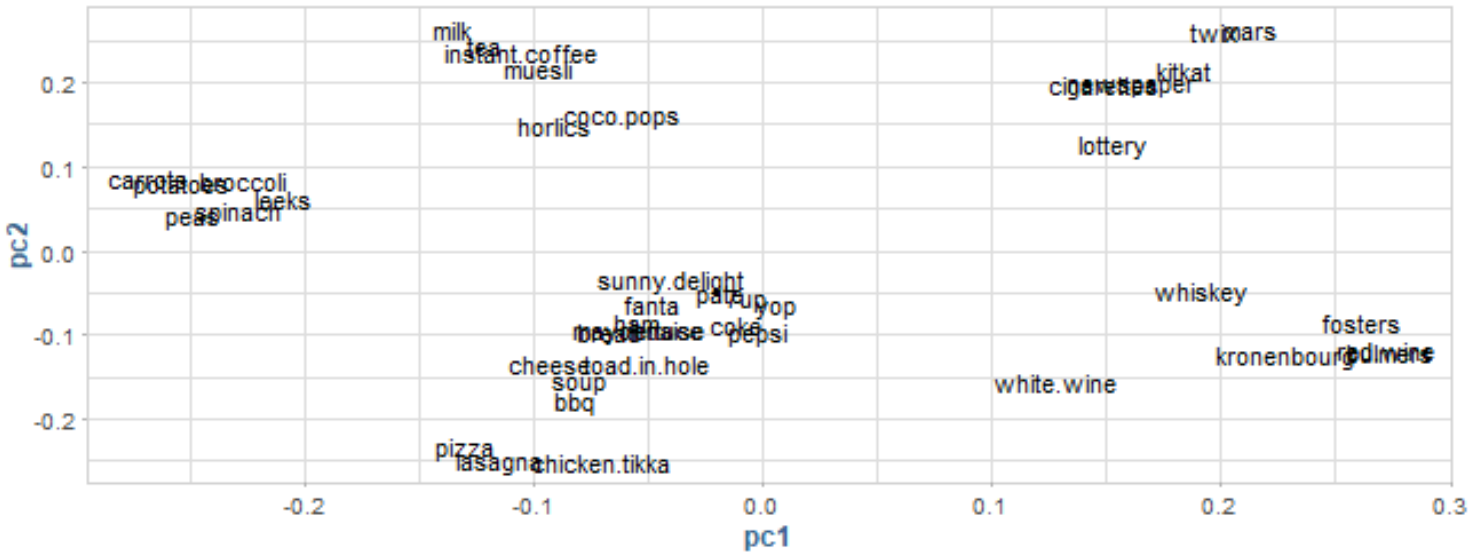
Loadings

```
my_pca@model$eigenvectors %>%
  as.data.frame() %>%
```

```
  mutate(feature = row.names(.)) %>%
  ggplot(aes(pc1, reorder(feature, pc1))) +
  geom_point()
```




```
my_pca@model$eigenvectors %>%
  as.data.frame() %>%
  mutate(feature = row.names(.)) %>%
  ggplot(aes(pc1, pc2, label = feature)) +
  geom_text()
```



Eigenvalue Criterion

```
eigen <- my_pca@model$importance["Standard deviation", ] %>%
  as.vector() %>%
```

```
  .^2

sum(eigen)
```

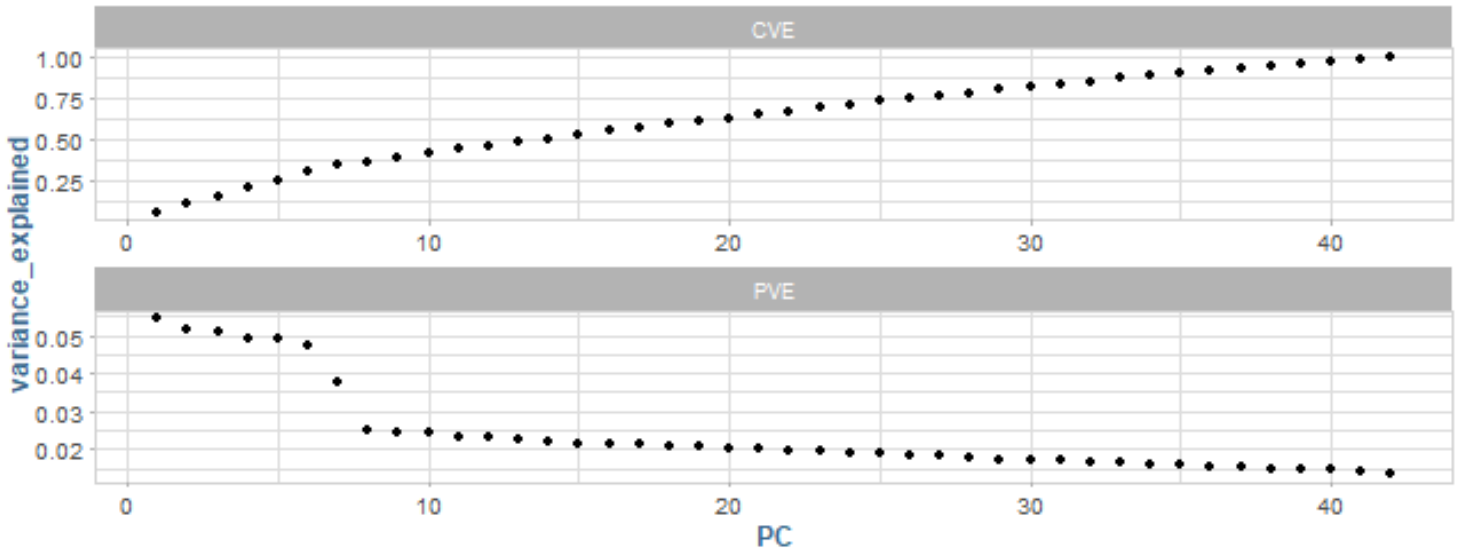
```
[1] 42
```

```
which(eigen >= 1)
```

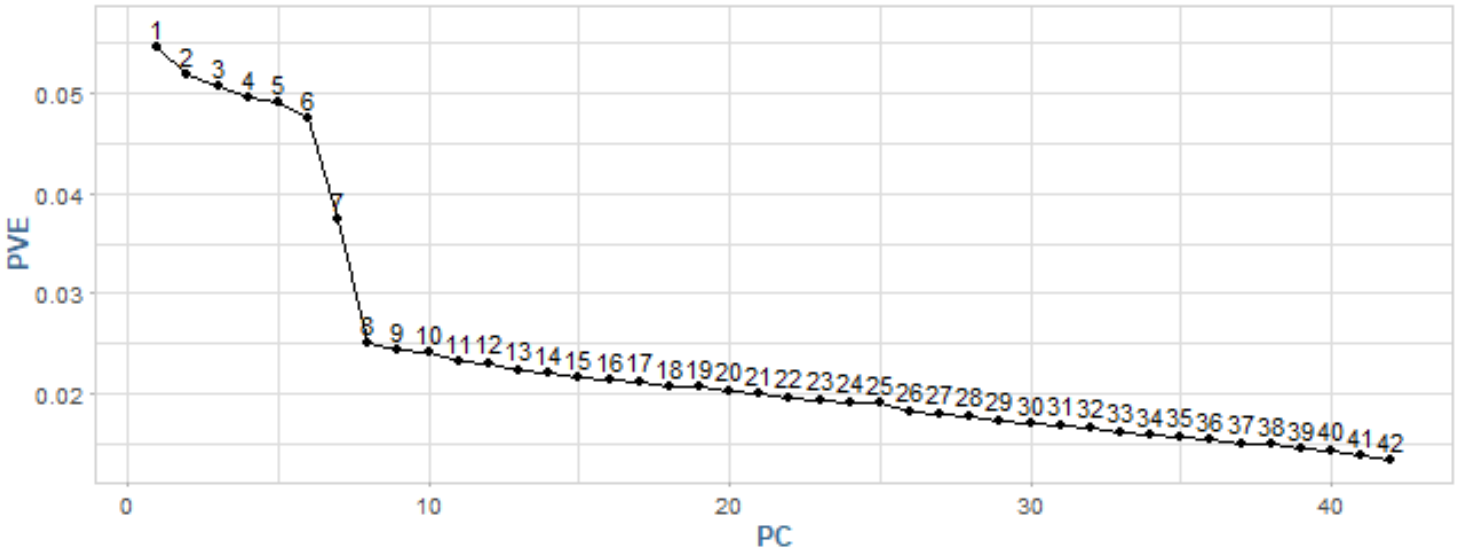
```
 [1]  1  2  3  4  5  6  7  8  9 10
```

Proportion of Variance Explained

```
data.frame(
  PC = my_pca@model$importance %>% seq_along(),
  PVE = my_pca@model$importance %>% .[2,] %>% unlist(),
  CVE = my_pca@model$importance %>% .[3,] %>% unlist()
) %>%
  tidyr::gather(metric, variance_explained, -PC) %>%
  ggplot(aes(PC, variance_explained)) +
  geom_point() +
  facet_wrap( ~ metric, ncol = 1, scales = "free")
```




```
data.frame(
  PC = my_pca@model$importance %>% seq_along,
  PVE = my_pca@model$importance %>% .[2,] %>% unlist()
) %>%
  ggplot(aes(PC, PVE, group = 1, label = PC)) +
  geom_point() +
  geom_line() +
  geom_text(nudge_y = 0.002)
```



Clean up

```
h2o.shutdown(prompt = FALSE)
```

```
# clean up
rm(list = ls())
```