THE OHIO STATE UNIVERSITY
FISHER COLLEGE OF BUSINESS

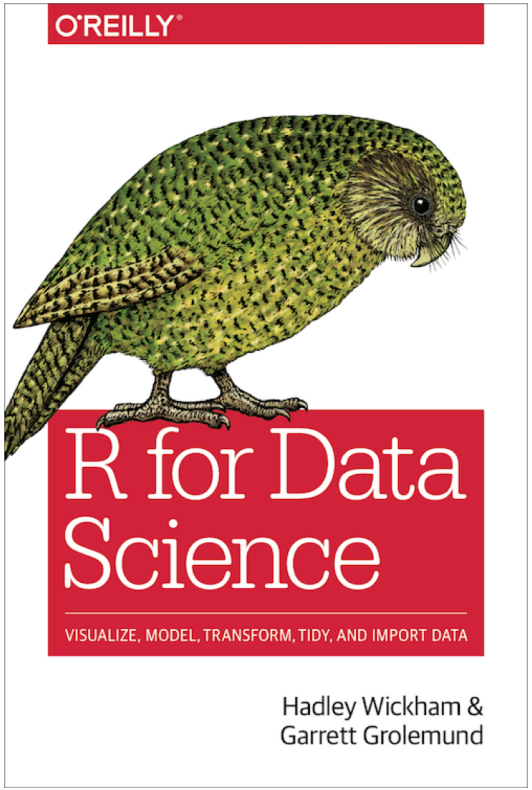
BUSMGT 7331: Descriptive Analytics and Visualization
Week 7

Visualizing Models, Dimensionality Reduction, and Clustering

Hyunwoo Park
Fisher College of Business
The Ohio State University

Textbook for this week

- R for Data Science
- Available free online: https://r4ds.had.co.nz/
- I will assign relevant chapters for reading.
- This book will be abbreviated as “R4DS” hereinafter.



Packages for this week

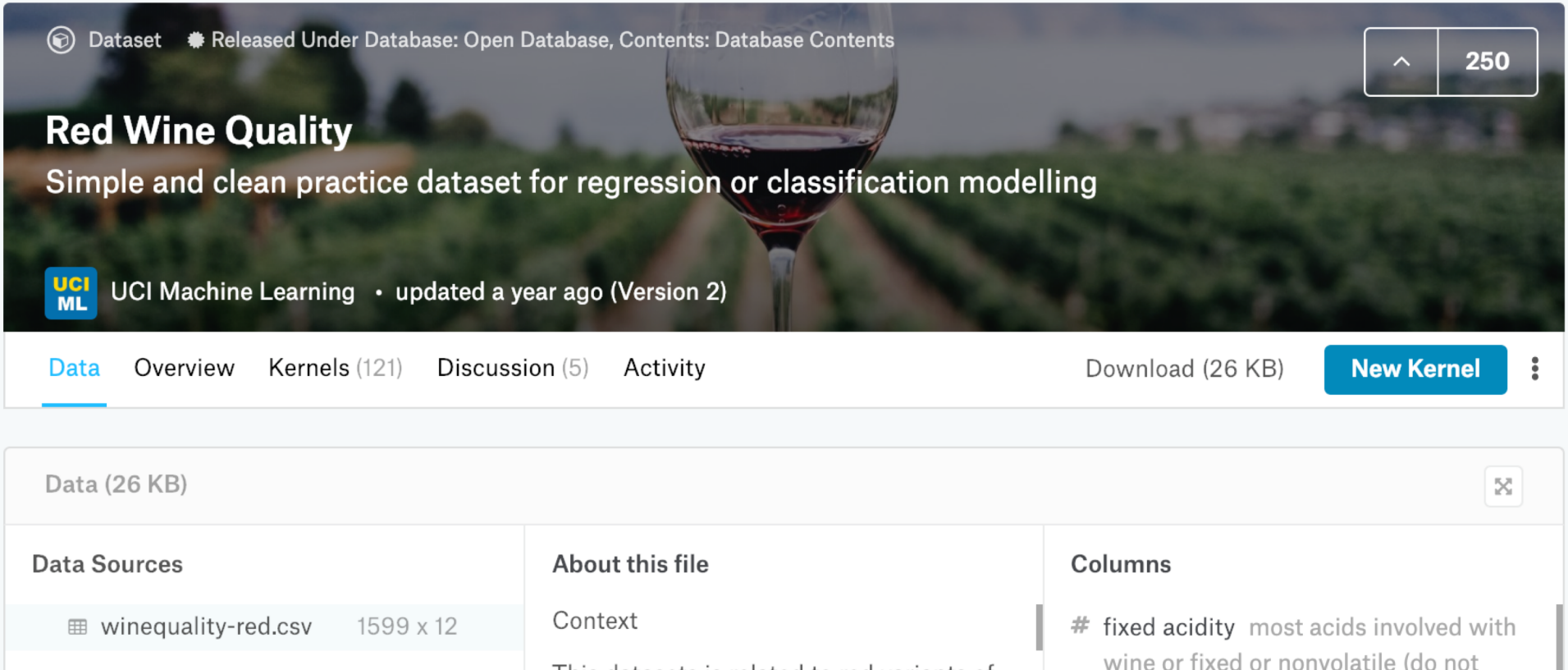
```
devtools::install_github("vqv/ggbiplot")
install.packages("ggdendro")

library(tidyverse)
library(modelr)
library(reshape2)
```

Linear Regression

Reading
- R4DS Chapter 18. Model Basics with modelr
- R4DS Chapter 19. Model Building

Dataset #1 for this week

- Red Wine Quality https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009



Loading and inspecting the data

```
1  wine <- read_csv("data/red-wine-quality-cortez-et-al-2009.zip")
2  wine
3  summary(wine)
```

```
> wine
# A tibble: 1,599 x 12
   `fixed acidity` `volatile acidi… `citric acid` `residual sugar` chlorides `free sulfur di…
             <dbl>            <dbl>         <dbl>            <dbl>     <dbl>            <dbl>
 1             7.4            0.7               0              1.9     0.076               11
 2             7.8            0.88              0              2.6     0.098               25
 3             7.8            0.76              0.04           2.3     0.092               15
 4            11.2            0.28              0.56           1.9     0.075               17
 5             7.4            0.7               0              1.9     0.076               11
 6             7.4            0.66              0              1.8     0.075               13
 7             7.9            0.6               0.06           1.6     0.069               15
 8             7.3            0.65              0              1.2     0.065               15
 9             7.8            0.580             0.02           2       0.073                9
10             7.5            0.5               0.36           6.1     0.071               17
# ... with 1,589 more rows, and 6 more variables: `total sulfur dioxide` <dbl>, density <dbl>,
#   pH <dbl>, sulphates <dbl>, alcohol <dbl>, quality <dbl>
```

```
> summary(wine)
 fixed acidity   volatile acidity  citric acid    residual sugar     chlorides       free sulfur dioxide
 Min.   : 4.60   Min.   :0.1200   Min.   :0.000   Min.   : 0.900   Min.   :0.01200   Min.   : 1.00
 1st Qu.: 7.10   1st Qu.:0.3900   1st Qu.:0.090   1st Qu.: 1.900   1st Qu.:0.07000   1st Qu.: 7.00
 Median : 7.90   Median :0.5200   Median :0.260   Median : 2.200   Median :0.07900   Median :14.00
 Mean   : 8.32   Mean   :0.5278   Mean   :0.271   Mean   : 2.539   Mean   :0.08747   Mean   :15.87
 3rd Qu.: 9.20   3rd Qu.:0.6400   3rd Qu.:0.420   3rd Qu.: 2.600   3rd Qu.:0.09000   3rd Qu.:21.00
 Max.   :15.90   Max.   :1.5800   Max.   :1.000   Max.   :15.500   Max.   :0.61100   Max.   :72.00
 total sulfur dioxide    density             pH          sulphates         alcohol         quality
 Min.   :  6.00       Min.   :0.9901   Min.   :2.740   Min.   :0.3300   Min.   : 8.40   Min.   :3.000
 1st Qu.: 22.00       1st Qu.:0.9956   1st Qu.:3.210   1st Qu.:0.5500   1st Qu.: 9.50   1st Qu.:5.000
 Median : 38.00       Median :0.9968   Median :3.310   Median :0.6200   Median :10.20   Median :6.000
 Mean   : 46.47       Mean   :0.9967   Mean   :3.311   Mean   :0.6581   Mean   :10.42   Mean   :5.636
 3rd Qu.: 62.00       3rd Qu.:0.9978   3rd Qu.:3.400   3rd Qu.:0.7300   3rd Qu.:11.10   3rd Qu.:6.000
 Max.   :289.00       Max.   :1.0037   Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :8.000
```

Simple linear regression

- Simple linear regression is a way to explain the variation of one variable with the variation of another variable.
- It's basically fitting a line between two variables.

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

$$\hat{y_d} = \hat{\alpha} + \hat{\beta} x_d$$

$$y_d = \hat{\alpha} + \hat{\beta} x_d + \varepsilon_d$$

- Simple linear regression estimates alpha and beta in the top equation above. Those estimated values are alpha-hat and beta-hat.

Running linear regression in R

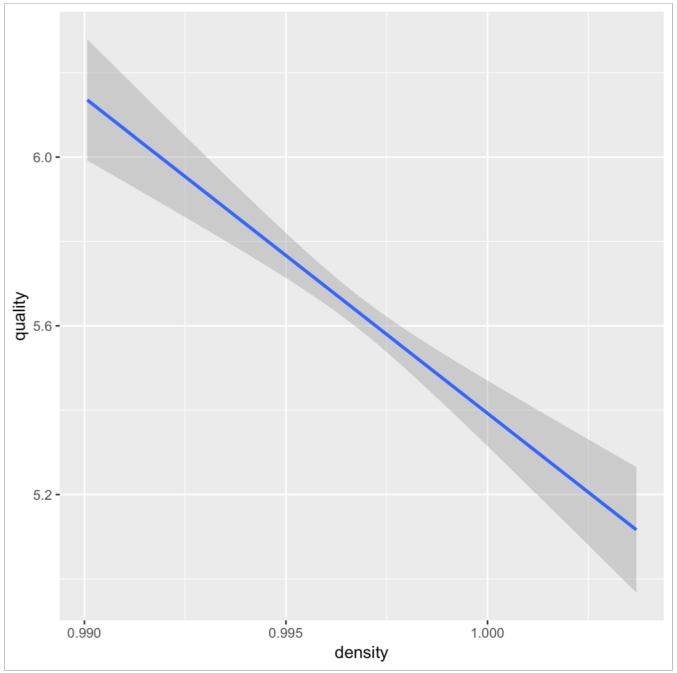
```
1  simple <- lm(quality ~ density, wine)
2  summary(simple)
3  simple$coefficients
4  simple$coefficients[1]
5  simple$coefficients[2]
```

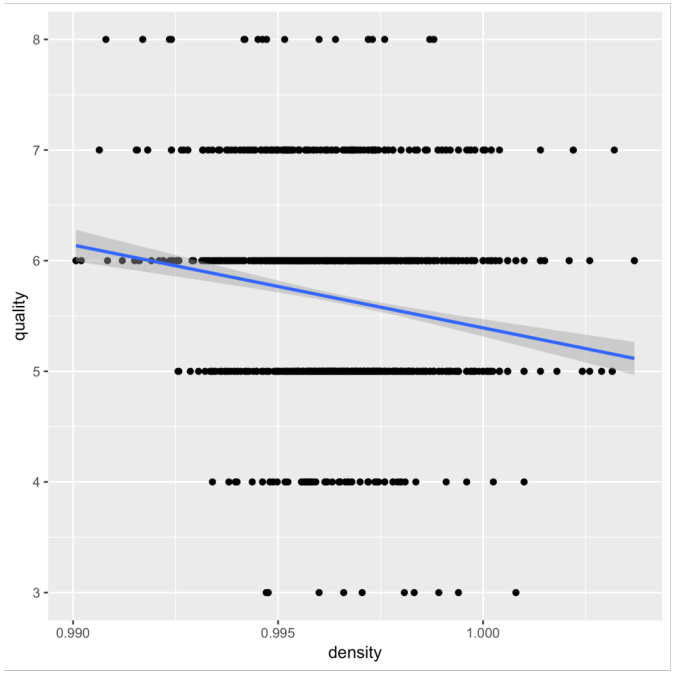
```
> simple <- lm(quality ~ density, wine)
> summary(simple)

Call:
lm(formula = quality ~ density, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7885 -0.6216  0.1554  0.4271  2.5177

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    80.24      10.51   7.636 3.83e-14 ***
density       -74.85      10.54  -7.100 1.87e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7954 on 1597 degrees of freedom
Multiple R-squared:  0.0306,    Adjusted R-squared:  0.02999
F-statistic: 50.41 on 1 and 1597 DF,  p-value: 1.875e-12
```

```
> simple$coefficients
(Intercept)     density
   80.23854   -74.84601
> simple$coefficients[1]
(Intercept)
   80.23854
> simple$coefficients[2]
  density
-74.84601
```

Visualizing a simple linear regression

```
1  ggplot(wine, aes(density, quality)) + geom_smooth(method='lm')
2  ggplot(wine, aes(density, quality)) + geom_point() + geom_smooth(method='lm')
```





Multiple linear regression

- You probably want to explain wine quality with more than just density of wine. Multiple linear regression is to linearly explain the outcome with multiple variables.

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \varepsilon_i$$

- Types of variables
  - Dependent variable (response variable, outcome variable)
  - Independent variable (explanatory variable, predictor, regressor)
  - Control variable

Interpreting a regression result

```
multiple <- lm(quality ~ `fixed acidity`+`volatile acidity`+`citric acid`+
  `residual sugar`+chlorides+`free sulfur dioxide`+`total sulfur dioxide`+
  density+pH+sulphates+alcohol, wine)
summary(multiple)
```

```
> summary(multiple)

Call:
lm(formula = quality ~ `fixed acidity` + `volatile acidity` +
    `citric acid` + `residual sugar` + chlorides + `free sulfur dioxide` +
    `total sulfur dioxide` + density + pH + sulphates + alcohol,
    data = wine)

Residuals:
     Min       1Q   Median       3Q      Max
-2.68911 -0.36652 -0.04699  0.45202  2.02498

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.197e+01  2.119e+01   1.036   0.3002
`fixed acidity`        2.499e-02  2.595e-02   0.963   0.3357
`volatile acidity`    -1.084e+00  1.211e-01  -8.948  < 2e-16 ***
`citric acid`         -1.826e-01  1.472e-01  -1.240   0.2150
`residual sugar`       1.633e-02  1.500e-02   1.089   0.2765
chlorides             -1.874e+00  4.193e-01  -4.470 8.37e-06 ***
`free sulfur dioxide`  4.361e-03  2.171e-03   2.009   0.0447 *
`total sulfur dioxide` -3.265e-03  7.287e-04  -4.480 8.00e-06 ***
density               -1.788e+01  2.163e+01  -0.827   0.4086
pH                    -4.137e-01  1.916e-01  -2.159   0.0310 *
sulphates              9.163e-01  1.143e-01   8.014 2.13e-15 ***
alcohol                2.762e-01  2.648e-02  10.429  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.648 on 1587 degrees of freedom
Multiple R-squared:  0.3606,    Adjusted R-squared:  0.3561
F-statistic: 81.35 on 11 and 1587 DF,  p-value: < 2.2e-16
```

Tabulating a multiple linear regression result

```
1  library(memisc)
2  lm1 <- lm(quality~density, wine)
3  lm2 <- lm(quality~density+alcohol, wine)
4  mtable("Model 1"=lm1, "Model 2"=lm2, summary.stats = c('R-
   squared','F','p','N'))
5  detach(package:memisc)
```

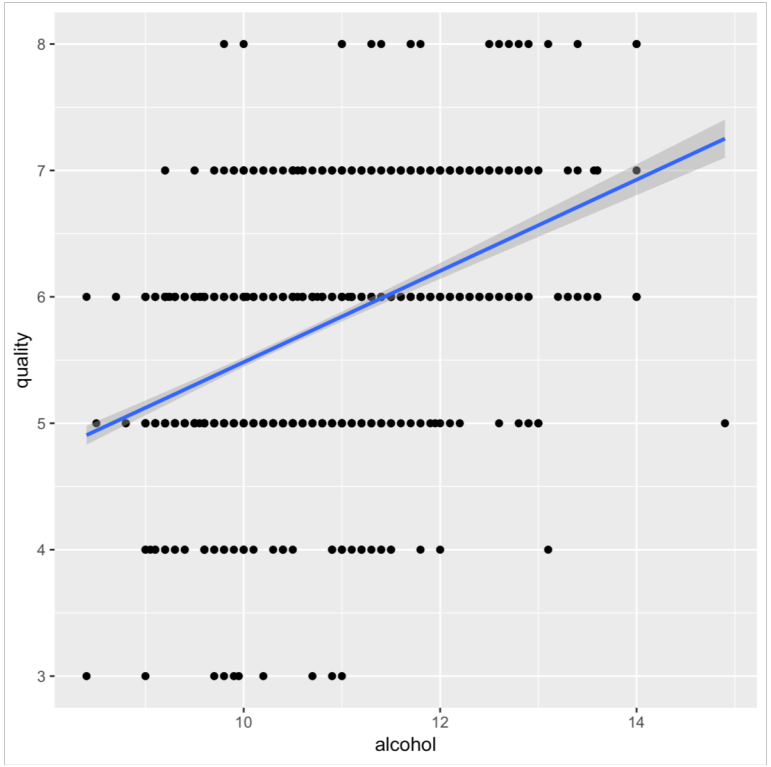
```
> mtable("Model 1"=lm1, "Model 2"=lm2, summary.stats = c('R-squared','F','p','N'))

Calls:
Model 1: lm(formula = quality ~ density, data = wine)
Model 2: lm(formula = quality ~ density + alcohol, data = wine)

========================================
                 Model 1      Model 2
----------------------------------------
  (Intercept)    80.239***   -33.152**
                (10.508)     (10.878)
  density       -74.846***    34.822**
                (10.542)     (10.813)
  alcohol                      0.391***
                              (0.019)
----------------------------------------
  R-squared       0.031        0.232
  F              50.405      240.693
  p               0.000        0.000
  N            1599         1599
========================================
```

Diagnostic plots for linear regression

- Some assumptions of linear regression
  - https://www.statisticssolutions.com/assumptions-of-multiple-linear-regression/
  - 1) Linear relationship: Dependent variable and independent variables are in a linear relationship.
  - 2) Normality of residuals: The residuals are distributed normally.
  - 3) No perfect multicollinearity: There should be no perfect correlations among regressors.
  - 4) Homoscedasticity: The variance of the residuals is roughly the same across all levels of predicted values.
- Diagnostic plots for checking these assumptions
  - 1) Scatterplot between x's and y
  - 2) Q-Q plot of the residuals
  - 3) Correlation matrix of all x's
  - 4) Scatterplot between residuals and y-hat (i.e., predicted value)

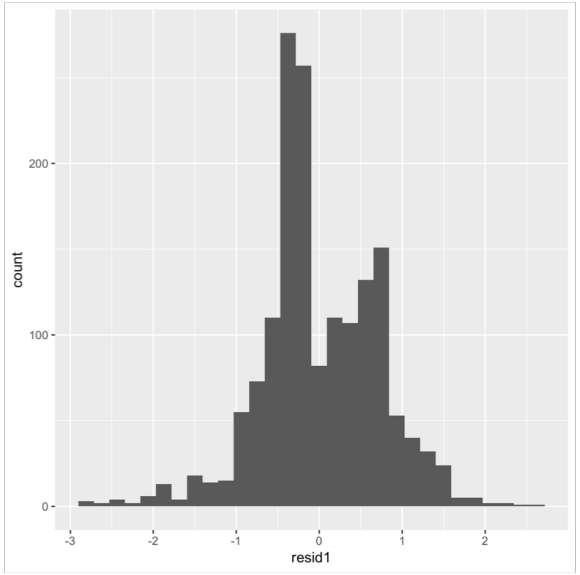
[1] Checking linearity of the relationship

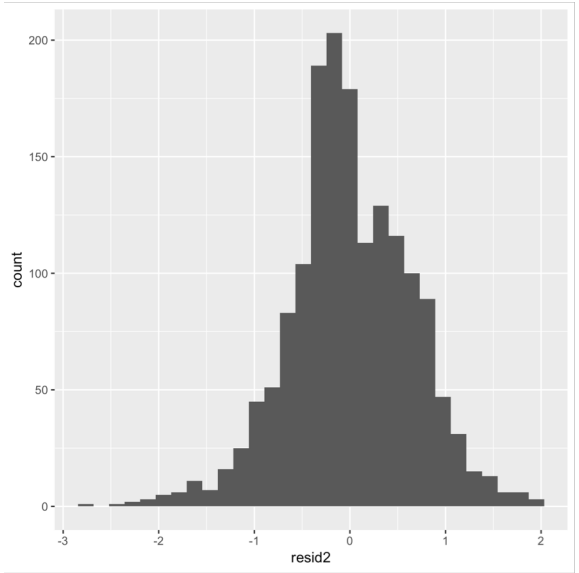
```
1  ggplot(wine, aes(alcohol, quality)) + geom_point() + geom_smooth(method='lm')
```



[2] Checking normality with histogram

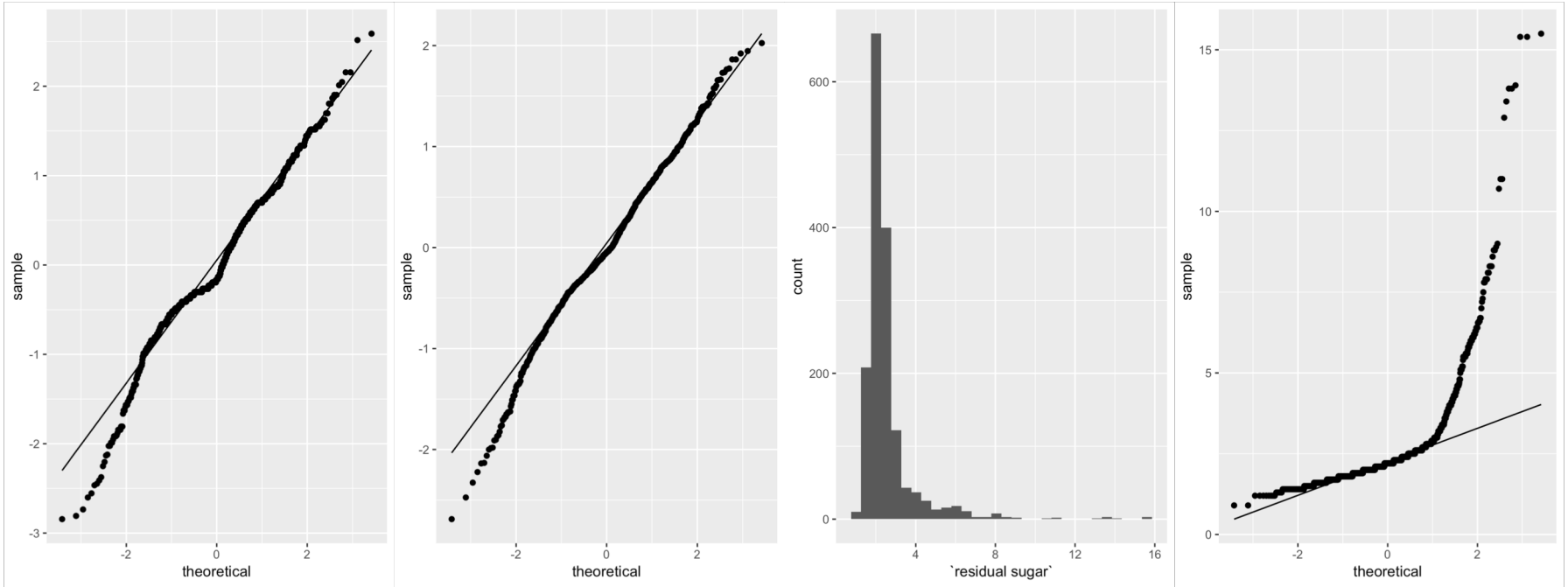
```
1  library(modelr)
2  qa <- lm(quality~alcohol, wine)
3  wineres <- wine %>% spread_residuals(qa, multiple) %>%
4    rename(resid1=qa, resid2=multiple)
5  ggplot(wineres, aes(resid1)) + geom_histogram()
6  ggplot(wineres, aes(resid2)) + geom_histogram()
```





[2] Checking normality with Q-Q plot

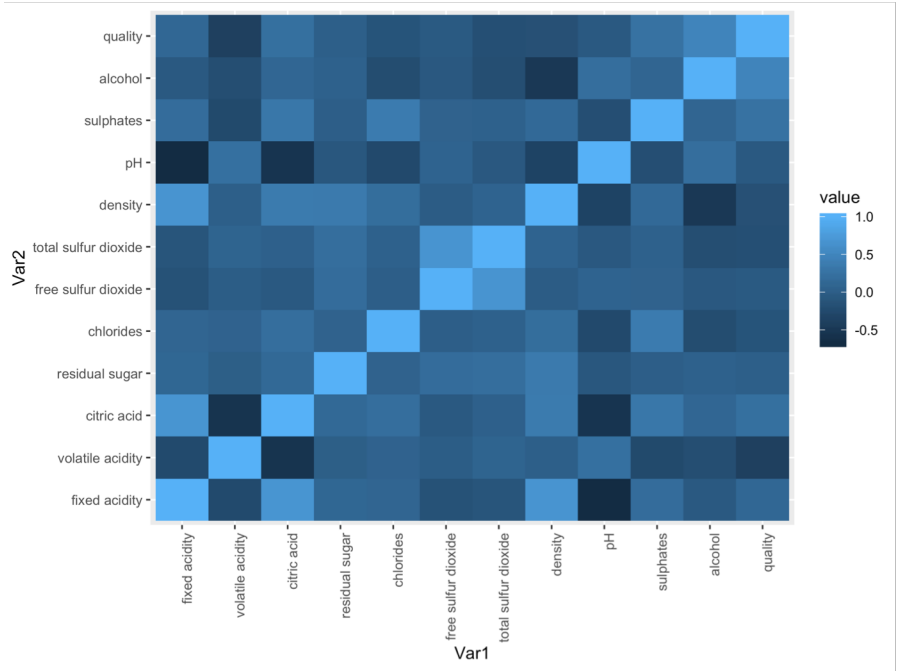
```
1  ggplot(wineres, aes(sample=resid1)) + stat_qq() + stat_qq_line()
2  ggplot(wineres, aes(sample=resid2)) + stat_qq() + stat_qq_line()
3  ggplot(wineres, aes(`residual sugar`)) + geom_histogram()
4  ggplot(wineres, aes(sample=`residual sugar`)) + stat_qq() + stat_qq_line()
```



[3] Visualizing correlation matrix

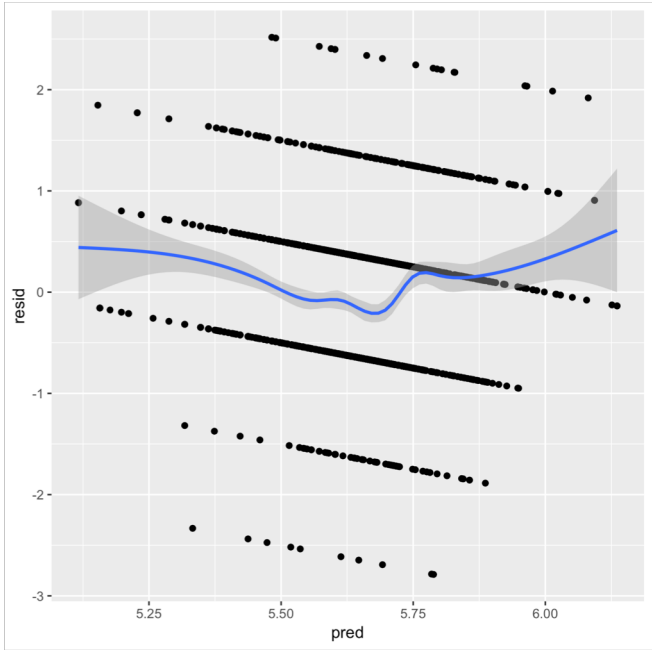
```
1  library(reshape2)
2  wine %>% cor() %>% melt() %>% ggplot() +
3    geom_tile(aes(Var1, Var2, fill=value)) +
4    theme(axis.text.x = element_text(angle=90, hjust=1))
```

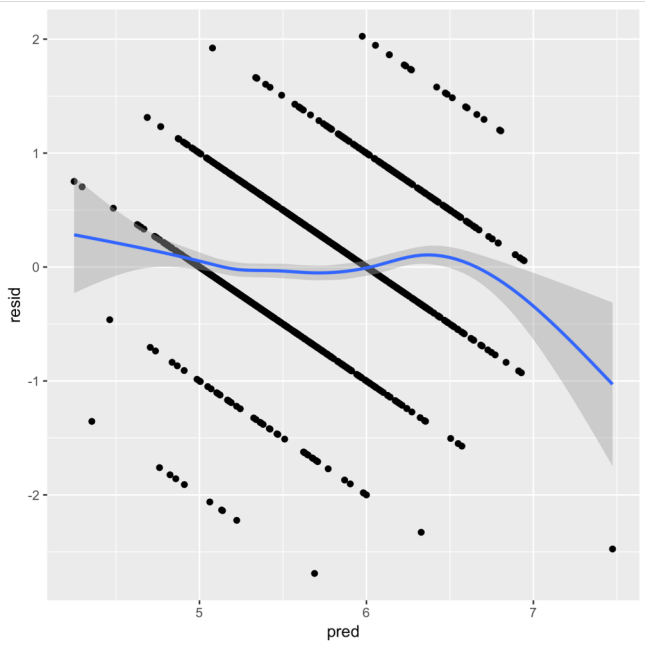
```
> wine %>% cor()
                      fixed acidity volatile acidity citric acid residual sugar    chlorides free sulfur dioxide
fixed acidity           1.00000000     -0.256130895  0.67170343    0.114776724  0.093705186        -0.153794193
volatile acidity       -0.25613089      1.000000000 -0.55249568    0.001917882  0.061297772        -0.010503827
citric acid             0.67170343     -0.552495685  1.00000000    0.143577162  0.203822914        -0.060978129
residual sugar          0.11477672      0.001917882  0.14357716    1.000000000  0.055609535         0.187048995
chlorides               0.09370519      0.061297772  0.20382291    0.055609535  1.000000000         0.005562147
free sulfur dioxide    -0.15379419     -0.010503827 -0.06097813    0.187048995  0.005562147         1.000000000
total sulfur dioxide   -0.11318144      0.076470005  0.03553302    0.203027882  0.047400468         0.667666450
density                 0.66804729      0.022026232  0.36494718    0.355283371  0.200632327        -0.021945831
pH                     -0.68297819      0.234937294 -0.54190414   -0.085652422 -0.265026131         0.070377499
sulphates               0.18300566     -0.260986685  0.31277004    0.005527121  0.371260481         0.051657572
alcohol                -0.06166827     -0.202288027  0.10990325    0.042075437 -0.221140545        -0.069408354
quality                 0.12405165     -0.390557780  0.22637251    0.013731637 -0.128906560        -0.050656057
                      total sulfur dioxide     density          pH     sulphates      alcohol      quality
fixed acidity              -0.11318144  0.66804729 -0.68297819  0.183005664 -0.06166827  0.12405165
volatile acidity            0.07647000  0.02202623  0.23493729 -0.260986685 -0.20228803 -0.39055778
citric acid                 0.03553302  0.36494718 -0.54190414  0.312770044  0.10990325  0.22637251
residual sugar              0.20302788  0.35528337 -0.08565242  0.005527121  0.04207544  0.01373164
chlorides                   0.04740047  0.20063233 -0.26502613  0.371260481 -0.22114054 -0.12890656
free sulfur dioxide         0.66766645 -0.02194583  0.07037750  0.051657572 -0.06940835 -0.05065606
total sulfur dioxide        1.00000000  0.07126948 -0.06649456  0.042946836 -0.20565394 -0.18510029
density                     0.07126948  1.00000000 -0.34169933  0.148506412 -0.49617977 -0.17491923
pH                         -0.06649456 -0.34169933  1.00000000 -0.196647602  0.20563251 -0.05773139
sulphates                   0.04294684  0.14850641 -0.19664760  1.000000000  0.09359475  0.25139708
alcohol                    -0.20565394 -0.49617977  0.20563251  0.093594750  1.00000000  0.47616632
quality                    -0.18510029 -0.17491923 -0.05773139  0.251397079  0.47616632  1.00000000
```



[4] Variance of the residuals along the predicted values

```
1  wine %>% add_residuals(simple) %>% add_predictions(simple) %>%
2    ggplot(aes(pred, resid)) + geom_point() + geom_smooth()
3  wine %>% add_residuals(multiple) %>% add_predictions(multiple) %>%
     ggplot(aes(pred, resid)) + geom_point() + geom_smooth()
```





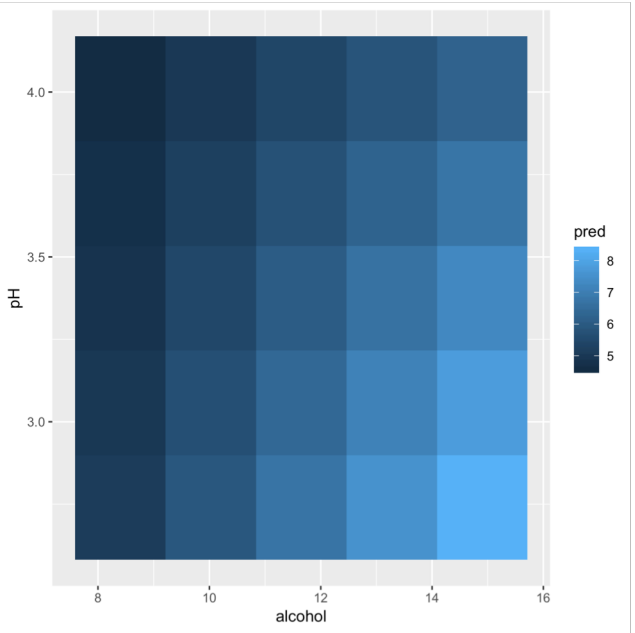
Understanding interaction effects

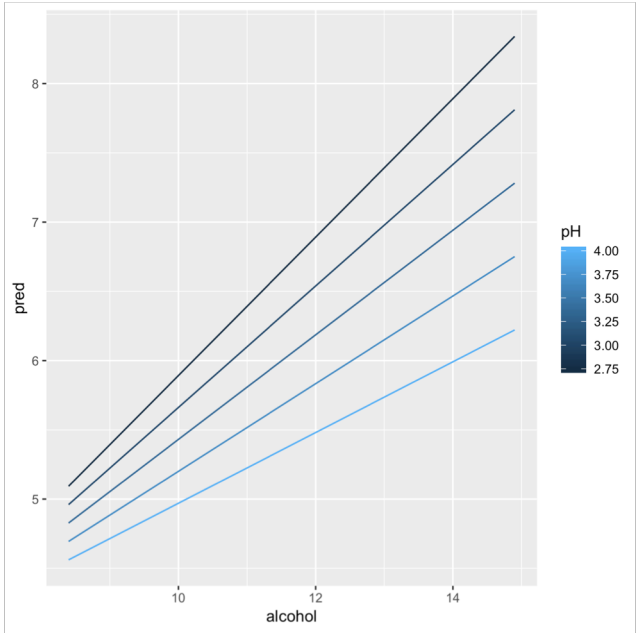
```
1  inter <- lm(quality~alcohol+pH+alcohol*pH, wine)
2  summary(inter)
```

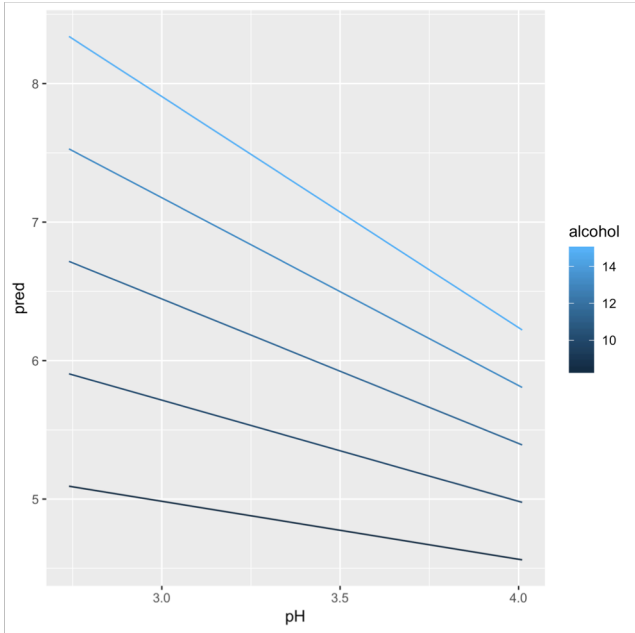
```
> inter <- lm(quality~alcohol+pH+alcohol*pH, wine)
> summary(inter)

Call:
lm(formula = quality ~ alcohol + pH + alcohol * pH, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9403 -0.3946 -0.1319  0.5049  2.4634

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.38434    3.17040  -0.752 0.452124
alcohol      1.02668    0.29643   3.463 0.000547 ***
pH           1.19713    0.95298   1.256 0.209229
alcohol:pH  -0.19234    0.08887  -2.164 0.030595 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6981 on 1595 degrees of freedom
Multiple R-squared:  0.2542,    Adjusted R-squared:  0.2528
F-statistic: 181.2 on 3 and 1595 DF,  p-value: < 2.2e-16
```

Visualizing interaction effects

```
1  grid <- wine %>% data_grid(alcohol=seq_range(alcohol, 5), pH=seq_range(pH,
   5)) %>% gather_predictions(inter)
2  ggplot(grid, aes(alcohol, pH)) + geom_tile(aes(fill=pred))
3  ggplot(grid, aes(alcohol, pred, color=pH, group=pH)) + geom_line()
4  ggplot(grid, aes(pH, pred, color=alcohol, group=alcohol)) + geom_line()
```



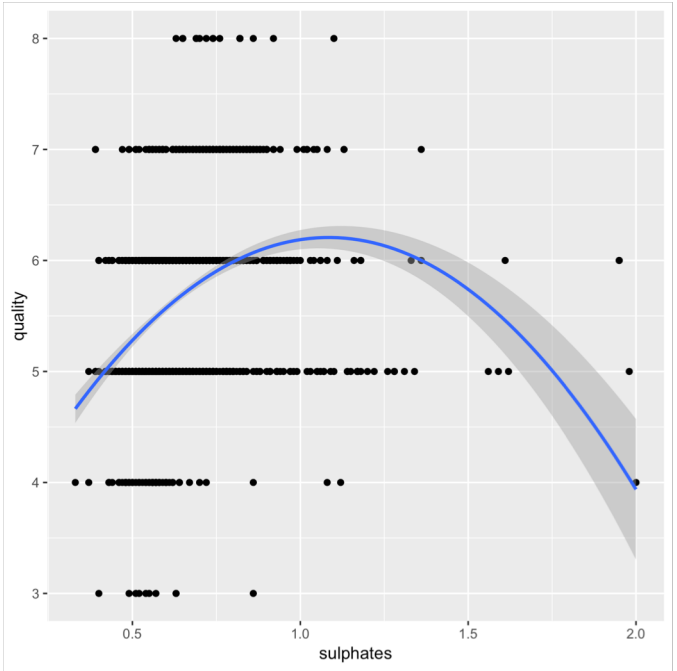




Effect of a quadratic term

```
1  quad <- lm(quality~poly(sulphates,2), wine)
2  summary(quad)
3  ggplot(wine, aes(sulphates, quality)) + geom_point() +
4    geom_smooth(method="lm", formula=y~poly(x,2))
```

```
> summary(quad)

Call:
lm(formula = quality ~ poly(sulphates, 2), data = wine)

Residuals:
     Min       1Q   Median       3Q      Max
-3.07005 -0.51662 -0.03107  0.46984  2.35340

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           5.63602    0.01879  299.97   <2e-16 ***
poly(sulphates, 2)1   8.11575    0.75131   10.80   <2e-16 ***
poly(sulphates, 2)2 -8.68419    0.75131  -11.56   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7513 on 1596 degrees of freedom
Multiple R-squared:  0.1356,    Adjusted R-squared:  0.1345
F-statistic: 125.1 on 2 and 1596 DF,  p-value: < 2.2e-16
```



Regression as a descriptive tool

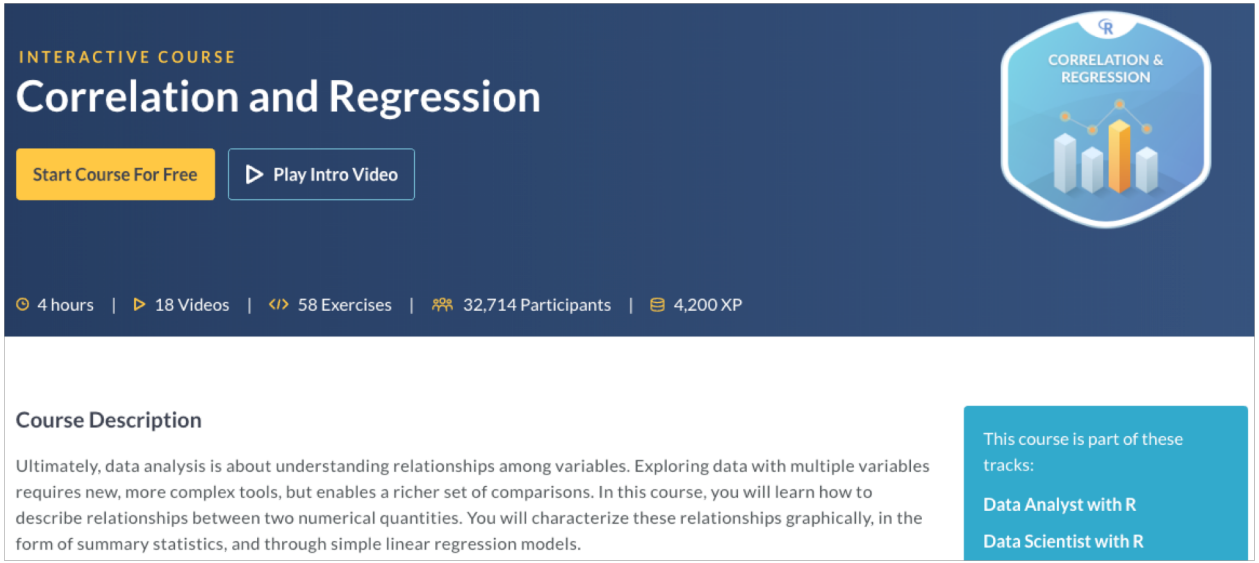
- The purpose of a linear regression can be both descriptive and predictive.
- When used as a descriptive tool, the purpose is to find which variables have significant explanatory power in the variation of the dependent variable.
  In this case, one must be careful in checking assumptions and validity of the model.
- When used as a predictive tool, the purpose is to make a prediction as accurate as possible. The correctness of the prediction is an important performance measure.
  But, there are many better predictive models than linear regression.
- Even if the procedure of fitting the model is the same for both purposes, what you want to show about and from the model differs depending on your purpose.

Correlation vs. causation

- One critical assumption for linear regression is the exogeneity assumption.
- Explanatory variables must not be correlated with the unexplained variation of the dependent variable.
- Suppose you are explaining employee performance with whether one participated in a training program. The unexplained variation of performance can probably partly explained by the variation of intelligence. If intelligence is also correlated with the participation in a training program, then the estimated effect of training program becomes invalid.
- It is often hard to make a causal argument from regression analysis on observational data alone. Keeping in mind regression analysis is also describing the correlation between the dependent variable and the independent variable after controlling for some confounding effects.

DataCamp course

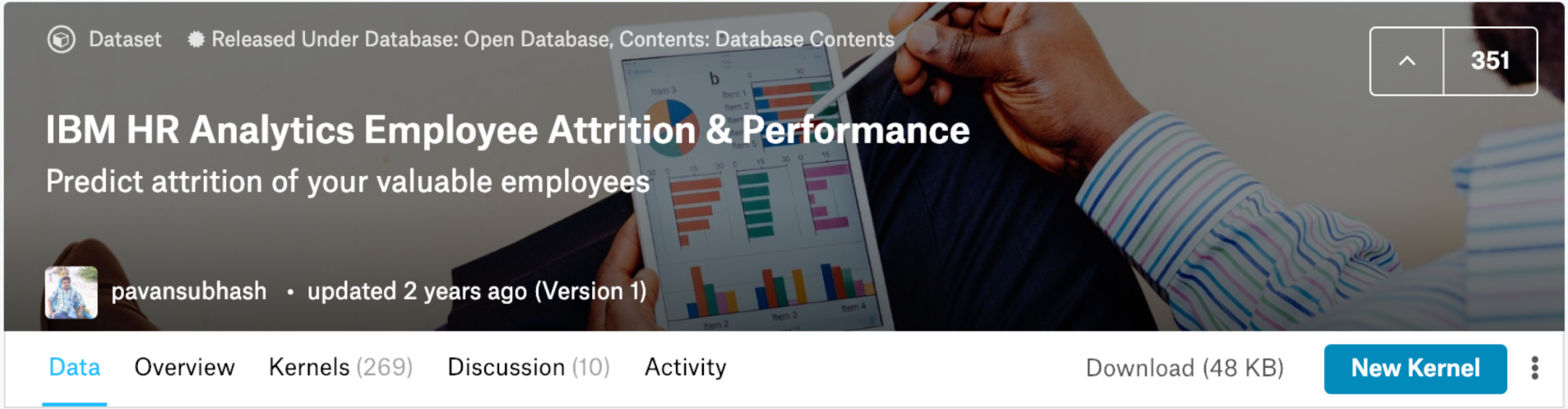
- Complete the following Data Camp courses on regression.
  - Correlation and Regression



Logistic Regression

Dataset #2 for this week

- IBM HR Analytics Employee Attrition & Performance
  https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset



Fitting a linear model

```
1  ibm <- read_csv("data/ibm-hr-analytics-employee-attrition-performance.zip")
   %>% mutate(attr=ifelse(Attrition=="Yes",1,0),
   female=ifelse(Gender=="Female",1,0))
2  m1 <- lm(attr~female+PerformanceRating+
3    log(MonthlyIncome)+DistanceFromHome, ibm)
4  summary(m1)
5  c(min(m1$fitted.values), max(m1$fitted.values))
```

```
> summary(m1)

Call:
lm(formula = attr ~ female + PerformanceRating + log(MonthlyIncome) +
    DistanceFromHome, data = ibm)

Residuals:
     Min       1Q   Median       3Q      Max
-0.34624 -0.20114 -0.13834 -0.04522  1.02051

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         1.083493   0.148256   7.308 4.43e-13 ***
female             -0.015621   0.019176  -0.815  0.41545
PerformanceRating  -0.003902   0.026037  -0.150  0.88088
log(MonthlyIncome) -0.109529   0.014147  -7.742 1.81e-14 ***
DistanceFromHome    0.003593   0.001158   3.102  0.00196 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3598 on 1465 degrees of freedom
Multiple R-squared:  0.04599,   Adjusted R-squared:  0.04338
F-statistic: 17.65 on 4 and 1465 DF,  p-value: 3.665e-14
```

```
> c(min(m1$fitted.values), max(m1$fitted.values))
[1] -0.02720952  0.39189067
```

Logistic regression

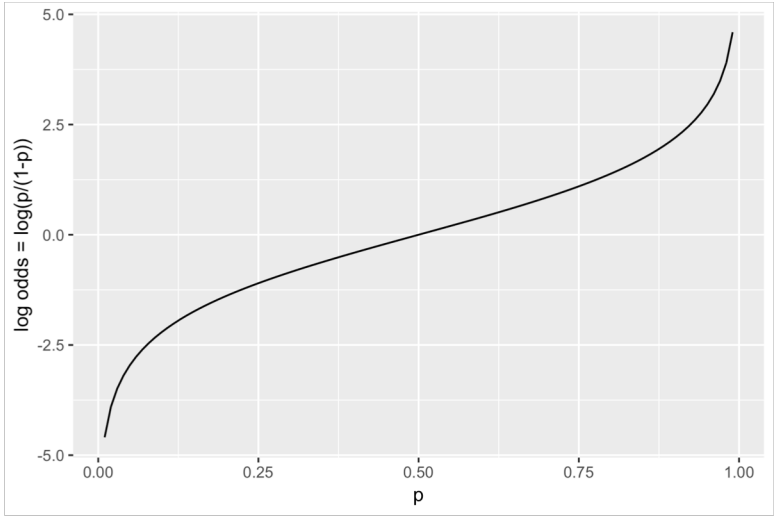
- In the previous slide, we fitted a linear probability model of attrition. The problem is that the predicted probability can be not within a [0, 1] interval.
- Logistic regression is often used when you have a binary (0 or 1) outcome variable.
- The equation to be estimated is:

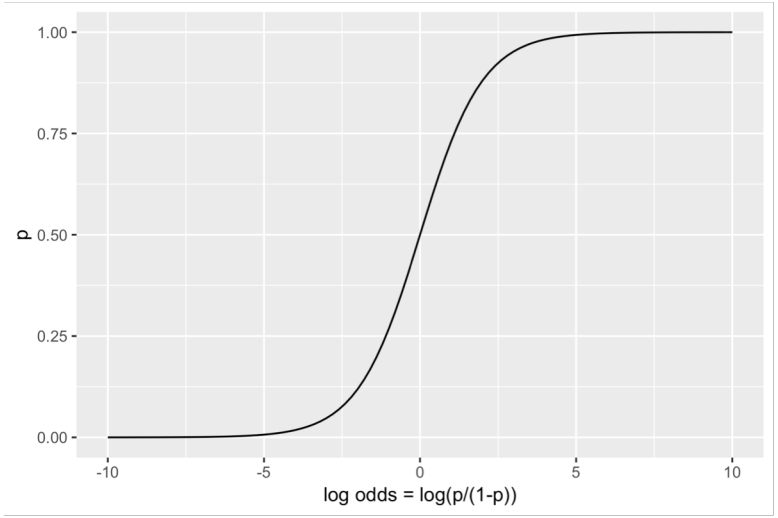
$$\begin{aligned}\text{logit}(p) &= \log(\text{odds}) = \log\left(\frac{p}{1-p}\right) \\ &= \alpha + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + \varepsilon_i\end{aligned}$$

- It's somewhat similar to linear regression except it involves this logit function.
- This family of models is called generalized linear model.

Odds ratio and logit function

```
ggplot(data.frame(x=c(0.01,0.99)), aes(x)) +
  stat_function(fun=function(p){log(p/(1-p))}) +
  labs(x="p", y="log odds = log(p/(1-p))")
ggplot(data.frame(x=c(-10,10)), aes(x)) +
  stat_function(fun=function(l){exp(l)/(1+exp(l))}) +
  labs(x="log odds = log(p/(1-p))", y="p")
```





Logistic regression in R

```
1   m2 <- glm(attr~female+PerformanceRating+
      log(MonthlyIncome)+DistanceFromHome, ibm, family="binomial")
2   summary(m2)
3   c(min(m2$fitted.values), max(m2$fitted.values))
```

```
> m2 <- glm(attr~female+PerformanceRating+
+   log(MonthlyIncome)+DistanceFromHome, ibm, family="binomial")
> summary(m2)

Call:
glm(formula = attr ~ female + PerformanceRating + log(MonthlyIncome) +
    DistanceFromHome, family = "binomial", data = ibm)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0719  -0.6500  -0.5110  -0.3508   2.5752

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         6.014266   1.226784   4.902 9.46e-07 ***
female             -0.109073   0.150399  -0.725  0.46832
PerformanceRating  -0.043717   0.201171  -0.217  0.82796
log(MonthlyIncome) -0.920406   0.123943  -7.426 1.12e-13 ***
DistanceFromHome    0.027564   0.008602   3.204  0.00135 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1298.6  on 1469  degrees of freedom
Residual deviance: 1226.5  on 1465  degrees of freedom
AIC: 1236.5

Number of Fisher Scoring iterations: 5
```

```
> c(min(m2$fitted.values), max(m2$fitted.values))
[1] 0.03410932 0.52661408
```

Interpreting a logistic regression result

```
1  exp(coef(m2)[2])
2  exp(coef(m2)[3])
3  exp(coef(m2)[4])
```

```
Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)         6.014266   1.226784   4.902 9.46e-07 ***
female             -0.109073   0.150399  -0.725  0.46832
PerformanceRating  -0.043717   0.201171  -0.217  0.82796
log(MonthlyIncome) -0.920406   0.123943  -7.426 1.12e-13 ***
DistanceFromHome    0.027564   0.008602   3.204  0.00135 **
```

```
> exp(coef(m2)[2])
   female
0.8966652
> exp(coef(m2)[3])
PerformanceRating
        0.9572249
> exp(coef(m2)[4])
log(MonthlyIncome)
         0.3983571
```

Other models for other types of dependent variables

- Ordinal logistic/probit regression
- Fractional logistic/probit regression
- Poisson regression
- Negative binomial regression

Dimensionality Reduction

How to visually summarize high-dimensional data?

- High dimensional data usually refers to the dataset where the number of variables (p) is greater than or close to the number of observations (n).
- Even when a situation is not technically high dimensional, even a modest number of variables pose a challenge for visualization because we usually only have a medium with two dimension (x and y).
- There are many techniques for dimensionality reduction.
  - The classical one: PCA (principal component analysis)
  - Recent one: t-SNE (t-distributed stochastic neighbor embedding)
- We will see how dimensionality reduction helps summarize many variables and describe the data through visualization.

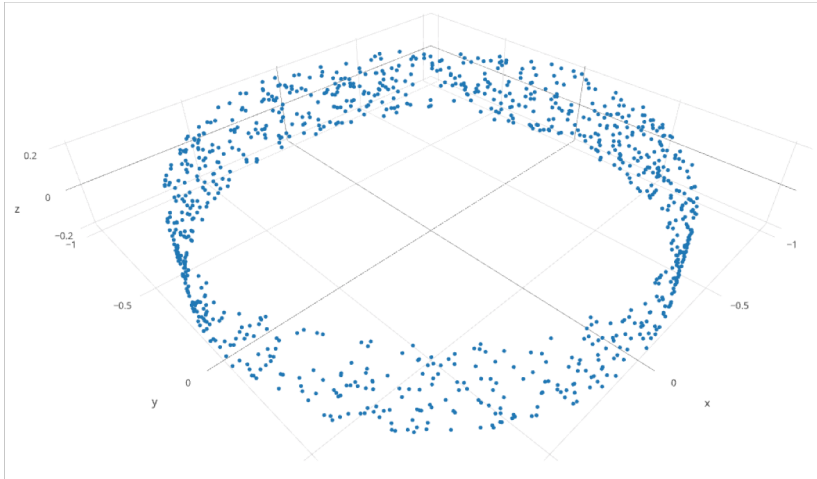
Principal Component Analysis (PCA)

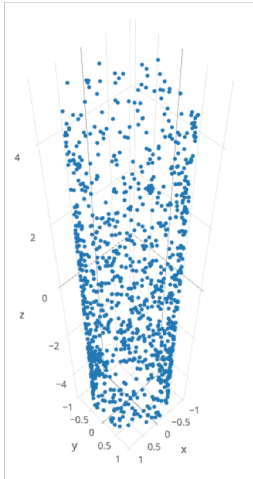
- PCA basically aims at finding a new axis as a linear combination of existing variables that best explains the variation of the data.
- If you give p variables to PCA, it will give you back a new set of p variables.
- Those new variables are called (first, second, third, …) principal components.
- The order of principal components is based on how well it explains the variation.
- Each principal component is designed to be uncorrelated with one another.

- A good tutorial on running and visualizing PCA: https://www.datacamp.com/community/tutorials/pca-analysis-r

Creating toy datasets

```
1  N <- 1000
2  flat <- tibble(theta=runif(N,0,2*pi), x=cos(theta), y=sin(theta), z=runif(N,-
   .2,.2)) %>% select(-theta)
3  plot_ly(x=flat$x, y=flat$y, z=flat$z, type="scatter3d", mode="markers",
   marker=list(size=3)) %>% layout(scene=list(aspectmode="data"))
4  long <- tibble(theta=runif(N,0,2*pi), x=cos(theta), y=sin(theta), z=runif(N,-
   5,5)) %>% select(-theta)
5  plot_ly(x=long$x, y=long$y, z=long$z, type="scatter3d", mode="markers",
   marker=list(size=3)) %>% layout(scene=list(aspectmode="data"))
```





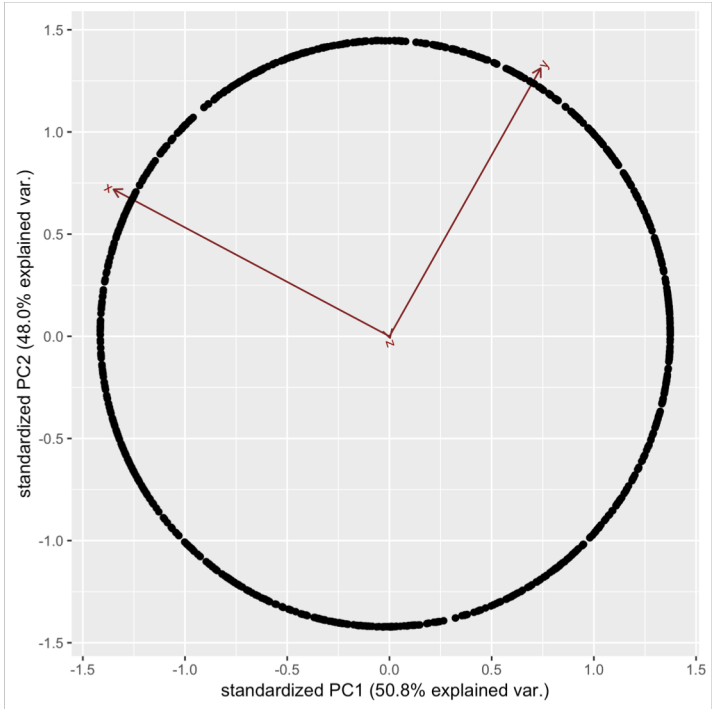
Running PCA on the toy example

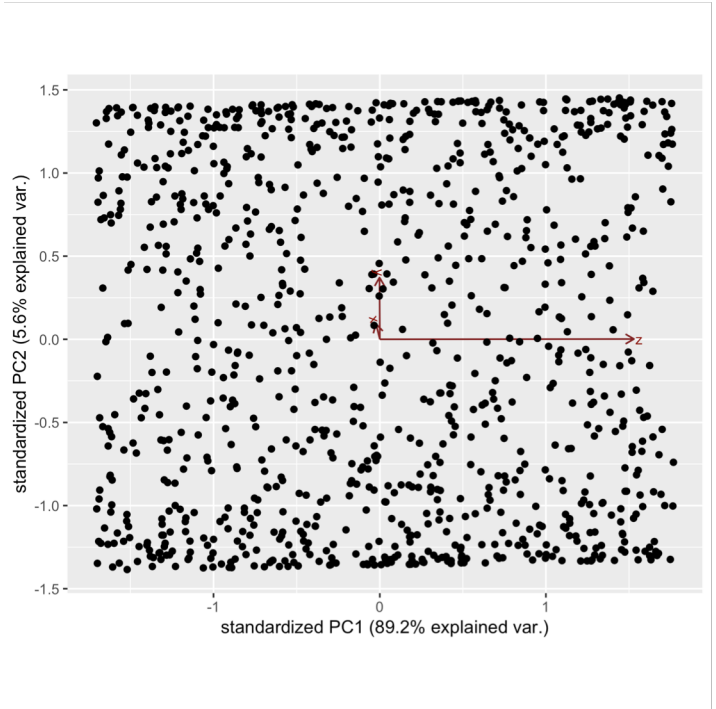
```
1  flat.pca <- prcomp(flat)
2  long.pca <- prcomp(long)
3  summary(flat.pca)
4  summary(long.pca)
```

```
> flat.pca <- prcomp(flat)
> long.pca <- prcomp(long)
> summary(flat.pca)
Importance of components:
                          PC1    PC2     PC3
Standard deviation     0.7173 0.6972 0.11293
Proportion of Variance 0.5077 0.4797 0.01258
Cumulative Proportion  0.5077 0.9874 1.00000
> summary(long.pca)
Importance of components:
                          PC1    PC2    PC3
Standard deviation     2.8723 0.7190 0.6935
Proportion of Variance 0.8921 0.0559 0.0520
Cumulative Proportion  0.8921 0.9480 1.0000
```

Visualizing data with first two principal components

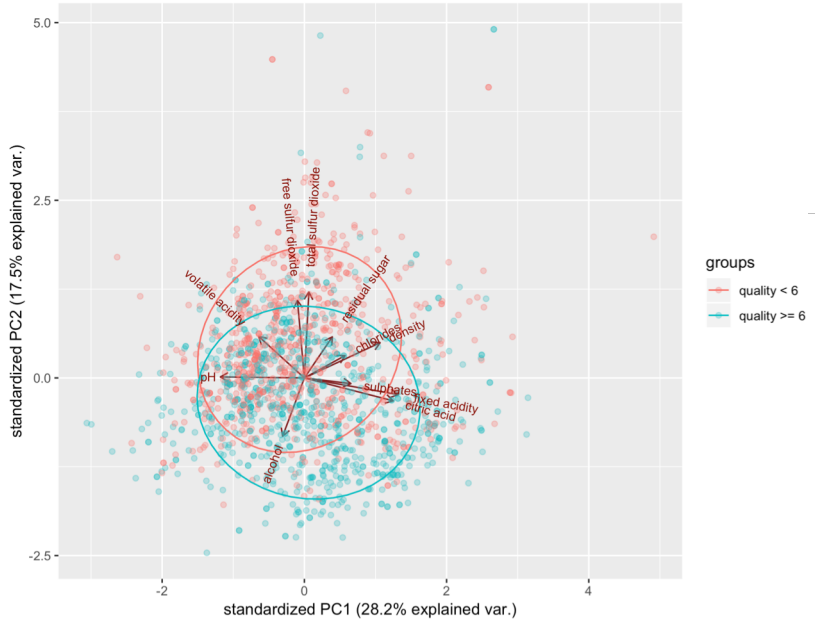
```
1  library(ggbiplot)
2  ggbiplot(flat.pca)
3  ggbiplot(long.pca)
```





Visualizing the wine data

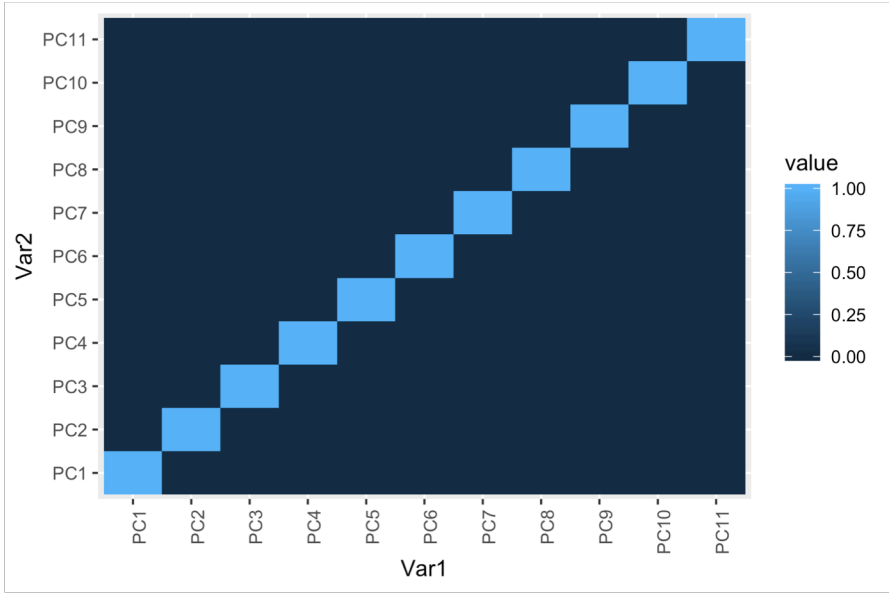
```
1  wine2 <- wine %>% mutate(highq = ifelse(quality>=6, "quality >= 6", "quality
   < 6"))
2  wine2.pca <- prcomp(wine2 %>% select(-highq,-quality), scale=T)
3  summary(wine2.pca)
4  ggbiplot(wine2.pca, alpha=.3, groups=wine2$highq, ellipse = TRUE)
```




```
> summary(wine2.pca)
Importance of components:
                          PC1    PC2    PC3    PC4
Standard deviation     1.7604 1.3878 1.2452 1.1015
Proportion of Variance 0.2817 0.1751 0.1410 0.1103
Cumulative Proportion  0.2817 0.4568 0.5978 0.7081
```

Are principal components uncorrelated?

```
1  as.tibble(cor(wine2.pca$x))
2  wine2.pca$x %>% cor() %>% melt() %>% ggplot() +
3    geom_tile(aes(Var1, Var2, fill=value)) +
4    theme(axis.text.x = element_text(angle=90, hjust=1))
```




```
> as.tibble(cor(wine2.pca$x))
# A tibble: 11 x 11
        PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8       PC9      PC10      PC11
      <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>     <dbl>
 1  1.00e+ 0 -2.20e-16  4.95e-16 -5.77e-16  2.94e-18  1.23e-16 -1.31e-15  5.72e-17 -1.01e-15 -9.36e-16 -3.64e-15
 2 -2.20e-16  1.00e+ 0  4.86e-16 -8.58e-16  2.19e-16  8.67e-16  6.54e-17  1.96e-16  2.74e-16 -1.27e-16  1.17e-15
 3  4.95e-16  4.86e-16  1.00e+ 0  6.56e-16  1.99e-17  7.46e-16 -4.74e-16  6.65e-16 -3.20e-16  6.68e-16 -1.05e-15
 4 -5.77e-16 -8.58e-16  6.56e-16  1.00e+ 0  9.77e-16  1.83e-16  7.34e-16 -5.63e-16  1.74e-17 -1.10e-15 -9.70e-16
 5  2.94e-18  2.19e-16  1.99e-17  9.77e-16  1.00e+ 0  6.11e-17  7.98e-16 -3.83e-16  2.25e-16  5.60e-16 -2.32e-15
 6  1.23e-16  8.67e-16  7.46e-16  1.83e-16  6.11e-17  1.00e+ 0  5.53e-16 -1.19e-15  5.99e-16  6.64e-16  8.29e-16
 7 -1.31e-15  6.54e-17 -4.74e-16  7.34e-16  7.98e-16  5.53e-16  1.00e+ 0 -4.06e-16 -3.32e-16  9.78e-16 -2.00e-15
 8  5.72e-17  1.96e-16  6.65e-16 -5.63e-16 -3.83e-16 -1.19e-15 -4.06e-16  1.00e+ 0  1.00e-15  2.43e-15  5.55e-16
 9 -1.01e-15  2.74e-16 -3.20e-16  1.74e-17  2.25e-16  5.99e-16 -3.32e-16  1.00e-15  1.00e+ 0 -2.70e-16 -1.52e-15
10 -9.36e-16 -1.27e-16  6.68e-16 -1.10e-15  5.60e-16  6.64e-16  9.78e-16  2.43e-15 -2.70e-16  1.00e+ 0 -9.96e-16
11 -3.64e-15  1.17e-15 -1.05e-15 -9.70e-16 -2.32e-15  8.29e-16 -2.00e-15  5.55e-16 -1.52e-15 -9.96e-16  1.00e+ 0
```

t-SNE

- https://lvdmaaten.github.io/tsne/

Laurens van der Maaten

Research scientist in machine learning and computer vision.

Email
Facebook
Google+
Github

t-SNE

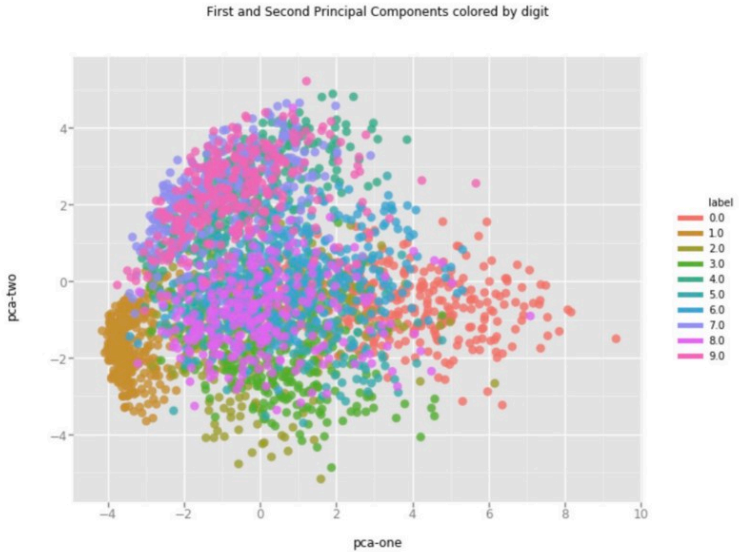
t-Distributed Stochastic Neighbor Embedding (t-SNE) is a (prize-winning) technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets. The technique can be implemented via Barnes-Hut approximations, allowing it to be applied on large real-world datasets. We applied it on data sets with up to 30 million examples. The technique and its variants are introduced in the following papers:

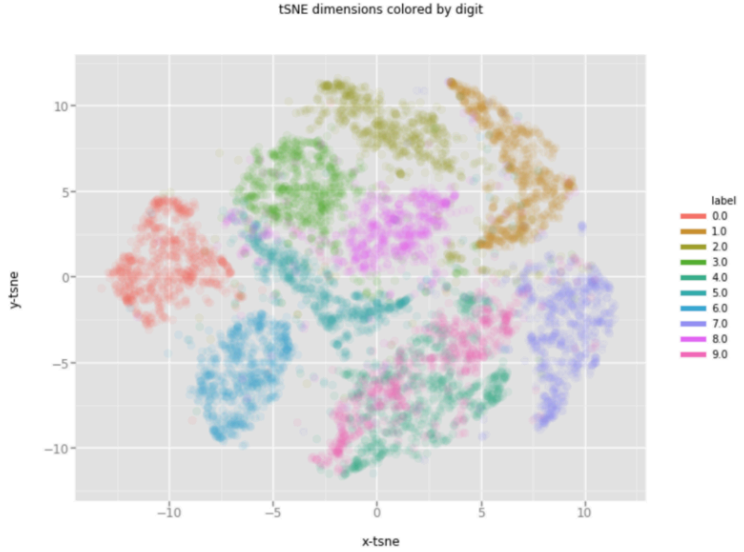
- L.J.P. van der Maaten. **Accelerating t-SNE using Tree-Based Algorithms**. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014. PDF [Supplemental material]
- L.J.P. van der Maaten and G.E. Hinton. **Visualizing Non-Metric Similarities in Multiple Maps**. *Machine Learning* 87(1):33-55, 2012. PDF
- L.J.P. van der Maaten. **Learning a Parametric Embedding by Preserving Local Structure**. In *Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP* 5:384-391, 2009. PDF
- L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008. PDF [Supplemental material] [Talk]

An accessible introduction to t-SNE and its variants is given in this Google Techtalk.

Comparing PCA with t-SNE

- A few articles on the web
  - https://www.r-bloggers.com/playing-with-dimensions-from-clustering-pca-t-sne-to-carl-sagan/
  - https://medium.com/@luckylwk/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b





Clustering

What is clustering and why?

- It’s like topic modeling in text mining.
- Inserting additional layer of abstraction to facilitate classification, interpretation, and discussion on potentially a large set of data points.
- It’s like customer segmentation in marketing.
- Along with dimensionality reduction, clustering is also a unsupervised learning task.

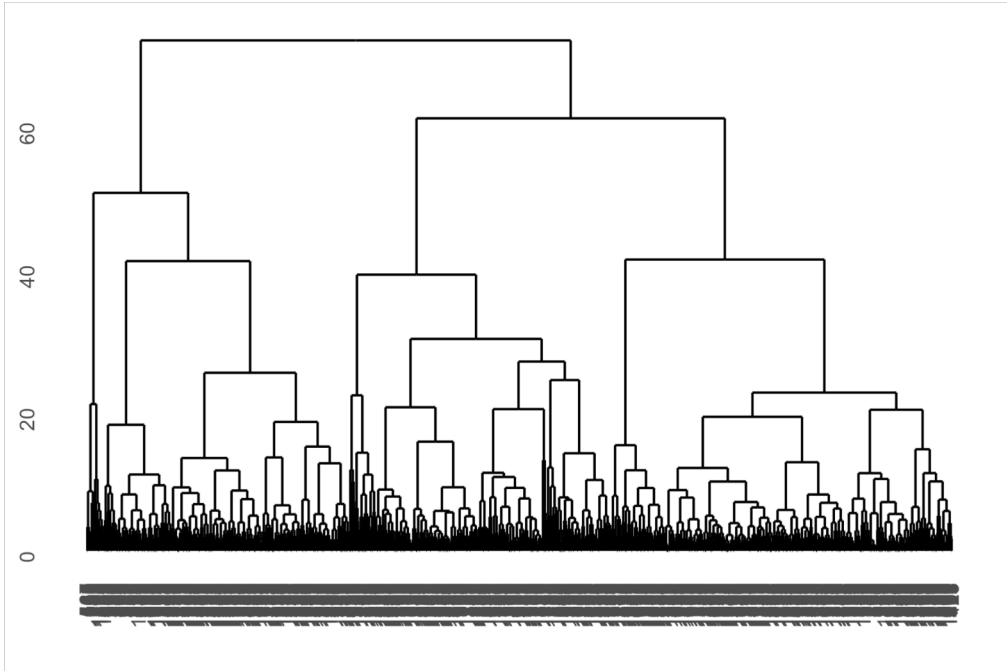
- The goal is:
  - Collect all points similar to one another into the same group.
  - Maximize the differences across groups.

Hierarchical clustering

- Hierarchical clustering groups observations in a hierarchical manner.
- The process can be both bottom-up or top-down.
  - The bottom-up approach is to start with all observations as separate clusters and keep merging up based on the distances among them until you end up with a single cluster.
  - The top-down approach is to start with all observations belonging to a single cluster and keep dividing down based on the distances among them until you end up with all clusters having a single observation.
- The result is a tree called dendrogram that records when observations are grouped together or divided into subgroups.
- The number of clusters you obtain depends on where to cut the tree.

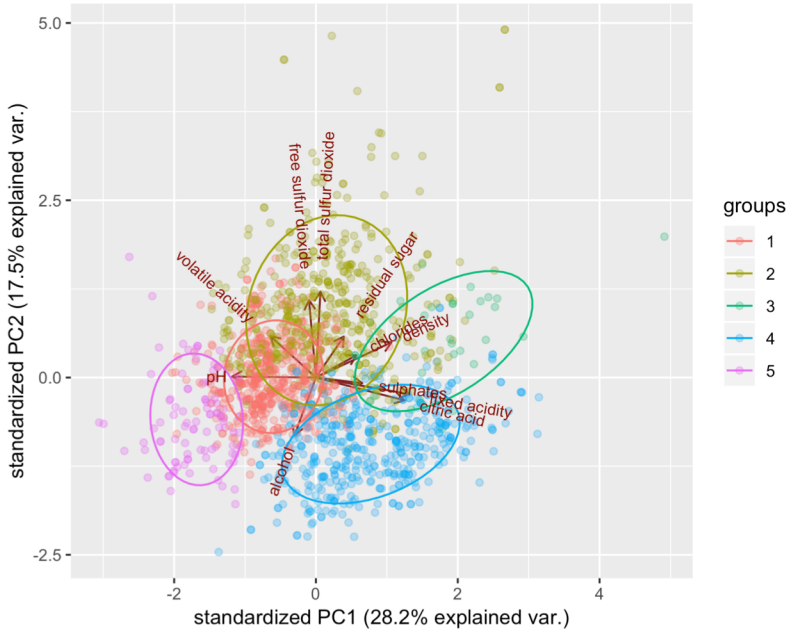
Hierarchical clustering in R

```
1  library(ggdendro)
2  wine2.hc <- hclust(dist(scale(wine2 %>% select(-highq,-quality))), "ward.D2")
3  plot(wine2.hc, labels=F)
4  ggdendrogram(wine2.hc, leaf_labels=F)
```



Visualizing the identified clusters on the PCA visualization

```
1  cutree(wine2.hc, k=5)
2  ggbiplot(wine2.pca, alpha=.3, groups=as.factor(cutree(wine2.hc, k=5)),
   ellipse=T)
```

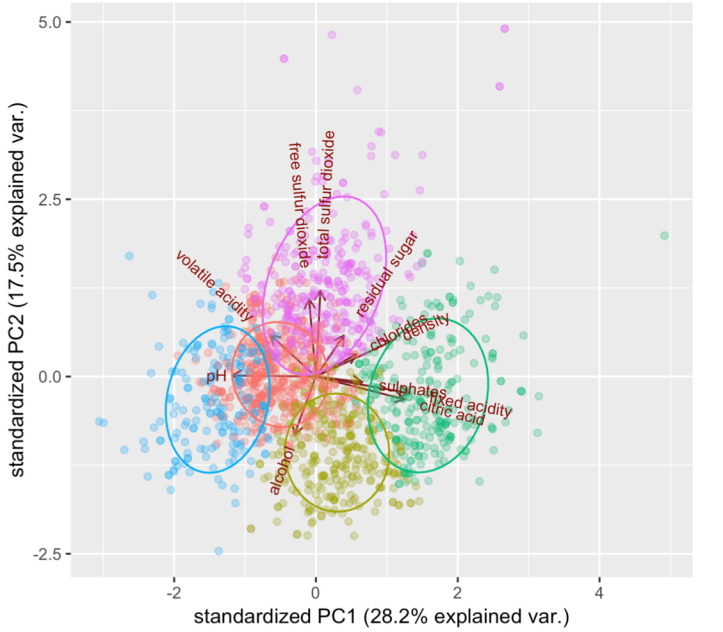


k-means clustering

- k-means clustering starts with k randomly selected observations from the dataset. They are initial cluster centroids.
- Then, all observations are assigned to the cluster centroid.
- Recompute the new cluster centroids.
- Repeat this process until within-cluster variation is minimized.

k-means clustering in R

```
1  wine2.km <- kmeans(scale(wine2 %>% select(-highq,-quality)), 5)
2  wine2.km
3  ggbiplot(wine2.pca, alpha=.3, groups=as.factor(wine2.km$cluster), ellipse=T)
```

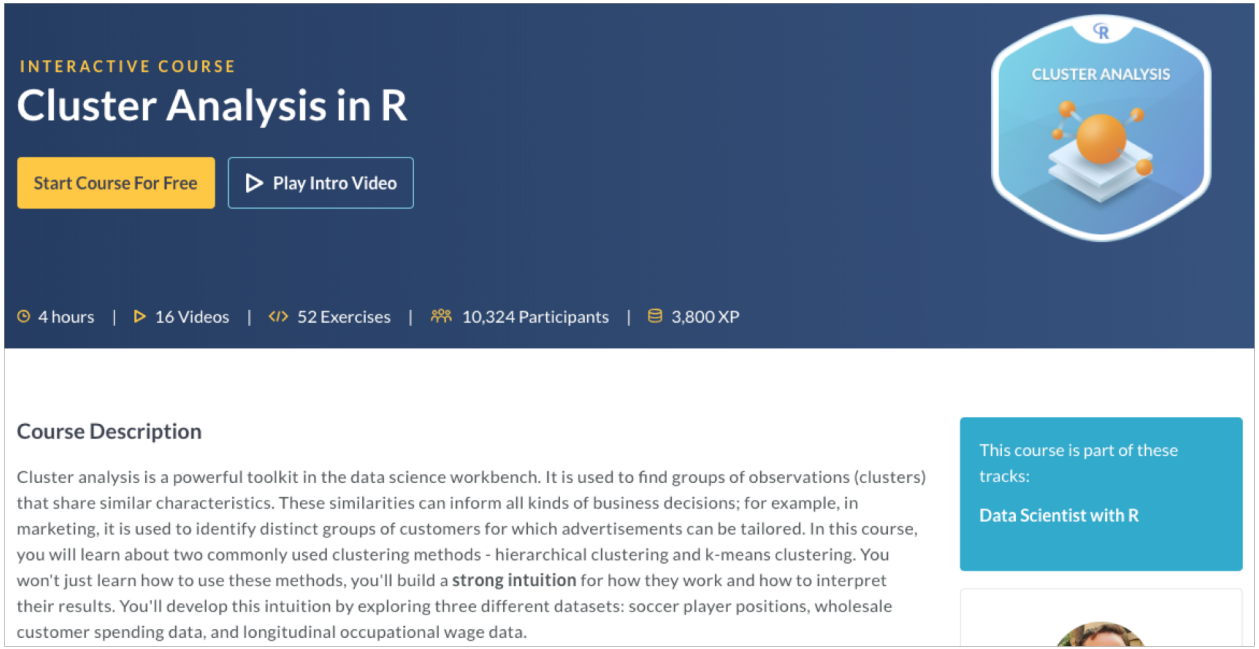



```
> wine2.km
K-means clustering with 5 clusters of sizes 312, 551, 336, 28, 372

Cluster means:
  fixed acidity volatile acidity citric acid residual sugar   chlorides free sulfur dioxide
1   -0.69368839      -0.451790291  -0.1481522     -0.2697385 -0.41238128           0.16593012
2   -0.47148896       0.680058862  -0.8028809     -0.2139763 -0.06864095          -0.39725227
3   -0.05452593       0.047491936   0.1115636      0.4792995  0.02467709           1.10994484
4    0.09538646       0.002199115   1.1811831     -0.3897502  5.78295058          -0.04950011
5    1.32223410      -0.671432064   1.1237966      0.1395907 -0.01002688          -0.54956897
  total sulfur dioxide     density         pH  sulphates    alcohol
1          -0.2091864 -1.22342891  0.6367778  0.1319354  1.2475505
2          -0.3754280 -0.06858235  0.3904764 -0.3992577 -0.4432697
3           1.3609929  0.39085084 -0.1575752 -0.1722083 -0.5629966
4           0.5101700  0.18001552 -1.7352487  3.6622665 -0.8694593
5          -0.5361597  0.76110854 -0.8395025  0.3606067  0.1841870
```

```
Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

DataCamp course

- Complete the following Data Camp courses on clustering.
  - Cluster Analysis in R



Outro

Course Recap

- Week 1: Data Wrangling and Descriptive Statistics
- Week 2: More Data Wrangling, Covariation, and Customizing Visualization
- Week 3: Describing Textual Data
- Week 4: Geographic Visualization
- Week 5: Network Visualization
- Week 6: Interactivity and Dashboard Design
- Week 7: Visualizing Models, Dimensionality Reduction, and Clustering

Creating slides from R Markdown

- Yes, you can do that.
- https://rmarkdown.rstudio.com/lesson-11.html
  - beamer_presentation - PDF presentations with beamer
  - ioslides_presentation - HTML presentations with ioslides
  - slidy_presentation - HTML presentations with slidy
  - powerpoint_presentation - PowerPoint presentation
  - revealjs::revealjs_presentation - HTML presentations with reveal.js

R vs Python

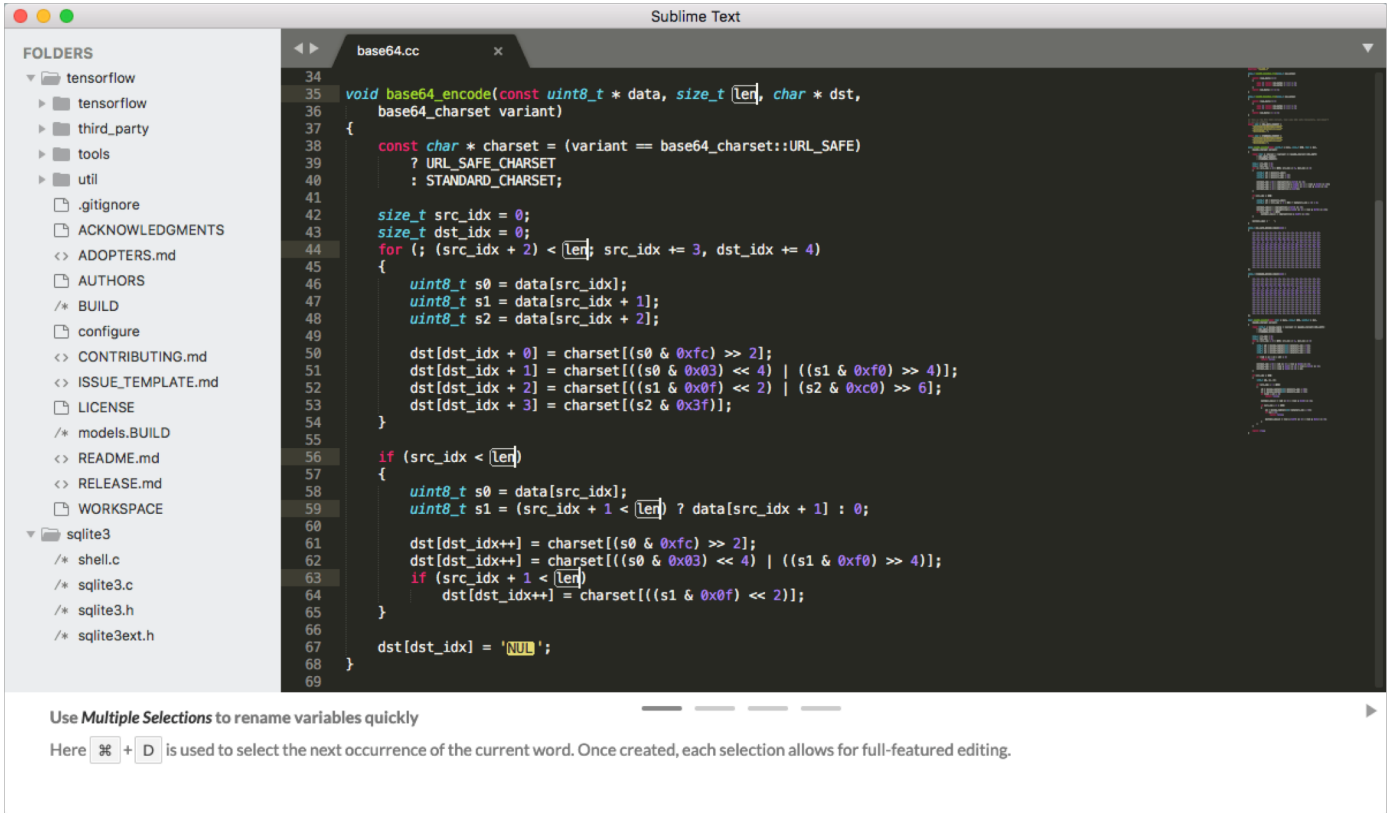
- You can search Google with the keyword such as “r vs python for data science”.
- R and Python are comparable in many aspects.
  - Both are (relatively) easy to program.
  - Programs written in these languages are readable.
  - They have a very active development community producing useful libraries.
- But, they also differ in some aspects.
  - R was developed by statisticians and Python was developed by software engineers.
  - By default, R is designed to handle vectors and matrices naturally.
  - By default, Python is designed to perform general programming tasks.
- Python can be a good complementary skill to R.

Textbooks may be better than MOOCs.

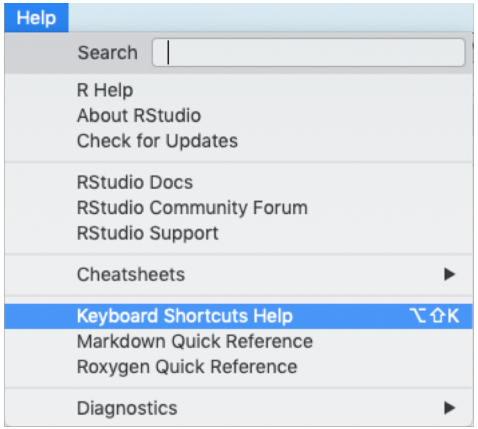
- There are many good lectures online these days.
- But, there are also many good textbooks these days (like R for Data Science).
- MOOCs are great in following guided examples, but they tend to take a long time to complete.
- Textbooks usually contain more detailed information and knowledge on the topic.
- So, be wise in choosing the medium from which you learn a material.
- And there are so many resources out there, so knowing what to learn and where/how to learn is a critical meta knowledge.

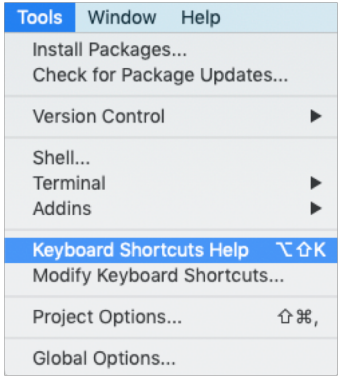
Use a good editor. (Or, spend time to get to know your editor.)

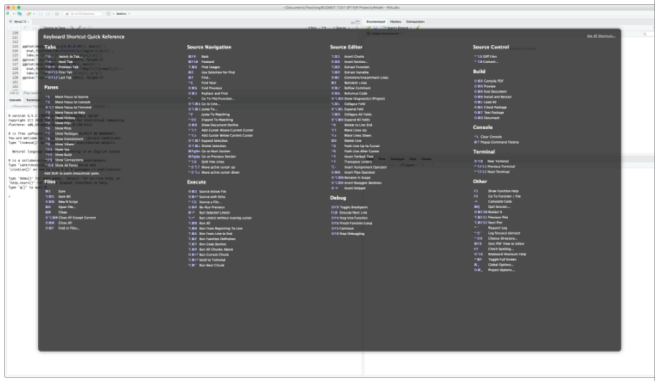
- Sublime Text (https://www.sublimetext.com/)



Use keyboard shortcuts.







RStudio

Keyboard Shortcuts

Console

Description	Windows & Linux	Mac
Move cursor to Console	Ctrl+2	Ctrl+2
Clear console	Ctrl+L	Ctrl+L
Move cursor to beginning of line	Home	Command+Left
Move cursor to end of line	End	Command+Right
Navigate command history	Up/Down	Up/Down
Popup command history	Ctrl+Up	Command+Up
Interrupt currently executing command	Esc	Esc
Change working directory	Ctrl+Shift+H	Ctrl+Shift+H

Source

Description	Windows & Linux	Mac
Goto File/Function	Ctrl+.	Ctrl+.
Move cursor to Source Editor	Ctrl+1	Ctrl+1
New document (except on Chrome/Windows)	Ctrl+Shift+N	Command+Shift+N
New document (Chrome only)	Ctrl+Alt+Shift+N	Command+Shift+Alt+N
Open document	Ctrl+O	Command+O
Save active document	Ctrl+S	Command+S
Close active document (except on Chrome)	Ctrl+W	Command+W
Close active document (Chrome only)	Ctrl+Alt+W	Command+Option+W
Close all open documents	Ctrl+Shift+W	Command+Shift+W
Preview HTML (Markdown and HTML)	Ctrl+Shift+K	Command+Shift+K
Knit Document (knitr)	Ctrl+Shift+K	Command+Shift+K
Compile Notebook	Ctrl+Shift+K	Command+Shift+K
Compile PDF (TeX and Sweave)	Ctrl+Shift+K	Command+Shift+K
Insert chunk (Sweave and Knitr)	Ctrl+Alt+I	Command+Option+I
Insert code section	Ctrl+Shift+R	Command+Shift+R
Run current line/selection	Ctrl+Enter	Command+Enter
Run current line/selection (retain cursor position)	Alt+Enter	Option+Enter
Re-run previous region	Ctrl+Shift+P	Command+Shift+P
Run current document	Ctrl+Alt+R	Command+Option+R
Run from document beginning to current line	Ctrl+Alt+B	Command+Option+B
Run from current line to document end	Ctrl+Alt+E	Command+Option+E
Run the current function definition	Ctrl+Alt+F	Command+Option+F
Run the current code section	Ctrl+Alt+T	Command+Option+T
Run previous Sweave/Rmd code	Ctrl+Alt+P	Command+Option+P
Run the current Sweave/Rmd chunk	Ctrl+Alt+C	Command+Option+C
Run the next Sweave/Rmd chunk	Ctrl+Alt+N	Command+Option+N
Source a file	Ctrl+Shift+O	Command+Shift+O
Source the current document	Ctrl+Shift+S	Command+Shift+S
Source the current document (with echo)	Ctrl+Shift+Enter	Command+Shift+Enter
Fold Selected	Alt+L	Cmd+Option+L
Unfold Selected	Shift+Alt+L	Cmd+Shift+Option+L
Fold All	Alt+O	Cmd+Option+O
Unfold All	Shift+Alt+O	Cmd+Shift+Option+O

My favorite ones for R Studio

Functionality	Windows & Linux	Mac
Go to line	Shift+Alt+G	Cmd+Shift+Option+G
Previous tab	Ctrl+F11	Ctrl+F11
Next tab	Ctrl+F12	Ctrl+F12
Move Lines Up/Down	Alt+Up/Down	Option+Up/Down
Copy Lines Up/Down	Shift+Alt+Up/Down	Command+Option+Up/Down
Delete Line	Ctrl+D	Cmmand+D
Move focus to Source Editor	Ctrl+1	Ctrl+1
Move focus to Console	Ctrl+2	Ctrl+2
Insert assignment operator	Alt+-	Option+-
Insert pipe operator	Ctrl+Shift+M	Cmd+Shift+M

Weekly Recap

Things we covered this week

- Linear Regression
  - Basics
  - Diagnostic Visualizations
- Logistic Regression
  - Odds Ratio
  - Interpreting coefficients
- Dimensionality Reduction
  - PCA
  - t-SNE
- Clustering
  - Hierarchical
  - K-Means
- Outro
  - Course Recap
  - R vs Python
  - General Programming Advice

Things to do this week

- 4 Quizzes (Due: Thursday, February 21, 11:59pm)
  - Week 7.1. Linear Regression
  - Week 7.2. Logistic Regression
  - Week 7.3. Dimensionality Reduction
  - Week 7.4. Clustering
- 1 Weekly Problem (Due: Friday, February 22, 11:59pm)
- 2 DataCamp Courses (Due: Saturday, February 23, 11:59pm)
  - Correlation and Regression
  - Cluster Analysis in R