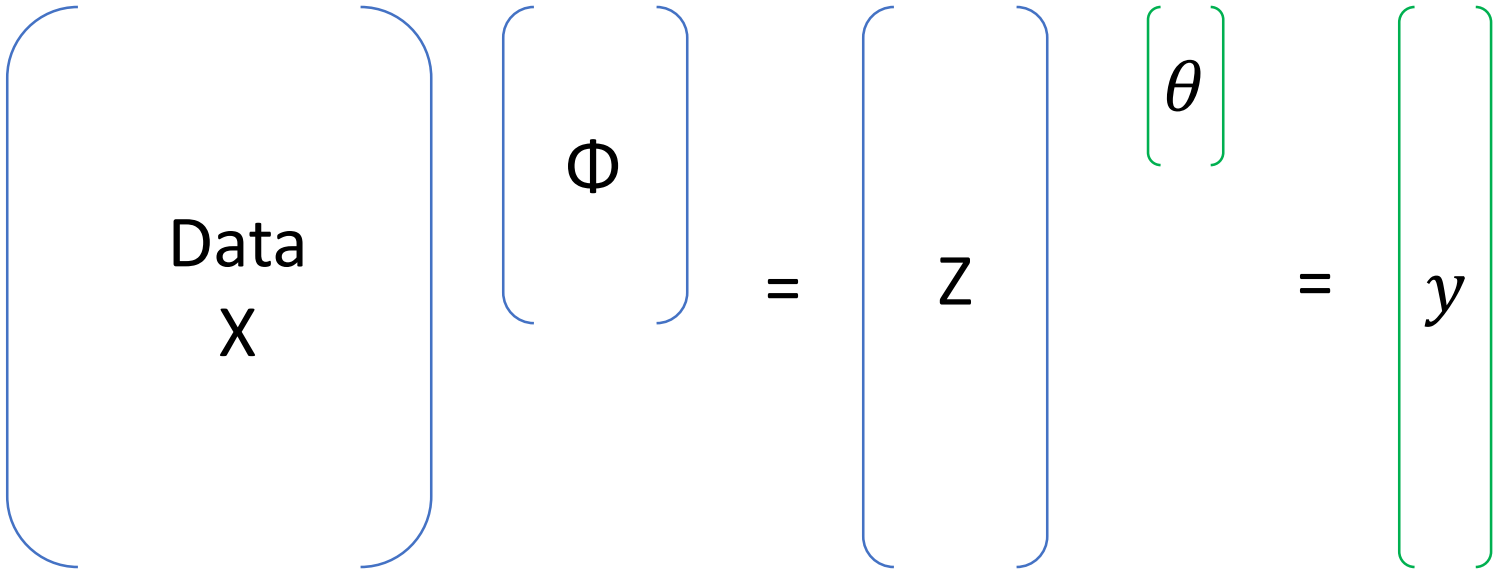
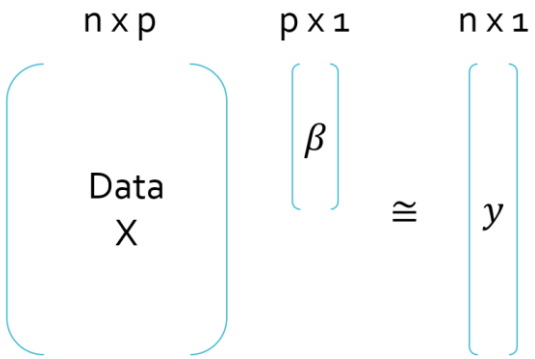
Reduce Dimension



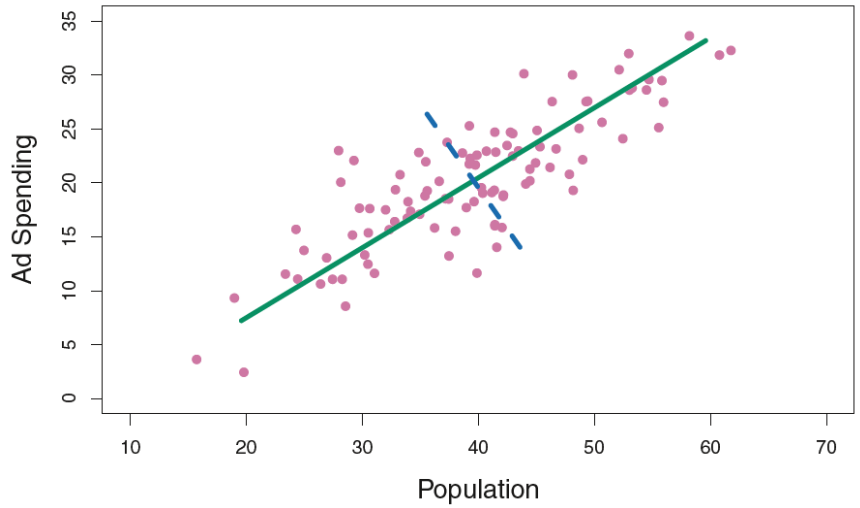
Linear Regression

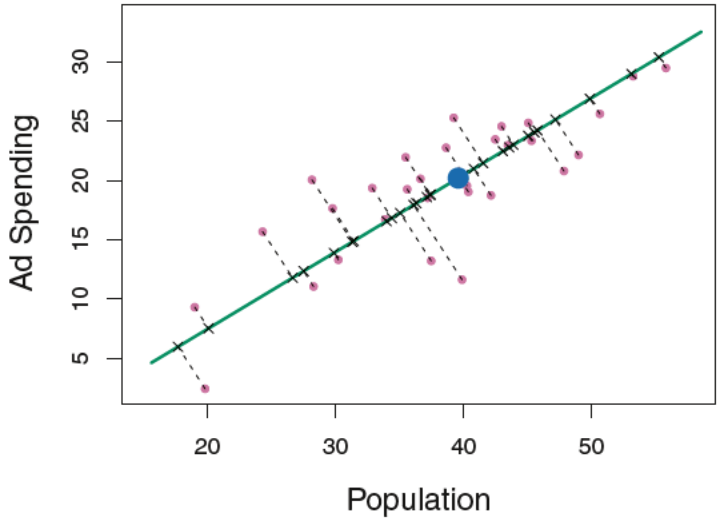


Learn coefficient vector β such that $\|X\beta - y\|^2$ is minimal.

- Compress X into Z, then learn coefficient θ such that $\|Z\theta - y\|^2$ is minimal
- Since m < p, we reduced the number of predictors
- Require: Z contains most of the information in X
- *Principle Component Regression*: Z is set to be the first m principal components of X

Principal Component Analysis (PCA)





- Each principal component is a linear combination of the original variables.
- The first principal component direction (green line) is that along which the observations vary the most. The green line is the line *closest* (in squared perpendicular distance) to the data.
- The second principal component is orthogonal to the first (blue dashed line)

Principle Component Regression (PCR)

- In general, there are p distinct principal components for X, the first m will capture most of the information in X
- The best m can be determined by CV
- PCR is not a feature selection method
  - Each of the m principal components is a linear combination of all p of the original features
- When performing PCR, *standardizing* each predictor is recommended.

```r
cars <- read.csv(file = "cars.csv",
                 header = T, row.names = 1)

res.pca <- prcomp(cars[, 1:9], scale = T)  # Perform PCA on numeric columns
sum.pca <- summary(res.pca)
barplot(sum.pca$importance[2,], xlab = "Principal Component", ylab = "Proportion of Variance Explained")
plot(sum.pca$importance[3,], type='b', xlab = "Dimension", ylab = "Cum. Prop. of Variance Explained")

biplot(res.pca)

library(ggfortify)
autoplot(res.pca, data = cars, colour = 'origin')
autoplot(res.pca, data = cars, colour = 'origin', label = T)
autoplot(res.pca, data = cars, colour = 'origin', loadings = T, loadings.label = T)
```