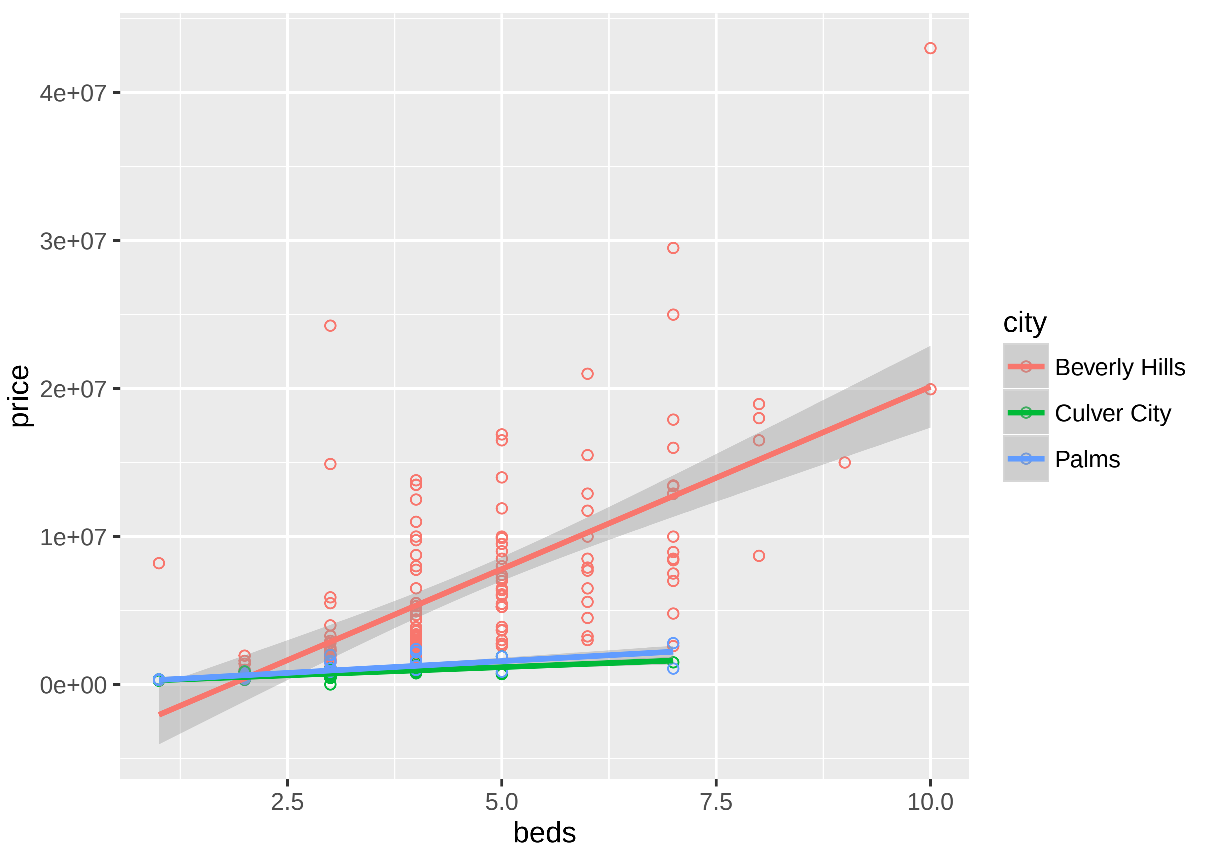
Jang.Junhyuk

Junhyuk Jang

April 14, 2017

SID : 004 728 134 LEC : 2 DIS : 2B

```r
library(ggplot2)
library("reshape2")
# Q1
df <- read.csv("~/Desktop/UCLA_Academic/Spring 2017/STAT 101_C/HW/LArealestate.csv")
df <- df[complete.cases(df),]
attach(df)
df$city[df$city == "culver city"] = "Culver City"
ggplot(df,aes(x=beds,y = price,color = city))+
        geom_point(shape = 1) +
        geom_smooth(method = lm)
```




```r
# By increasing the number of beds which cities house price is most rapidly
# grow?
# Based on my qplot, it is obivous that by incresing beds, the house price
# of "beverly hills" is most rapidly growth.
# There no rapid rapid price growth for "Culver city","Palms".
```

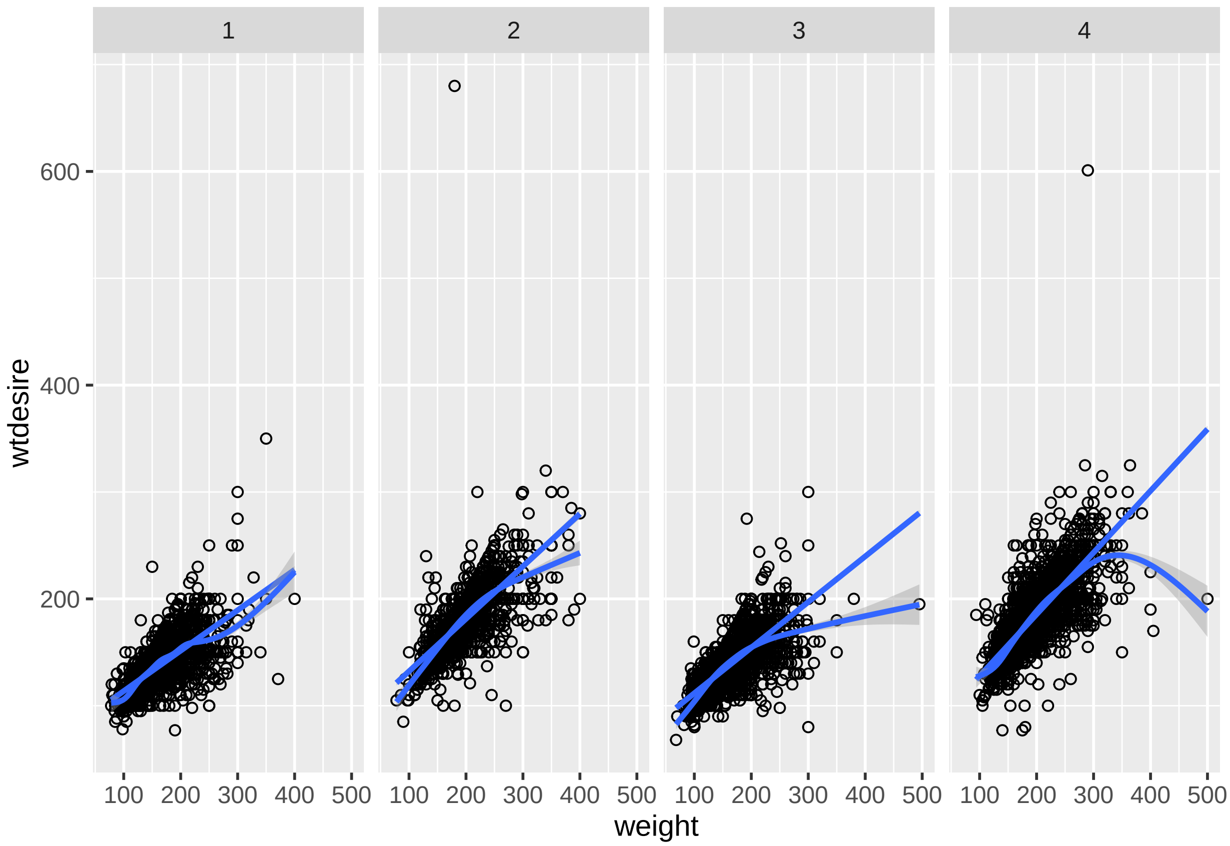
```r
# Q2
df <- read.csv("~/Desktop/UCLA_Academic/Spring 2017/STAT 101_C/HW/cdc.csv")
head(df)
```

```
##   state    genhlth physhlth exerany hlthplan smoke100 height weight
## 1    22       good        0       0        1        0     70    175
## 2    25       good       30       0        1        1     64    125
## 3     6       good        2       1        1        1     60    105
## 4     6       good        0       1        1        0     66    132
## 5    39 very good        0       0        1        0     61    150
## 6    42 very good        0       1        1        0     64    114
##   wtdesire age gender
## 1      175  77      m
## 2      115  33      f
## 3      105  49      f
## 4      124  42      f
## 5      130  55      f
## 6      114  55      f
```

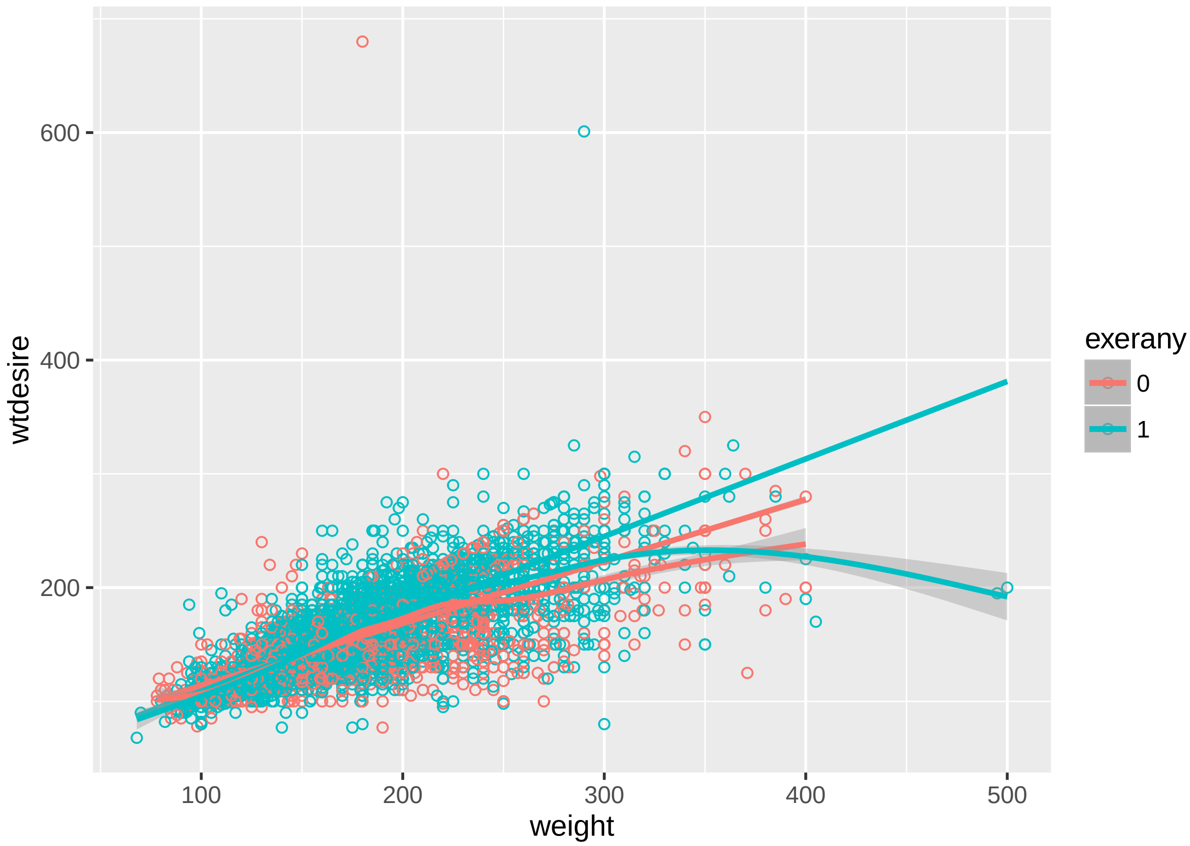
```r
df$exerany <- as.factor(df$exerany)

df$group[df$exerany == "0" & df$gender == "f"] <- 1
df$group[df$exerany == "0" & df$gender == "m"] <- 2
df$group[df$exerany == "1" & df$gender == "f"] <- 3
df$group[df$exerany == "1" & df$gender == "m"] <- 4

sp <- ggplot(df,aes(x = weight,y = wtdesire)) + geom_point(shape = 1)
sp + facet_grid(.~group) + geom_smooth(method = lm) + geom_smooth()
```




```
sp1 <- ggplot(df,aes(x = weight,y = wtdesire,color = exerany)) + geom_point(shape = 1)
sp1 + geom_smooth(method = lm) + geom_smooth()
```




```
# (a) There is positve linear relationship between weight and desire weight.
# (b) In this case the plots have a linear pattern. So, it would be better
# to make our model with low felxibility. When we use smooth line, the smooth line
# too hard to find a pattern that it is overfitting the plots. In other words,smooth model
# could be less biased than inflexible model but in this case will have too high variance.


# Q3

df <- read.table("~/Desktop/UCLA_Academic/Spring 2017/STAT 101_C/HW/banknote.csv",header = T)
library(class)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.5

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.2.5
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.2.5
```

```r
normalize <- function(x){return((x-min(x))/(max(x)-min(x)))}
attach(df)
df_norm <- cbind(as.data.frame(lapply(df[,1:6],normalize)),Y)
df_norm$Y = as.factor(df_norm$Y)
summary(df_norm)
```

```
##      Length            Left            Right            Bottom
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3200   1st Qu.:0.4500   1st Qu.:0.3333   1st Qu.:0.1818
##  Median :0.4400   Median :0.6000   Median :0.4762   Median :0.3455
##  Mean   :0.4384   Mean   :0.5607   Mean   :0.4555   Mean   :0.4032
##  3rd Qu.:0.5200   3rd Qu.:0.7000   3rd Qu.:0.5833   3rd Qu.:0.6182
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##       Top            Diagonal       Y
##  Min.   :0.0000   Min.   :0.0000   0:100
##  1st Qu.:0.5217   1st Qu.:0.3696   1:100
##  Median :0.6304   Median :0.5761
##  Mean   :0.6414   Mean   :0.5834
##  3rd Qu.:0.7609   3rd Qu.:0.8043
##  Max.   :1.0000   Max.   :1.0000
```

```r
set.seed(33445566)
sample <- sample(seq(1,200),140,replace = F)
df_train <- df_norm[sample,]
df_test <- df_norm[-sample,]

df_train$Y = as.factor(df_train$Y)
train_control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
knn_fit <- train(Y ~., data = df_train, method = "knn",
                 trControl=train_control,
                 preProcess = c("center", "scale"),
                 tuneLength = 10)

knn_fit
```

```
## k-Nearest Neighbors
##
## 140 samples
##   6 predictor
##   2 classes: '0', '1'
##
## Pre-processing: centered (6), scaled (6)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 125, 126, 126, 125, 126, 127, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.9880708  0.9761117
##    7  0.9856899  0.9713498
##    9  0.9856899  0.9713498
##   11  0.9880708  0.9761117
##   13  0.9880708  0.9761117
```

```
##   15  0.9930159  0.9859717
##   17  0.9930159  0.9859717
##   19  0.9930159  0.9859717
##   21  0.9930159  0.9859717
##   23  0.9907937  0.9814672
##
## Accuracy was used to select the optimal model using  the largest value.
## The final value used for the model was k = 21.
```

```r
# k = 1
m1 <- knn(train = df_train[,1:6],test = df_test[,1:6],cl = df_train[,7],k=1)
(t1 <- table(df_test[,7],m1))
```

```
##    m1
##      0  1
##   0 33  1
##   1  0 26
```

```r
(accur1 <- (t1[1,1]+t1[2,2])/(sum(t1)))
```

```
## [1] 0.9833333
```

```r
# k = 3
m3 <- knn(train = df_train[,1:6],test = df_test[,1:6],cl = df_train[,7],k=3)
(t3 <- table(df_test[,7],m3))
```

```
##    m3
##      0  1
##   0 34  0
##   1  0 26
```

```r
(accur3 <- (t3[1,1]+t3[2,2])/(sum(t3)))
```

```
## [1] 1
```

```r
# k = 5
m5 <- knn(train = df_train[,1:6],test = df_test[,1:6],cl = df_train[,7],k=5)
(t5 <- table(df_test[,7],m5))
```

```
##    m5
##      0  1
##   0 33  1
##   1  0 26
```

```r
(accur5 <- (t5[1,1]+t5[2,2])/(sum(t5)))
```

```
## [1] 0.9833333
```

```r
# Misclassification rate
(Miss1 <-1 / 60)
```

```
## [1] 0.01666667
```

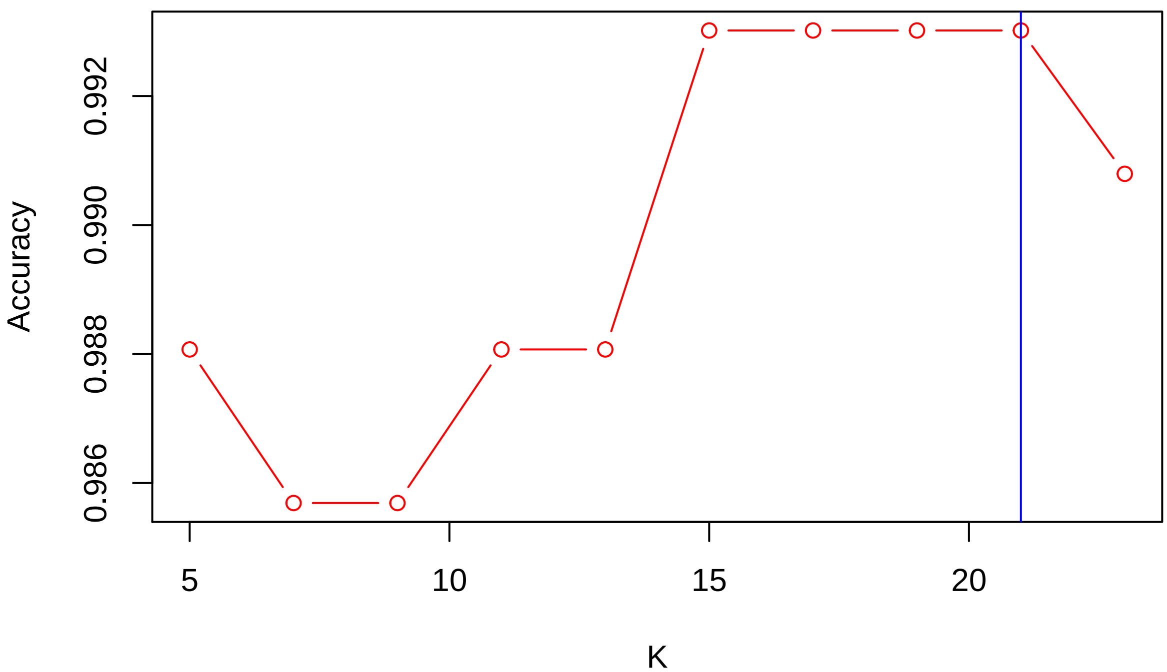
```r
(Miss3 <- 0 /60)
```

```
## [1] 0
```

```r
(Miss5 <- 0 / 60)
```

```
## [1] 0
```

```r
vv <- data.frame(knn_fit[4])
plot(vv[,1],vv[,2],type = "b",col = "Red",xlab = "K",
     ylab = "Accuracy")
abline(v = 21, col = "blue")
```




```r
# Best k that maximizes the accuracy of my classifier is
# K = 21 based on k vs accuracy.

# Q4) 2.4.7
obs1 <- c(0,3,0)
obs2 <- c(2,0,0)
obs3 <- c(0,1,3)
obs4 <- c(0,1,2)
obs5 <- c(-1,0,1)
obs6 <- c(1,1,1)
t <- c(0,0,0)
# (a)
dist(rbind(t,obs1))
```

```
##      t
## obs1 3
```

```r
dist(rbind(t,obs2))
```

```
##      t
## obs2 2
```

```r
dist(rbind(t,obs3))
```

```
##             t
## obs3 3.162278
```

```r
dist(rbind(t,obs4))
```

```
##             t
## obs4 2.236068
```

```r
dist(rbind(t,obs5))
```

```
##             t
## obs5 1.414214
```

```r
dist(rbind(t,obs6))
```

```
##             t
## obs6 1.732051
```

```r
# (b)
# If k = 1 our test point will be color "Green" because d5 is the closest from
# t. Assigned color for d5 is green and it is the closest from t indicates that
# the point classified as green under k = 1.
# (c)
# If k = 3, the 3 nearest points are d5 = 1.414214,d6 = 1.732051,d2 = 2.
# Assigned color for obs5 is green, for obs6 is Red and for obs2 is Red.
# Color "Green"" and "Red" has 1:2 ratio implies that I have to classify the point
# as "Red" under k = 3
```