




MULTIVARIATE ANALYSIS & STORYTELLING

Dr. Jingyuan Zhao



AGENDA (L9)

1

Linear Regression

- Pairwise plot using GGally & ggplot2
- Collinearity
- ANOVA test

2

Principal Component Analysis (PCA)

3

Model Selection

4

Model Explainability

5

Storytelling is important!

Linear Regression Model

Univariate

- Univariate: Univariate analysis is the simplest form of data analysis where the data being analyzed contains only one variable. Since it's a single variable it doesn't deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

It is used for the identification of insights, performance understanding & comparison.

Example: What's the SKU sales distribution? Symmetric? Long-tail?

month-to-month sales comparison

SKU sales comparison by category, by promotion

Bi-variate

- Bivariate: Bivariate analysis is used to find out if there is a relationship between two different variables. Something as simple as creating a scatterplot by plotting one variable against another can sometimes give you a picture of what the data is trying to tell you. If the data seems to fit a line or curve then there is a relationship or correlation between the two variables.

It is used to identify the drivers (business perspective), understand the relation and pre-processing variable if needed (technical perspective)

For example:

Your CMO wants to understand: Does promotion increase SKU sales?

Your customer engagement manager want to understand: Does product rating has any impact on SKU sales?

All these insights will help them to make a better decision/strategy in future. NOT ALL business problems are predictive analytics. 30% of your real-life work will focus on descriptive analytics

Multivariate

- Multivariate analysis: Multivariate analysis is the analysis of three or more variables. they can examine more complex phenomena and find data patterns that more accurately represent the real world.

It is used for a more accurate prediction.

For example:

What's the next month sales given some factors such as promotion, seller score?

House price data

```
> lm_data[1:5,]
  Dist_Taxi Dist_Market Dist_Hospital Carpet Builtup      Parking City_Category
1      9796       5250      10703    1659    1961         Open         CAT B
2      8294       8186      12694    1461    1752 Not Provided         CAT B
3     11001      14399      16991    1340    1609 Not Provided         CAT A
4      8301      11188      12289    1451    1748      Covered         CAT B
5     10510      12629      13921    1770    2111 Not Provided         CAT B
  Rainfall House_Price
1      530     6649000
2      210     3982000
3      720     5401000
4      620     5373000
5      450     4662000
```

What is the drivers for house price?

```
> lm5<-lm(House_Price~Carpet, data = lm_data)
> summary(lm5)

Call:
lm(formula = House_Price ~ Carpet, data = lm_data)

Residuals:
    Min       1Q   Median       3Q      Max
-4289482 -1285853  -65147   1265322  5366672

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 4948991.9   344711.8   14.357  < 2e-16 ***
Carpet         661.8      228.8     2.892  0.00392 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1713000 on 895 degrees of freedom
Multiple R-squared:  0.00926,    Adjusted R-squared:  0.008153
F-statistic: 8.365 on 1 and 895 DF,  p-value: 0.003917
```

Significant

Carpet is a driver of house price.

Regression model with all variables

```
lm1<-lm(House_Price~., data = lm_data)
summary(lm1)
```

P-value: Not significant

```
lm(formula = House_Price ~ ., data = lm_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-3573707	-805345	-61164	760782	4399519

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.595e+06	3.672e+05	15.235	< 2e-16 ***
Dist_Taxi	2.979e+01	2.682e+01	1.111	0.2670
Dist_Market	1.194e+01	2.080e+01	0.574	0.5659
Dist_Hospital	4.934e+01	3.008e+01	1.640	0.1013
Carpet	-5.242e+02	3.467e+03	-0.151	0.8799
Builtup	1.107e+03	2.893e+03	0.383	0.7021
ParkingNo Parking	-6.128e+05	1.387e+05	-4.419	1.11e-05 ***
ParkingNot Provided	-4.926e+05	1.235e+05	-3.990	7.16e-05 ***
ParkingOpen	-2.635e+05	1.126e+05	-2.341	0.0194 *
City_CategoryCAT B	-1.877e+06	9.599e+04	-19.554	< 2e-16 ***
City_CategoryCAT C	-2.895e+06	1.057e+05	-27.380	< 2e-16 ***
Rainfall	-9.953e+01	1.541e+02	-0.646	0.5185

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1222000 on 885 degrees of freedom

Multiple R-squared: 0.5014, Adjusted R-squared: 0.4952

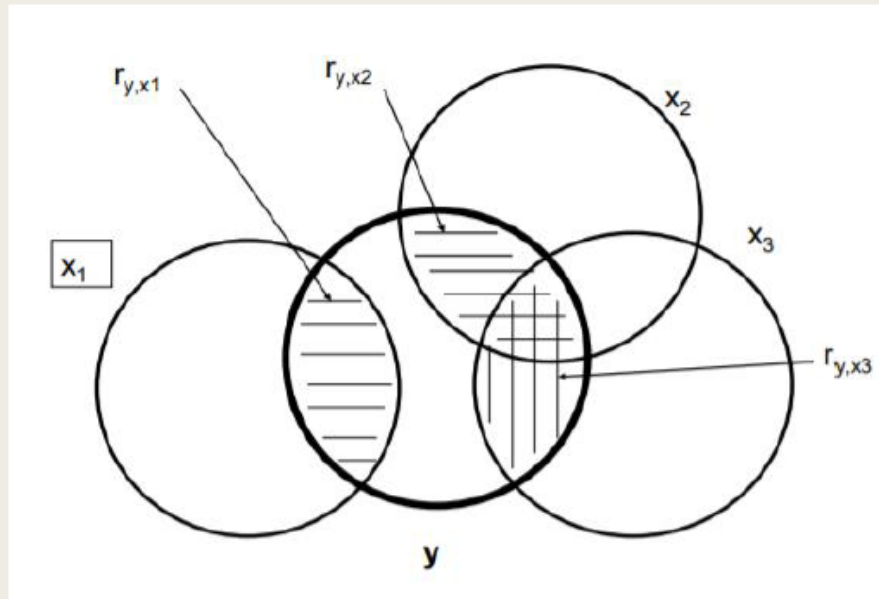
F-statistic: 80.89 on 11 and 885 DF, p-value: < 2.2e-16

Bivariate vs multivariate

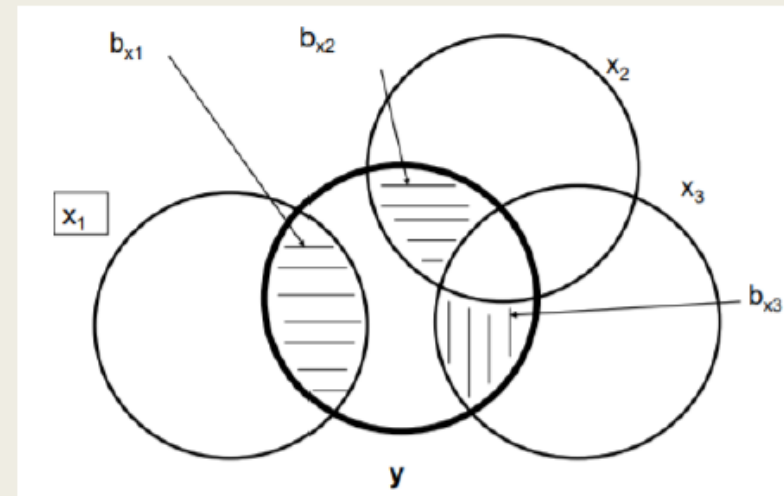
- a simple correlation tells the direction and strength of the linear relationship between two quantitative/binary variables
- a regression weight from a simple regression tells the expected change (direction and amount) in the criterion for a 1-unit change in the predictor
- a regression weight from a multiple regression model tells the expected change (direction and amount) in the criterion for a 1-unit change in that predictor,

Understand “Coefficients”

- the b of each predictor represents the part of that predictor shared with the criterion that is not shared with any other predictor – the unique contribution of that predictor to the model

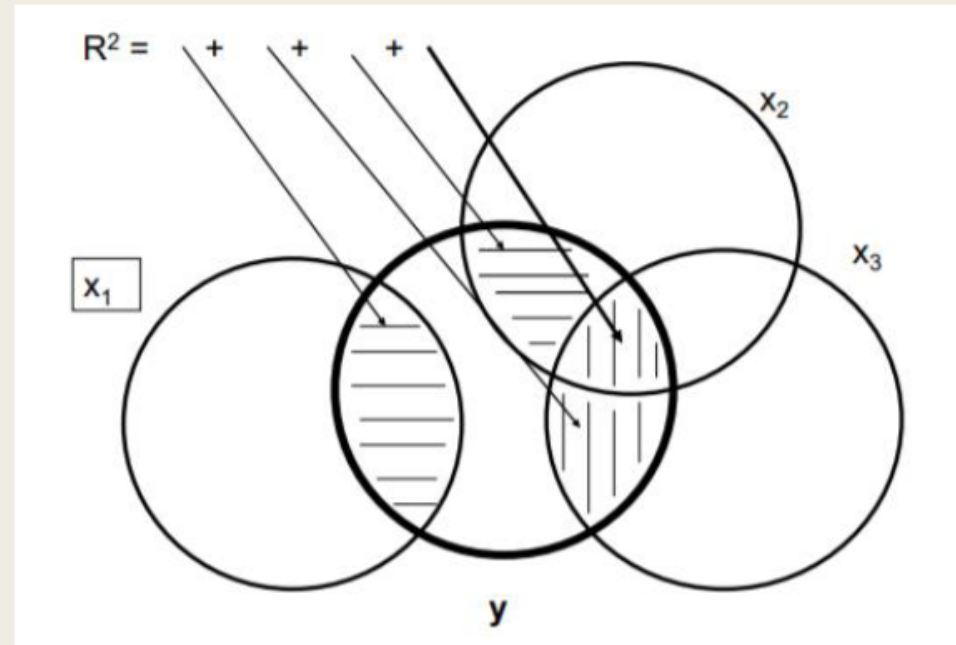


$$y' = b_1x_1 + b_2x_2 + b_3x_3 + a$$



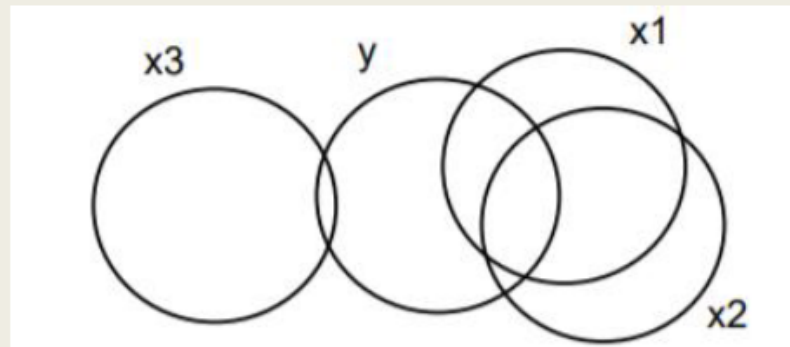
Understand “R²”

Remember R² is the total variance shared between the model (all of the predictors) and the criterion (not just the accumulation of the parts uniquely attributable to each predictor).



No contribution vs no association

- There are two different reasons that a predictor might not be contributing to a multiple regression model
 - *the variable isn't correlated with the criterion*
 - *the variable is correlated with the criterion, but is collinear with one or more other predictors, and so, has no independent contribution to the multiple regression model*



Bivariate vs. Multivariate Analyses & Interpretations

Perform both bivariate and multivariate analyses

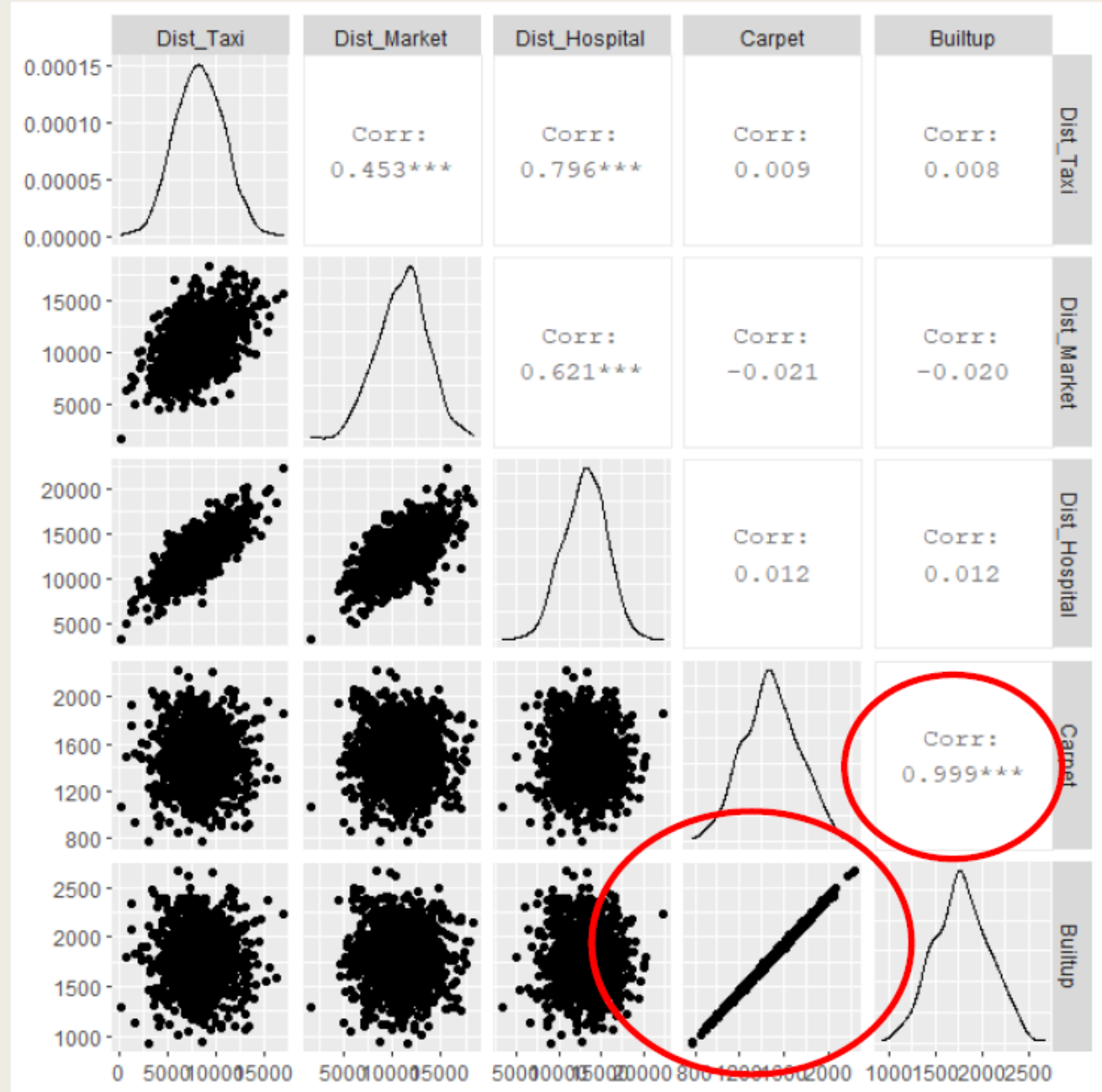
- correlations ask whether variables each have a relationship with the criterion
- bivariate regressions add information about the details of that relationship (how much change in Y for how much change in that X) -- Driver identification
- multivariate regressions tell whether variables have a unique contribution to a particular model. So, it is important to understand the different outcomes possible when performing both bivariate and multivariate analyses with the same set of predictors. - Prediction

Pairwise plot

```
library(ggplot2)
```

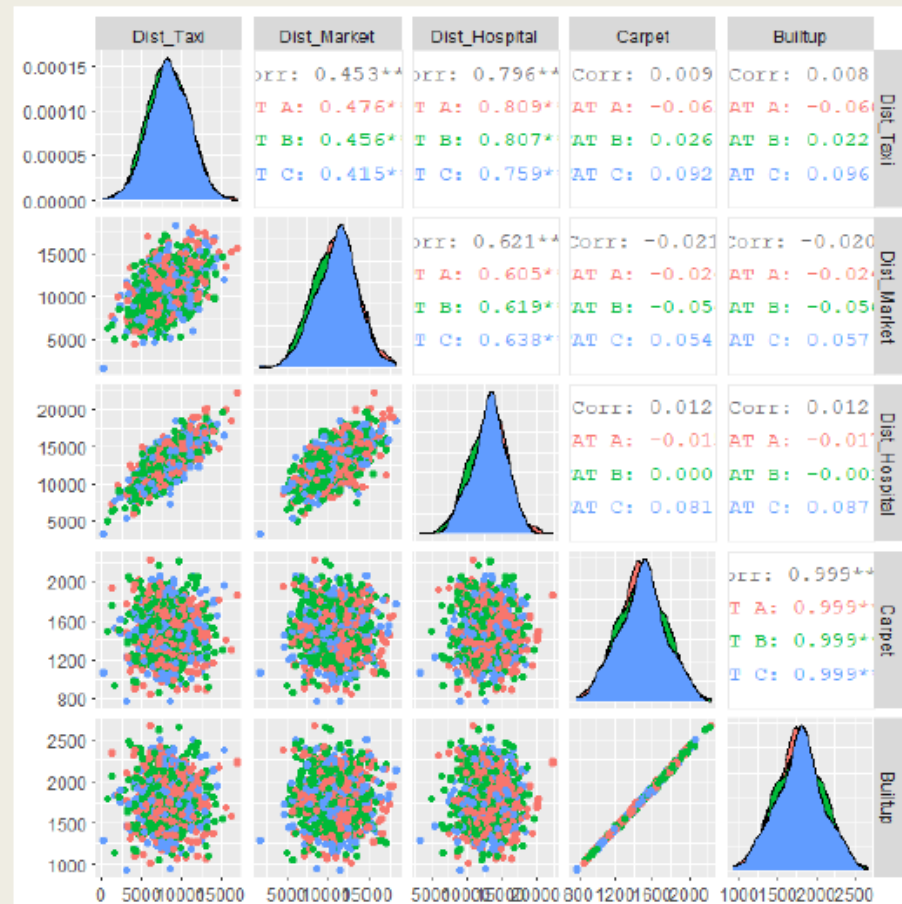
```
library(GGally) #extension of ggplot2,  
including pairwise scatterplot matrix
```

```
ggpairs(lm_data, columns=1:5)
```



With mapping color

```
ggpairs(lm_data, columns = 1:5, aes(color = City_Category))
```



Correlation

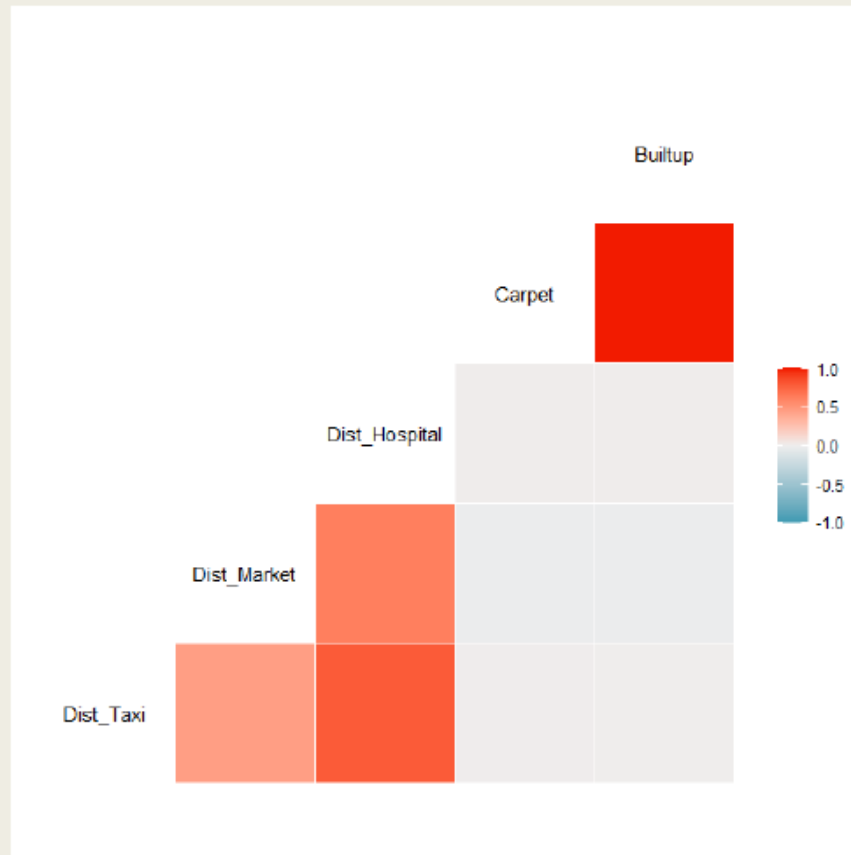
```
> cor(lm_data[,1:5])
```

	Dist_Taxi	Dist_Market	Dist_Hospital	Carpet	Builtup
Dist_Taxi	1.000000000	0.45347918	0.79552026	0.008702941	0.008230246
Dist_Market	0.453479183	1.00000000	0.62146637	-0.020778050	-0.020384289
Dist_Hospital	0.795520264	0.62146637	1.00000000	0.011706496	0.011960487
Carpet	0.008702941	-0.02077805	0.01170650	1.000000000	0.998885410
Builtup	0.008230246	-0.02038429	0.01196049	0.998885410	1.000000000

- Carpet and Builtup are highly correlated
- Dist_Taxi, Dist_Market, Dist_Hospital are associated

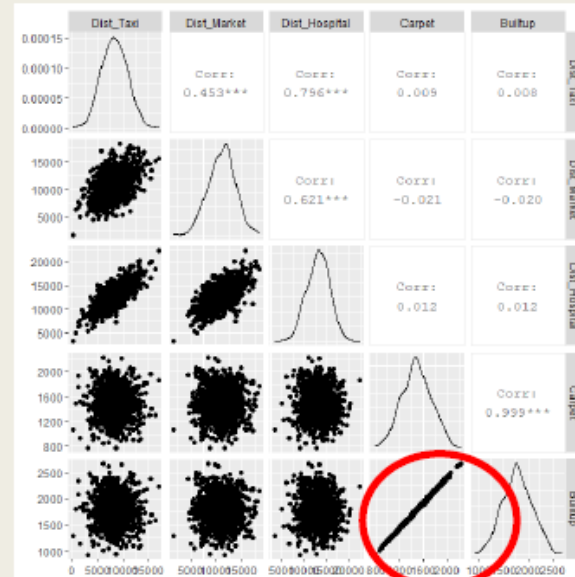
Visualization of correlation

```
ggcorr(lm_data[,1:5], method = c("everything", "pearson")) #GGally
```



Collinearity

- Multicollinearity occurs when independent variables in a regression model are correlated.
- The regression coefficients are not uniquely determined. In turn it hurts the interpretability of the model as then the regression coefficients are not unique and have influences from other features



Two major problems caused by collinearity

- The coefficient estimates can swing wildly based on which other independent variables are in the model. The coefficients become very sensitive to small changes in the model.
- Multicollinearity reduces the precision of the estimate coefficients, which weakens the statistical power of your regression model. You might not be able to trust the p-values to identify independent variables that are statistically significant.

Testing for Multicollinearity with Variance Inflation Factors (VIF)

- VIFs start at 1 and have no upper limit. A value of 1 indicates that there is no correlation between this independent variable and any others. VIFs between 1 and 5 suggest that there is a moderate correlation, but it is not severe enough to warrant corrective measures. VIFs greater than 5 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

```
> vif(lml) ##test collinearity
              GVIF Df GVIF^(1/(2*Df))
Dist_Taxi      2.754537  1      1.659680
Dist_Market    1.657549  1      1.287458
Dist_Hospital  3.574539  1      1.890645
Carpet         451.022120  1     21.237281
Builtup        451.077557  1     21.238587
Parking        1.037649  3      1.006179
City_Category  1.024015  2      1.005951
Rainfall       1.014681  1      1.007314
```

Library(car)

How to deal with Collinearity

The potential solutions include the following:

- Remove some of the highly correlated independent variables.
- Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.

```
lm(formula = House_Price ~ ., data = lm_data)
```

Residuals:

Min	1Q	Median	3Q	Max
-3573707	-805345	-61164	760782	4399519

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.595e+06	3.672e+05	15.235	< 2e-16 ***
Dist_Taxi	2.979e+01	2.682e+01	1.111	0.2670
Dist_Market	1.194e+01	2.080e+01	0.574	0.5659
Dist_Hospital	4.934e+01	3.008e+01	1.640	0.1013
Carpet	-5.242e+02	3.467e+03	-0.151	0.8799
Builtup	1.107e+03	2.893e+03	0.383	0.7021
ParkingNo Parking	-6.128e+05	1.387e+05	-4.419	1.11e-05 ***
ParkingNot Provided	-4.926e+05	1.235e+05	-3.990	7.16e-05 ***
ParkingOpen	-2.635e+05	1.126e+05	-2.341	0.0194 *
City_CategoryCAT B	-1.877e+06	9.599e+04	-19.554	< 2e-16 ***
City_CategoryCAT C	-2.895e+06	1.057e+05	-27.380	< 2e-16 ***
Rainfall	-9.953e+01	1.541e+02	-0.646	0.5185

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1222000 on 885 degrees of freedom

Multiple R-squared: 0.5014, Adjusted R-squared: 0.4952

F-statistic: 80.89 on 11 and 885 DF, p-value: < 2.2e-16

Call:

```
lm(formula = House_Price ~ ., data = lm_data[, -4])
```

Residuals:

Min	1Q	Median	3Q	Max
-3573721	-805639	-67929	756487	4407977

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.595e+06	3.670e+05	15.243	< 2e-16 ***
Dist_Taxi	2.971e+01	2.680e+01	1.108	0.2680
Dist_Market	1.196e+01	2.079e+01	0.575	0.5653
Dist_Hospital	4.941e+01	3.006e+01	1.644	0.1005
Builtup	6.698e+02	1.369e+02	4.892	1.19e-06 ***
ParkingNo Parking	-6.130e+05	1.386e+05	-4.423	1.09e-05 ***
ParkingNot Provided	-4.934e+05	1.233e+05	-4.001	6.82e-05 ***
ParkingOpen	-2.633e+05	1.125e+05	-2.341	0.0195 *
City_CategoryCAT B	-1.877e+06	9.592e+04	-19.571	< 2e-16 ***
City_CategoryCAT C	-2.896e+06	1.057e+05	-27.403	< 2e-16 ***
Rainfall	-9.960e+01	1.540e+02	-0.647	0.5180

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1222000 on 886 degrees of freedom

Multiple R-squared: 0.5013, Adjusted R-squared: 0.4957

F-statistic: 89.08 on 10 and 886 DF, p-value: < 2.2e-16

ANOVA test: nested model comparison

- $H_0 : [\beta_1 = 0 | \beta_0]$ is testing $\beta_1 = 0$ (or not) given that only the intercept β_0 is in the model
- $H_0 : [\beta_1 = 0 | \beta_0, \beta_2]$ is testing $\beta_1 = 0$ assuming that an intercept β_0 and a weight effect β_2 are in the model.

They make different assumptions, may reach different results.

The `anova` function, when given two (or more) different models, does an f-test by default.

Source	df	SS	MS
$\beta_2 \beta_0$	1	$SS(\beta_2 \beta_0)$	$SS(\beta_2 \beta_0) / 1$
$\beta_1 \beta_0, \beta_2$	1	$SS(\beta_1 \beta_0, \beta_2)$	$SS(\beta_1 \beta_0, \beta_2) / 1$
Error	$n - 3$	$\sum_{i=1}^n (y_i - \hat{y}_i)^2$	$SSE_{\text{Error}} / (n - 3)$
Total	$n - 1$	$\sum_{i=1}^n (y_i - \bar{y})^2$	

Fact: if H_0 is correct, $F = MS(\beta_1 | \beta_0, \beta_2) / MSE_{\text{Error}} \sim F_{1, n-3}$.

ANOVA test: nested model comparison

Remove “Carpet” variable and test the model difference

```
> anova(lm1,lm7)  #Model comparison
Analysis of Variance Table

Model 1: House_Price ~ Dist_Taxi + Dist_Market + Dist_Hospital + Carpet +
  Builtup + Parking + City_Category + Rainfall
Model 2: House_Price ~ Dist_Taxi + Dist_Market + Dist_Hospital + Builtup +
  Parking + City_Category + Rainfall
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1     885 1.3224e+15
2     886 1.3224e+15 -1 -3.4157e+10 0.0229 0.8799
```

All variables important?

overfitting problem

lose explainability

many non-informative features

Dimension Reduction

■ Feature Selection:

- *correlation/chi-square*
- *forward/backward/stepwise variable selection,*
- *LASSO/Elastic Net,*
- *Random Forest/Xgboost*

■ Feature Extraction:

- *Principal Component Analysis (PCA) – better when the numbers of sample per class is small and*
- *Linear Discriminant Analysis (LDA) – supervised learning, better with large sample size*

Feature extraction

- Create “new” independent variables, where each “new” independent variable is a combination of each of the “old” independent variables.
- keep as many of the new independent variables as we want, but drop the “least important ones.” to reduce the dimension

Principal Component Analysis (PCA)

Principal Component Analysis (PCA)

- 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
- The most significant variance is found on the first coordinate, and each subsequent coordinate is orthogonal to the last and has a lesser variance.
- Each principal component sums up a certain percentage of the total variation in the dataset.
- **Principal Components are independent**
- **Works for numerical variables**

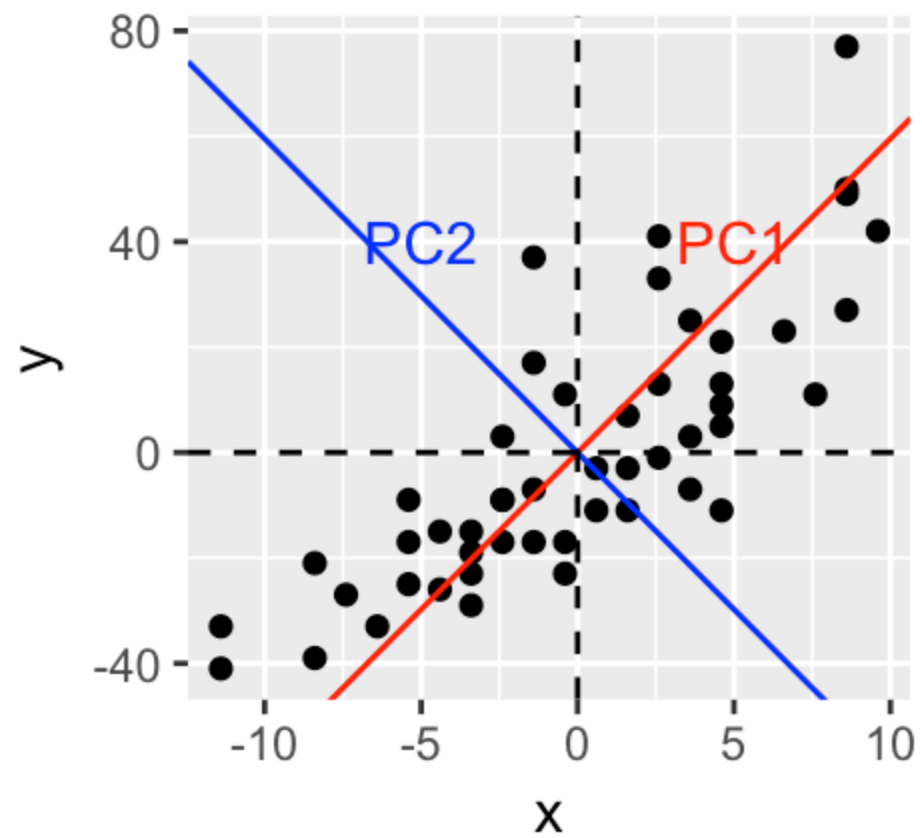
When to use PCA?

- The initial variables are strongly correlated with one another, you will be able to approximate most of the complexity in your dataset with just a few principal components.
- identify hidden pattern in a data set,
- reduce the dimension of the data by removing the noise and redundancy in the data,
- identify correlated variables

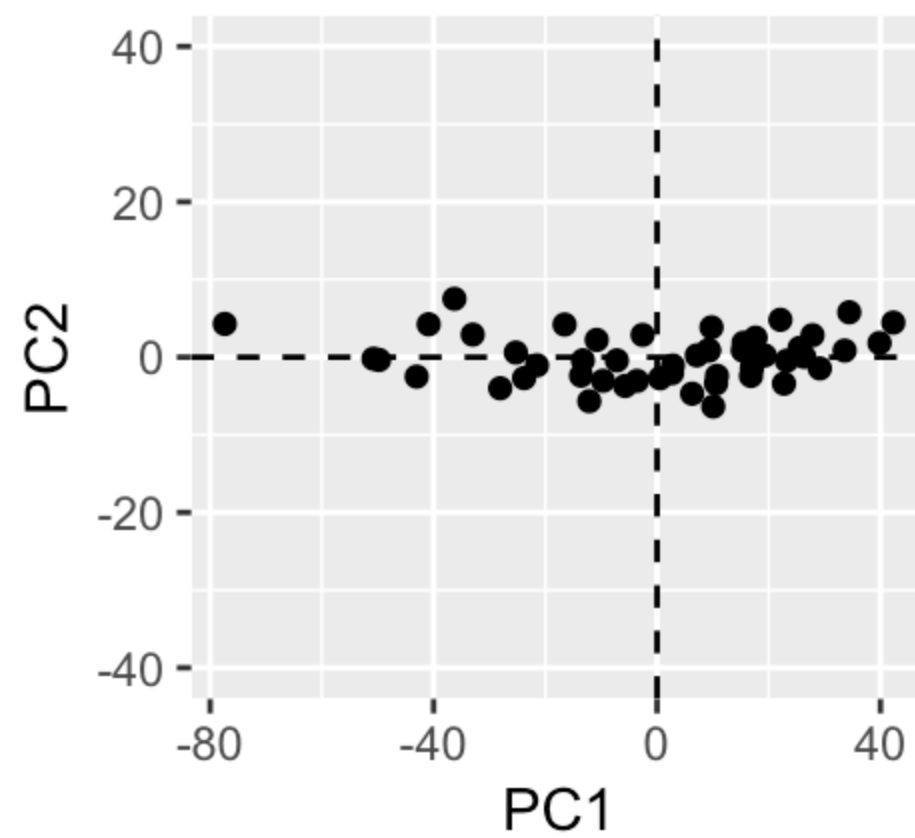
Eigenvector and Eigenvalue

- Eigenvectors, and eigenvalues come in pairs
- an eigenvector is a direction, while an eigenvalue is a number telling you how much variance there is in the data in that direction (measure the amount of variance)
- The eigenvector with the highest eigenvalue is, therefore, the first principal component.
- The number of eigenvalues and eigenvectors that exists is equal to the number of dimensions the data set has.
- reframe a dataset in terms of these eigenvectors and eigenvalues without changing the underlying information.

Plot 1A



Plot 1B



Data Standardization

- In principal component analysis, variables are often scaled (i.e. standardized). This is particularly recommended when variables are measured in different scales (e.g: kilograms, kilometers, centimeters, ...); otherwise, the PCA outputs obtained will be severely affected.
- The goal is to make the variables comparable. Generally variables are scaled to have i) standard deviation one and ii) mean zero.
- The standardization of data is an approach widely used in the context of gene expression data analysis before PCA and clustering analysis.

Compute Principal Components in R

- Performs a principal components analysis on the given data matrix and returns the results as an object of class prcomp

Prcomp()

prcomp(x, retx = TRUE, center = TRUE, scale. = FALSE, tol = NULL, rank. = NULL, ...)

predict(object, newdata, ...)

Visualize Principal Component Analysis using R package factoextra

- Principal component analysis (PCA) reduces the dimensionality of multivariate data, to two or three that can be visualized graphically with minimal loss of information. `fviz_pca()` provides ggplot2-based elegant visualization of PCA outputs from: i) `prcomp` and `princomp` [in built-in R stats], ii) PCA [in FactoMineR], iii) `dudi.pca` [in `ade4`] and `epPCA` [ExPosition].

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

```
library(factoextra)
```

`fviz_pca_ind()`: Graph of individuals

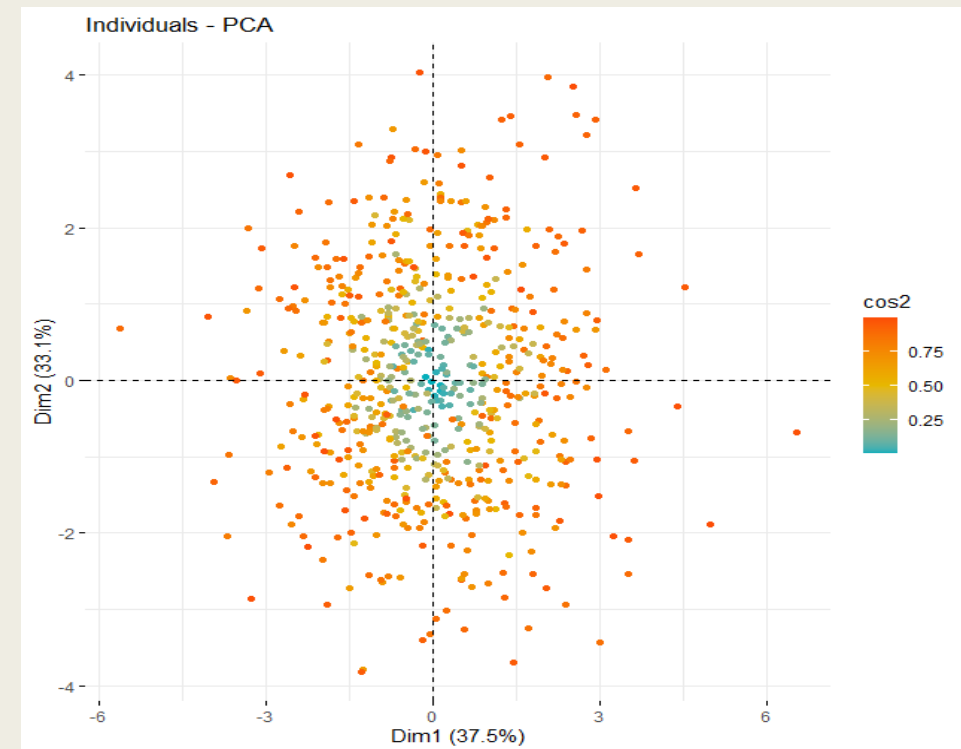
`fviz_pca_var()`: Graph of variables

`fviz_pca_biplot()`: Biplot of individuals and variables

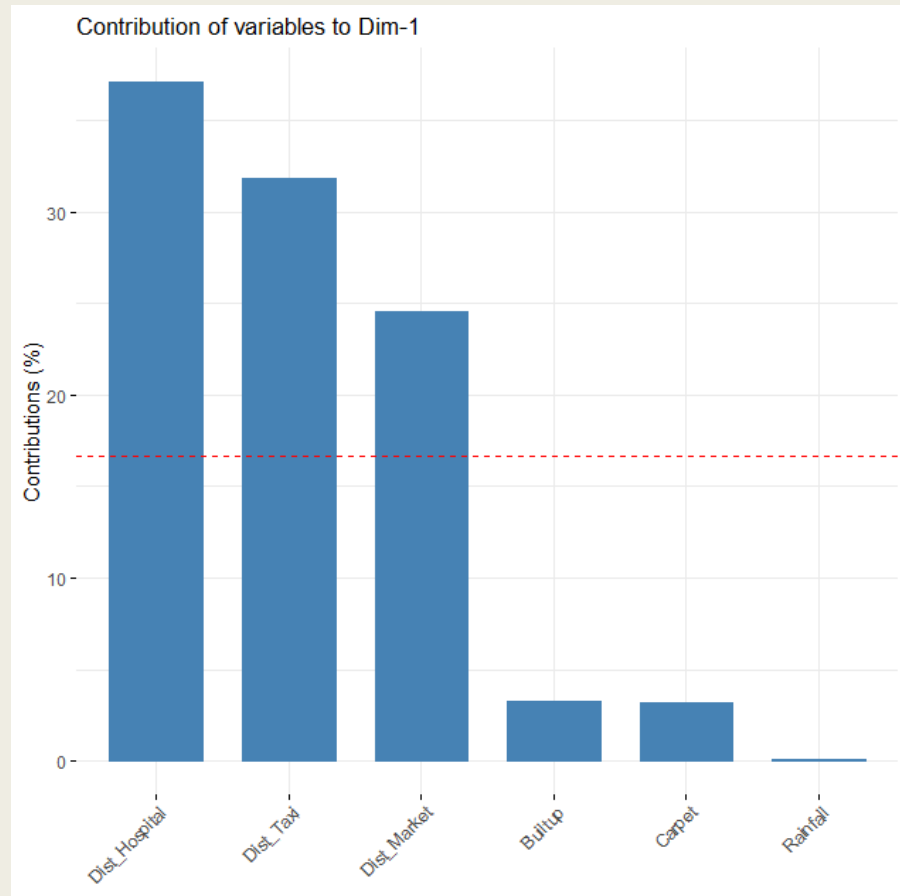
`fviz_pca()`: An alias of `fviz_pca_biplot()`

Graph of individuals

```
fviz_pca_ind(pca,  
  axes = c(1,2), #a numeric vector of length 2 specifying the  
  dimensions to be plotted.  
  geom = "point", # plot type  
  col.ind = "cos2", # Color by the quality of representation  
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),  
  repel = TRUE    # Avoid text overlapping  
)
```



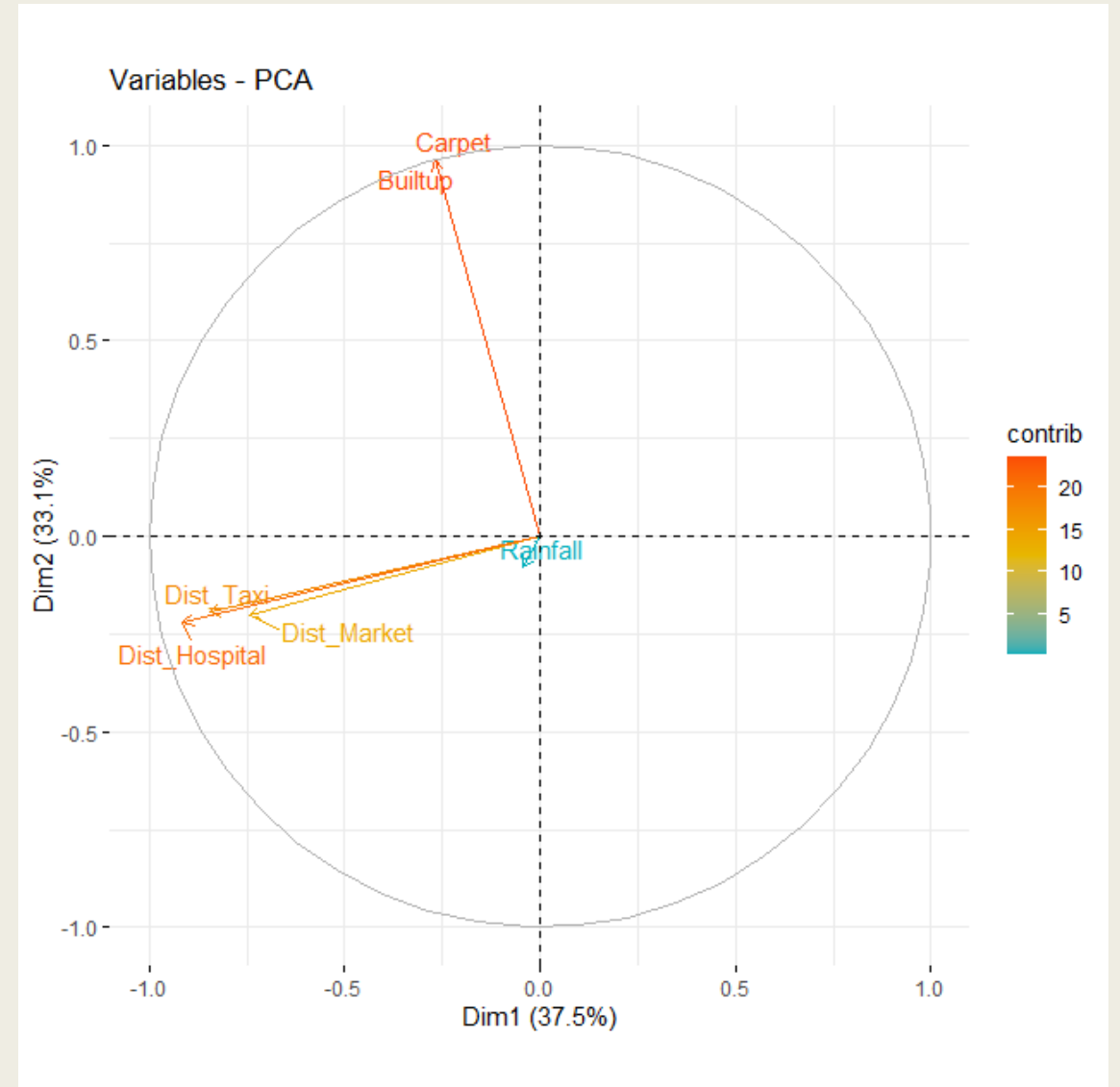
Contributions of variables to PCs



```
fviz_contrib(pca, choice = "var", axes = 1, top = 10)
```

Graph of variables: Correlation circle

```
fviz_contrib(pca, choice = "var", axes = 1:2, top = 10)
```



How many Principal Components we should keep?

- Eigenvalue criterion (Kaiser Rule)
- Proportion of variance explained criterion
- Scree plot criterion

1. Kaiser Rule (Kaiser 1961)

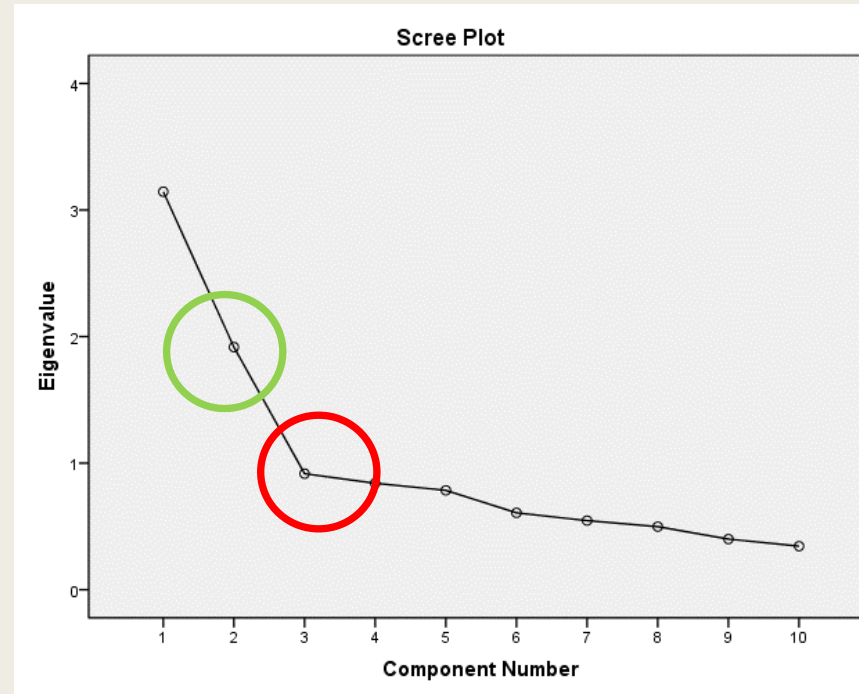
- The more variables that load onto a particular component (i.e., have a high correlation with the component), the more important the factor is in summarizing the data.
- An eigenvalue is an index that indicates how good a component is as a summary of the data. An eigenvalue of 1.0 means that the factor contains the same amount of information as a single variable

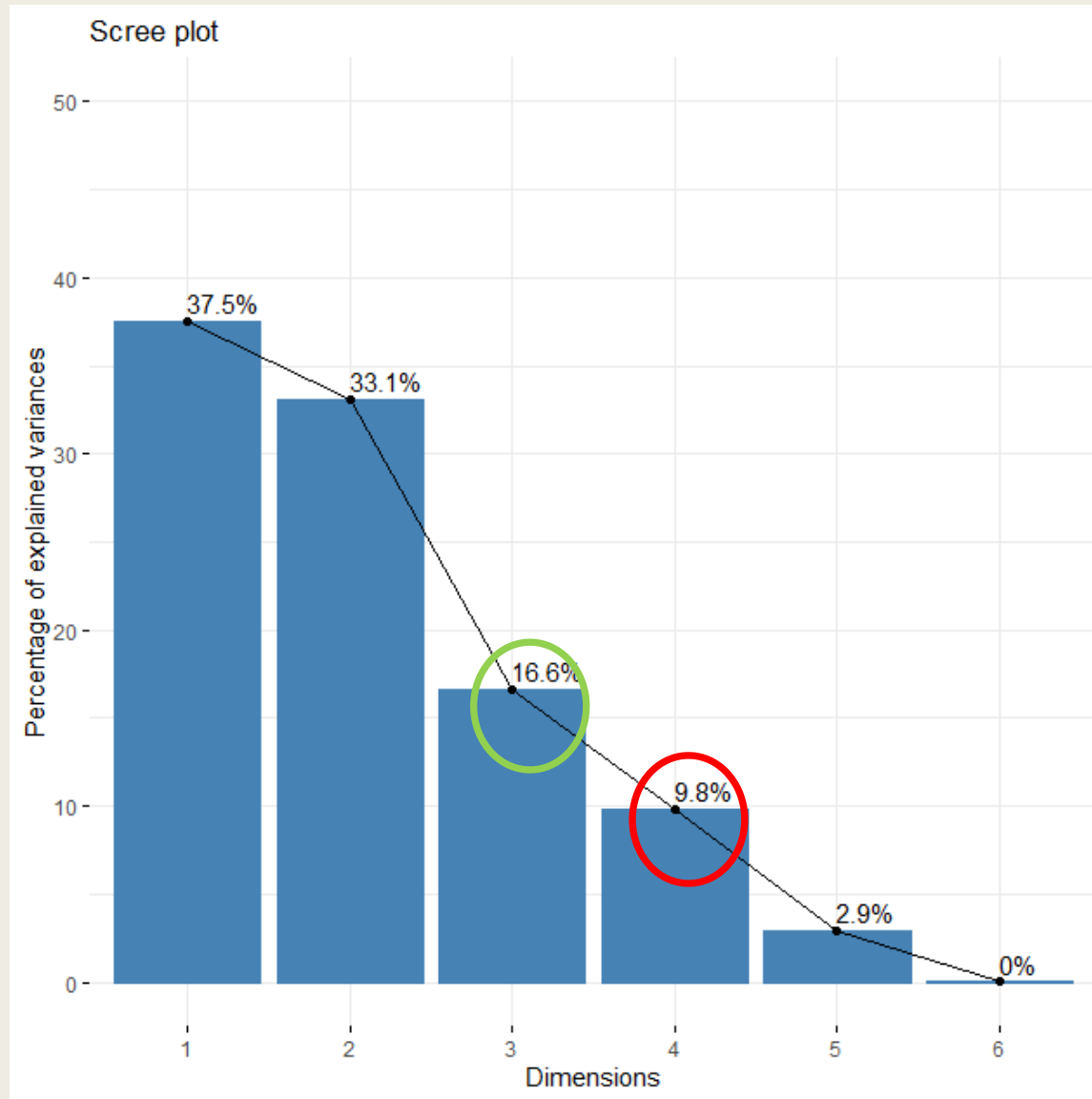
get_eigenvalue(pca)

```
> get_eigenvalue(pca)
  eigenvalue variance.percent cumulative.variance.percent
Dim.1 2.252151211      37.53585352           37.53585
Dim.2 1.984194431      33.06990718           70.60576
Dim.3 0.997365297      16.62275496           87.22852
Dim.4 0.590365878       9.83943131           97.06795
Dim.5 0.174802287       2.91337145           99.98132
Dim.6 0.001120895       0.01868159          100.00000
```

2. Scree plot

- The following scree *plot* shows the number of [Eigenvalues](#) from the example shown on the main [principal components analysis page](#), ordered from biggest to smallest. Some researchers conclude that the correct number of components is the number that appear prior to the *elbow*





```
fviz_eig(pca,addlabels = TRUE, ylim = c(0, 50))
```

3. Proportion of variance explained

- 70% of variance –acceptable

```
> get_eigenvalue(pca)
```

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	2.252151211	37.53585352	37.53585
Dim.2	1.984194431	33.06990718	70.60576
Dim.3	0.997365297	16.62275496	87.22852
Dim.4	0.590365878	9.83943131	97.06795
Dim.5	0.174802287	2.91337145	99.98132
Dim.6	0.001120895	0.01868159	100.00000

Results for variables

```
# Results for Variables  
res.var <- get_pca_var(pca)  
res.var$coord      # Coordinates  
res.var$contrib     # Contributions to the PCs
```

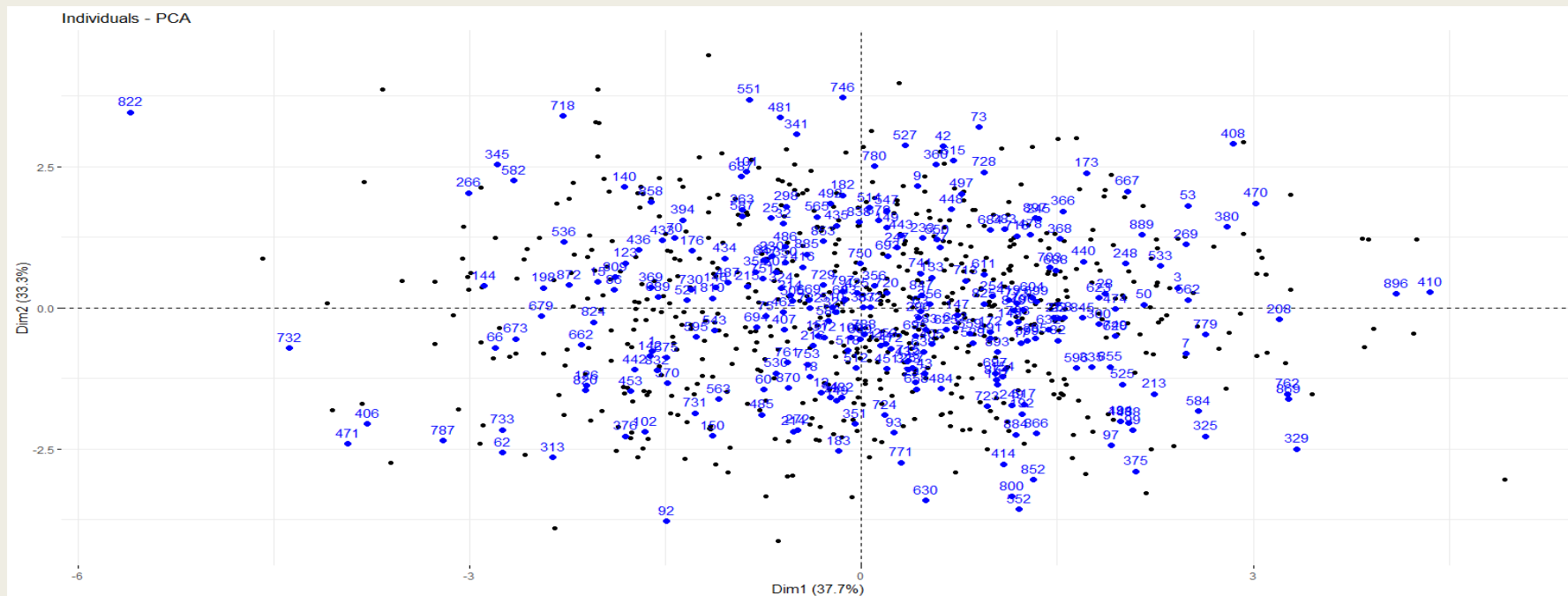
Results for individuals

```
# Results for individuals  
res.ind <- get_pca_ind(pca)  
res.ind$coord      # Coordinates → new sets of variables  
res.ind$contrib     # Contributions to the PCs
```

New sample projection

```
pred <- predict(pca, newdata=lm_data[test,c(1:5,8)])
```

```
p <- fviz_pca_ind(pca, geom = "point", repel = T)
fviz_add(p, pred, color = "blue")
```



Model Selection

Training & Testing data sets

1. Reserve a small sample of the data set as your testing data
2. Build (or train) the model using the remaining part of the data set
3. Test the effectiveness of the model on the reserved sample of the data set. If the model works well on the test data set, then it's good.

How to measure the model accuracy

1. **R-squared (R2)**, representing the squared correlation between the observed outcome values and the predicted values by the model. The higher the adjusted R2, the better the model.
2. **Root Mean Squared Error (RMSE)**

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

3. **Mean Absolute Error (MAE)**

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

Good or Good luck?

- Build a model on a fraction of the data set only, possibly leaving out some interesting information about data, leading to higher bias.
- The test error rate can be highly variable, depending on which observations are included in the training set and which observations are included in the validation set.
- How to Solve? –Repeat multiple times to get a more reliable result! (K-fold cross-validation)

K-fold cross-validation

- The k-fold cross-validation method evaluates the model performance on different subset of the training data and then calculate the average prediction error rate. The algorithm is as follow:
 1. *Randomly split the data set into k -subsets (or k -fold) (for example 5/10 subsets)*
 2. *Reserve one subset and train the model on all other subsets*
 3. *Test the model on the reserved subset and record the prediction error*
 4. *Repeat this process until each of the k subsets has served as the test set.*
 5. *Compute the average of the k recorded errors. This is called the cross-validation error serving as the performance metric for the model.*
- K-fold cross-validation (CV) is a **robust** method for estimating the accuracy of a model.

Regression model with all variables

```
lm1<-lm(House_Price~., data = lm_data_training) #regression model using all the variables
summary(lm1)
estimate<-predict(lm1,type='response',newdata=lm_data_testing) # prediction
observed = lm_data_testing$House_Price #observed outcome
format(cor(estimate,observed)^2,digits=4) #R2
```

```
[1] "0.4848"
```

Regression model with top variables

```
> cor(lm_data[,1:5])
```

	Dist_Taxi	Dist_Market	Dist_Hospital	Carpet	Builtup
Dist_Taxi	1.000000000	0.45347918	0.79552026	0.008702941	0.008230246
Dist_Market	0.453479183	1.00000000	0.62146637	-0.020778050	-0.020384289
Dist_Hospital	0.795520264	0.62146637	1.00000000	0.011706496	0.011960487
Carpet	0.008702941	-0.02077805	0.01170650	1.00000000	0.998885410
Builtup	0.008230246	-0.02038429	0.01196049	0.998885410	1.00000000

Select 2
variables
from 5

```
lm2<-lm(House_Price~.,data =lm_data_training[,c(3,4,6,7,8,9)]) # Remove the highly correlated variables:
```

Dist_Hospital, Carpet, Rainfall and all categorical variables

```
summary(lm2)
```

```
estimate<-predict(lm2,type='response',newdata=lm_data_testing)
```

```
observed = lm_data_testing$House_Price
```

```
format(cor(estimate,observed)^2,digits=4)
```

```
[1] "0.4985"
```

Regression model with significant PCs

```
lm3<-lm(House_Price~., data = lm_data_training_new) #regression model using all the variables
summary(lm3)
estimate<-predict(lm3,type='response',newdata=lm_data_testing_new)
observed = lm_data_testing_new$House_Price
format(cor(estimate,observed)^2,digits=4)
```

```
[1] "0.5004"
```

Doing k-fold cross-validation in R using caret package

```
install.packages("caret")
```

```
library(caret)
```

```
train.control <- trainControl(method = "cv", number = 10) #10-fold cv
```


Model selection using k-fold cv

```
> ##Full model##
> model <- train(House_Price ~., data = lm_data, method = "lm",
+               trControl = train.control) #Training and test model
> print(model)
Linear Regression

897 samples
  8 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 806, 806, 807, 807, 808, 807, ...
Resampling results:
```

RMSE	Rsquared	MAE
1233220	0.4918937	976834

Tuning parameter 'intercept' was held constant at a value of TRUE

```
> ##With selected variables ONLY
> model2<- train(House_Price ~., data = lm_data[,c(3,4,6,7,8,9)], method = "lm",
+               trControl = train.control)
> print(model2)
Linear Regression

897 samples
  5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 808, 808, 807, 808, 807, 807, ...
Resampling results:
```

RMSE	Rsquared	MAE
1228613	0.4931574	972980.7

Tuning parameter 'intercept' was held constant at a value of TRUE

```
> ##With top 3 principal components
> model3<- train(House_Price ~., data = lm_data_new, method = "lm",
+               trControl = train.control)
> print(model3)
Linear Regression

897 samples
  5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 808, 807, 808, 807, 807, 808, ...
Resampling results:
```

RMSE	Rsquared	MAE
1227059	0.4941013	970984.5

Tuning parameter 'intercept' was held constant at a value of TRUE

Your final prediction model

- After model selection, we have chosen the best model based on accuracy metrics such as R^2 /MAE/RMSE.
- Here, the best model means the best set of variables.
- What is your Final Prediction model?

Short answer: Train the final model using all the data points and the best set of variables you have chosen by k-fold CV.

Model Explainability

Why need to explain models?

- If your model results will drive a critical action, the model could not be black-box.

For example, in Tax Compliance space, using models to automatically score millions of individual tax returns to identify suspicious ones for further inspection.

- Any other examples from you?

Model Explainability (1) - interpretable models

- Interpretable model is the model who explain themselves
- Choose models such as regression model, tree
- Need to balance between accuracy and explainability

Model Explainability (2) - Model-agnostic methods

- Most ML models could not explain themselves
- model-agnostic method could be used to explain any model for any machine learning model, from support vector machines to neural networks
- Two popular methods: Local Interpretable Model-agnostic Explanations (LIME) and Shapley Additive Explanations (SHAP).
- Both have R and Python package

<https://christophm.github.io/interpretable-ml-book/lime.html>

<https://towardsdatascience.com/model-agnostic-methods-for-interpreting-any-machine-learning-model-4f10787ef504>

Shapley value

- Global explanation

Coefficients, Gini value to explain feature importance

- Local explanation with Shapley value using game theory

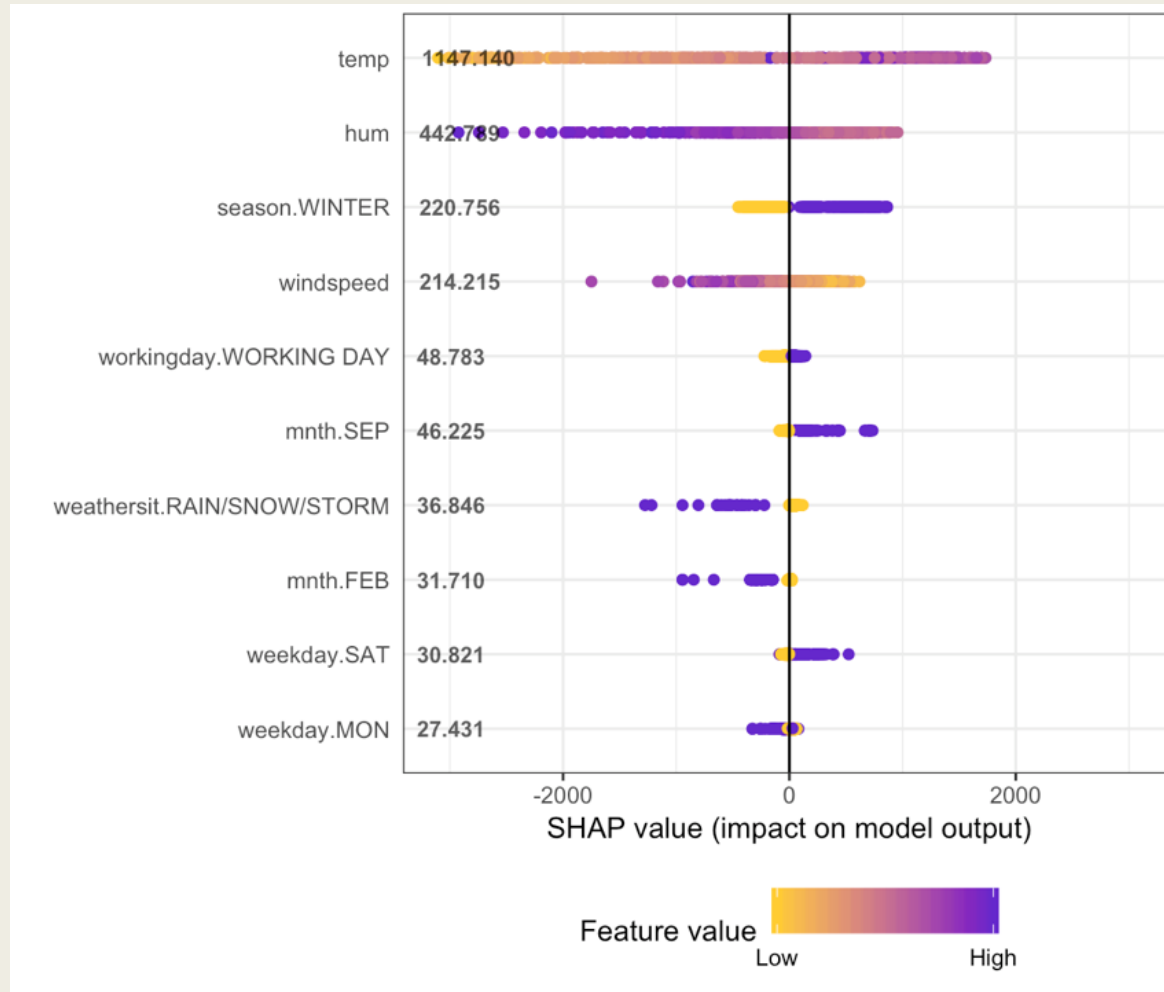
Shapley Value – SHAP (SHapley Additive exPlanations) developed by Scott M. Lundberg

Shapley computes feature contributions for single predictions with the Shapley value, an approach from cooperative game theory. The features values of an instance cooperate to achieve the prediction. The Shapley value fairly distributes the difference of the instance's prediction and the datasets average prediction among the features.

- If the original data has 200 samples and 10 variables, the shap value table will have the same dimension (200 x 10).

<https://christophm.github.io/interpretable-ml-book/shapley.html>

SHAP summary plot



- The y-axis indicates the variable name, in order of importance from top to bottom. The value next to them is the mean SHAP value.
- On the x-axis is the SHAP value. Indicates how much is the change in log-odds. From this number we can extract the probability of success.
- Gradient color indicates the original value for that variable. In booleans, it will take two colors, but in number it can contain the whole spectrum.
- Each point represents a row from the original dataset.

Storytelling

Why storytelling is essential?

A study by Stanford professor Chip Heath (Made to Stick author) found 63% could remember stories, but only 5% could remember a single statistic. While 2.5 statistics were used on average in the exercise and only 10% of the participants incorporated a story, the stories are what caught people's attention.

<https://www.forbes.com/sites/brentdykes/2016/03/31/data-storytelling-the-essential-data-science-skill-everyone-needs/#68a1d31d52ad>

How to prepare a story?

1. Know the audience and adapt the story to their needs
2. Understand the business problem
3. Identify the probable questions and preparing answers
4. Get the right data at hand

What should be included in your story

Step 1: State your assumptions

Step 2: Lay the groundwork

Step 3: Explain your findings for the first part of your analysis

Step 4: Present and explain your second piece of analytics

Step 5: Summarize your findings

Step 6: Link your findings

Apply Amazon Star interview technique to create your storyline!

The **STAR method** is a structured manner of responding to a behavioral-based **interview** question by discussing **the specific situation, task, action, and result of the situation you are describing**.

- **Situation:** Describe the situation that you were in or the task that you needed to accomplish. You must describe a specific event or situation, not a generalized description of what you have done in the past. Be sure to give enough detail for the interviewer to understand.
- **Task:** What goal were you working toward?
- **Action:** Describe the actions you took to address the situation with an appropriate amount of detail and keep the focus on YOU. What specific steps did you take and what was your particular contribution? Be careful that you don't describe what the team or group did when talking about a project, but what you actually did. Use the word "I," not "we" when describing actions.
- **Result:** Describe the outcome of your actions and don't be shy about taking credit for your behavior. What happened? How did the event end? What did you accomplish? What did you learn? Make sure your answer contains multiple positive results.