Lab 04 Regression Analysis

Lusine Zilfimian

March 11 (Wednesday), 2020

Contents

Needed packages

- These packages are required for this Session

```
library(knitr) # for kable()
library(dplyr) # for data manipulation
library(ggplot2) # for visualization
library(ggthemes) # for theme_pender()
library(stargazer) # for stargazer()
library(MASS) # for stepAIC()
library(car) # for vif()
```

- Install these packages if you do not have them

Data preparation

- Let's load and use the subset of the data

```
hd <- read.csv("housing.csv")[100:200,]; colnames(hd)
```

```
##  [1] "id"                "date"              "price"
##  [4] "bedrooms"          "bathrooms"         "sqft_living"
##  [7] "floors"            "waterfront"        "view"
## [10] "condition"         "grade"             "zipcode"
## [13] "Sqft_with_garden"
```

```
hd <- dplyr::select(hd, c("price", "sqft_living", "condition",
  "Sqft_with_garden"))
hd$price = hd$price/1000
table(hd$condition)
```

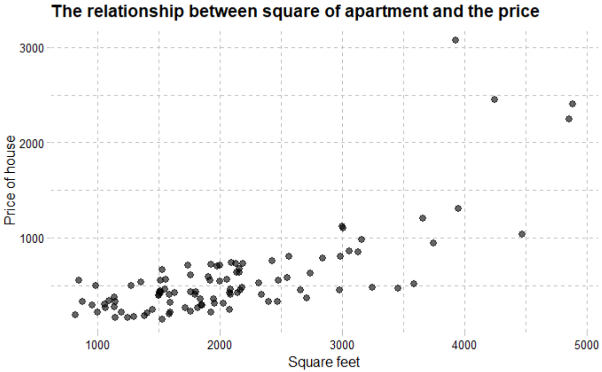
```
##
##  2  3  4  5
##  1 56 34 10
```

```
hd$condition <- factor(ifelse(hd$condition == 2 | hd$condition == 3,
  "fair", ifelse(hd$condition == 4, "good", "excellent")))
```

Undestanding the data

- Visualization of the main numeric variables:

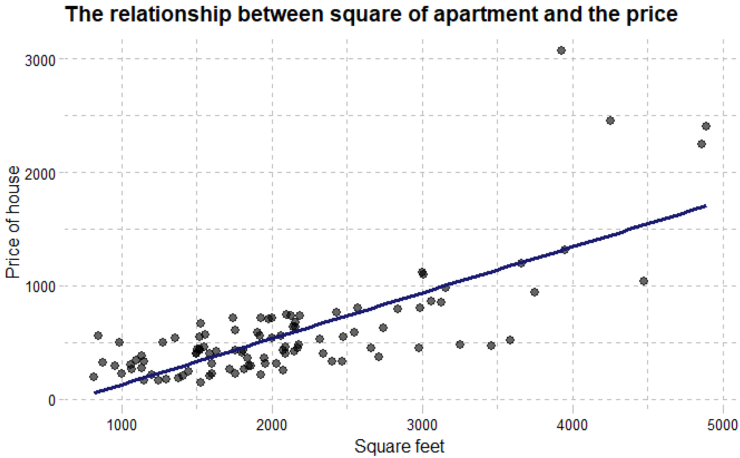
```
g1 <- ggplot(hd, aes(x = sqft_living, y = price))+
        geom_point(size = 2.5, alpha = 0.6)+
        ggtitle("The relationship between square of apartment and the price
        xlab("Square feet")+
        ylab("Price of house")+
        theme_pander()
```



Undestanding the data

- Adding the regression line

```
g2 <- g1+
  geom_smooth(method = "lm", se = F, col = "midnightblue", size = 1.2)
```



Intercept-only model

```
model0 <- lm(price ~ 1, data = hd)
names(model0)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "call"          "terms"         "model"
```

```
model0$coefficients
```

```
## (Intercept)
##    575.0682
```

Intercept-only model

```r
summary(model0)
```

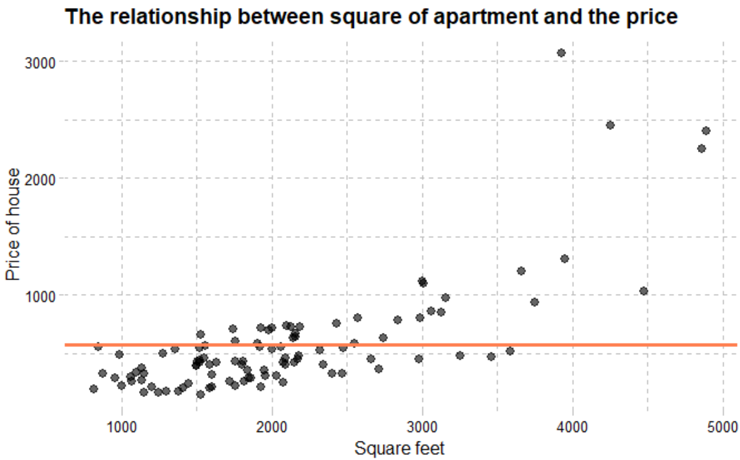
```
##
## Call:
## lm(formula = price ~ 1, data = hd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -427.57 -258.57 -125.07   88.93 2494.93
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   575.07      47.06   12.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 473 on 100 degrees of freedom
```

```r
mean(hd$price)
```

```
## [1] 575.0682
```

Intercept-only model

```
g3 <- g1 + geom_hline(yintercept = model0$coefficients ,
  col = "coral", size = 1.2)
```



Regression with one explanatory variable

```
model1 <- lm(price ~ sqft_living, data = hd)
coef(model1)
```

```
##  (Intercept)  sqft_living
## -275.6591736    0.4049002
```

- $\hat{\beta}_0 =$ -275.6591736
- $\hat{\beta}_1 =$ 0.4049002
- $\hat{Price} = \hat{\beta}_0 + \hat{\beta}_1 sqrt\_living$

Regression with one explanatory variable

```r
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ sqft_living, data = hd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -662.93 -167.58   -7.02  140.02 1754.40
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -275.65917   80.41911  -3.428 0.000888 ***
## sqft_living    0.40490    0.03532  11.465  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 311.6 on 99 degrees of freedom
## Multiple R-squared:  0.5704, Adjusted R-squared:  0.5661
## F-statistic: 131.4 on 1 and 99 DF,  p-value: < 2.2e-16
```

Understanding the output of regression in R

- Let's derive the Resudials table:

```
min(resid(model1))
```

```
## [1] -662.9326
```

```
max(resid(model1))
```

```
## [1] 1754.401
```

```
median(resid(model1))
```

```
## [1] -7.015203
```

```
quantile(resid(model1), probs = c(0.25,0.75))
```

```
##       25%       75%
## -167.5823  140.0229
```

Understanding the output of regression in R

- Beta coefficients and SE

```
kable(head(X <- data.frame("X0" = 1,  "X1" = hd[,"sqft_living"])))
```

X0	X1
1	2340
1	2160
1	2320
1	1384
1	1820
1	2130

```
X <- data.matrix(X) # to apply the solve() function
Y <- data.matrix(hd[,"price"])
(b <- solve(t(X) %*% X) %*% t(X) %*% Y)
```

```
##             [,1]
## X0 -275.6591736
## X1    0.4049002
```

Understanding the output of regression in R

- Beta coefficients and SE

```
(RSE <- sqrt(sum((resid(model1)^2))/(dim(hd)[1] - 2)))
```

```
## [1] 311.5822
```

```
(bse <- RSE/sqrt(sum((hd$sqft_living - mean(hd$sqft_living))^2)))
```

```
## [1] 0.03531637
```

Understanding the output of regression in R

- t and p values (RSE is the same)

```r
(t <- coef(model1)[2]/bse)
```

```
## sqft_living
##    11.46494
```

```r
(p.value_t <- 2*pt(-t, df = 99))
```

```
##  sqft_living
## 7.221469e-20
```

```r
(p.value_t <- 2*pt(-3.428, df = 99))
```

```
## [1] 0.0008873304
```

```r
(RSE <- sqrt(sum((resid(model1)^2))/(dim(hd)[1] - 2)))
```

```
## [1] 311.5822
```

```r
summary(model1)$sigma
```

```
## [1] 311.5822
```

Understanding the output of regression in R

- R^2 and F

```
(Rsquare <- sum((predict(model1) - mean(hd$price))^2)/
    sum(((hd$price - mean(hd$price))^2)))
```

```
## [1] 0.5703962
```

```
var(model1$fitted.values)/var(hd$price)
```

```
## [1] 0.5703962
```

```
cor(model1$fitted.values, hd$price)^2
```

```
## [1] 0.5703962
```

```
summary(model1)$r.sq
```

```
## [1] 0.5703962
```

```
(Fstat <- t^2)
```

```
## sqft_living
##    131.4449
```

Understanding the output of regression in R

- p values and confidence intervals

```
(p.valuef <- 2*pf(-Fstat,df1 = 2-1, df2 = 101-2))
```

```
## sqft_living
##           0
```

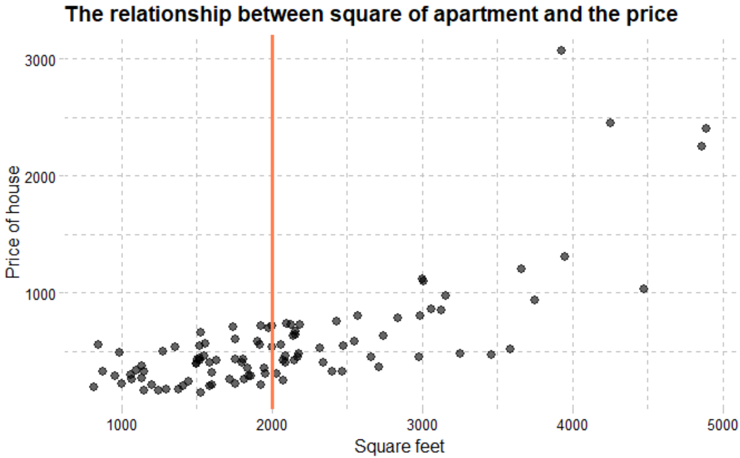
```
confint(model1)
```

```
##                  2.5 %       97.5 %
## (Intercept) -435.2281421 -116.0902050
## sqft_living    0.3348249    0.4749756
```

Interpretation

- The average or expected value given corresponding X

```
g4 <- g1 + geom_vline(xintercept = 2000, col = "coral", size = 1.2)
```



Prediction

- What will be the predicted price for the first three observations of the data?

```r
pred.dat <- hd[1:3,]
pred.dat
```

```
##       price sqft_living condition Sqft_with_garden
## 100 404.95        2340      good             2351
## 101 671.50        2160 excellent             2269
## 102 530.00        2320      fair             2477
```

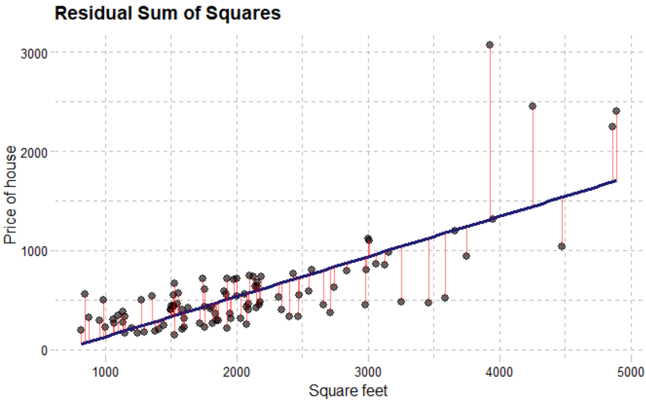
```r
predict(model1, newdata = pred.dat)
```

```
##      100      101      102
## 671.8073 598.9253 663.7093
```

TSS, ESS, RSS

- Visualisation of RSS

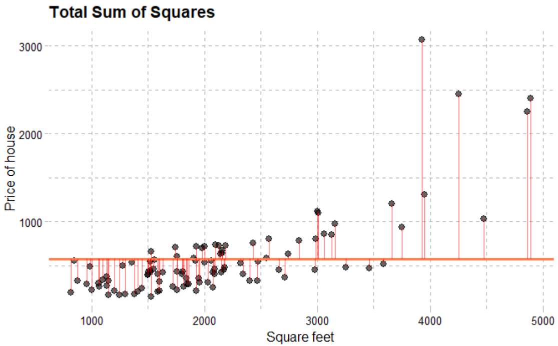
```
g5 <- g2 +
    geom_segment(aes(xend = sqft_living, yend = predict(model1)),
      alpha=0.5, col = "red")+
   ggtitle("Residual Sum of Squares")
```



TSS, ESS, RSS

- Visualisation of TSS

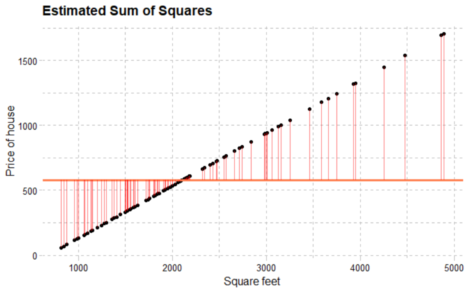
```
g6 <- g1 +
  geom_hline(yintercept = model0$coefficients , col = "coral", size = 1.2)+
  geom_segment(aes(xend = sqft_living, yend = mean(price)),
    alpha=0.5, col = "red")+
  ggtitle("Total Sum of Squares")
```



TSS, ESS, RSS

- Visualisation of ESS

```
g7 <- ggplot(hd, aes(x = sqft_living, fitted(model1)))+
  geom_point()+
  geom_hline(yintercept = model0$coefficients , col = "coral", size = 1.2)+
  geom_segment(aes(xend = sqft_living, yend = mean(price)),
    alpha=0.5, col = "red") +
    xlab("Square feet") + ylab("Price of house") +
      theme_pander() + ggtitle("Estimated Sum of Squares")
```



Multiple Regression Model Selection

- Let's consider the following models

```
model2 <- lm(price ~. ,data = hd)
model3 <- lm(price ~.-condition , data = hd)
model4 <- lm(price ~ sqft_living + condition, data = hd)
```

Multiple Regression Model Selection

```
summary(model3)
```

```
##
## Call:
## lm(formula = price ~ . - condition, data = hd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -650.70 -160.09  -14.93  136.39 1724.98
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -247.2449    96.4261  -2.564   0.0119 *
## sqft_living         0.7269     0.5990   1.213   0.2279
## Sqft_with_garden   -0.3200     0.5942  -0.539   0.5914
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 312.7 on 98 degrees of freedom
## Multiple R-squared:  0.5717, Adjusted R-squared:  0.5629
## F-statistic:  65.4 on 2 and 98 DF,  p-value: < 2.2e-16
```

F test

```
(Fstat4<-(summary(model3)$r.sq/(3-1))/((1-summary(model3)$r.sq)/(101-3)))
```

```
## [1] 65.39611
```

Regression with Categorical variables

```
model4 <- lm(price ~ sqft_living + condition, data = hd)
summary(model4)
```

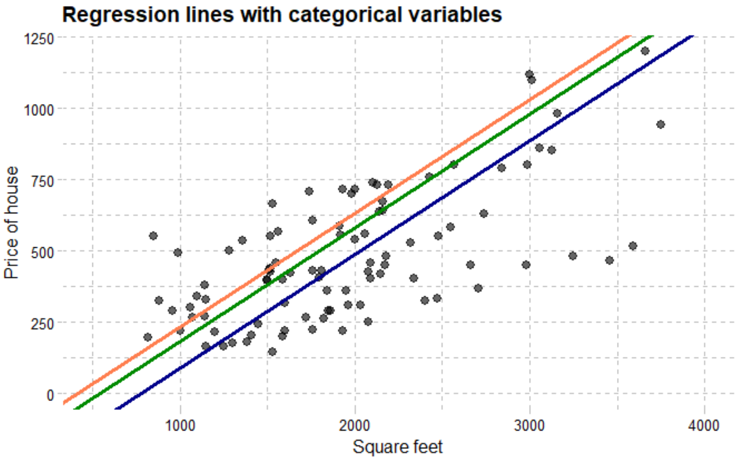
```
##
## Call:
## lm(formula = price ~ sqft_living + condition, data = hd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -607.5 -139.8  -34.6  117.5 1717.4
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -164.16439  130.87350  -1.254    0.213
## sqft_living       0.39864    0.03545  11.247   <2e-16 ***
## conditionfair  -144.47088  107.21565  -1.347    0.181
## conditiongood   -49.93705  112.17615  -0.445    0.657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Regression with Categorical variables

- The levels of condition are fair, good, excellent. One of the categories is the base. R takes the one that comes first in alphabetical order

```
g8 <- g1+
  # excellent
  geom_abline(intercept = coef(model4)[1],
    slope = coef(model4)[2], col = "coral", size = 1.2) +
  # fair
  geom_abline(intercept = coef(model4)[1] + coef(model4)[3],
    slope = coef(model4)[2], col = "darkblue", size = 1.2) +
  #good
  geom_abline(intercept = coef(model4)[1] + coef(model4)[4],
    slope = coef(model4)[2], col = "green4", size = 1.2) +
  ggtitle("Regression lines with categorical variables") +
  ylim(1, 1200) + xlim(500, 4000)
```

Regression with Categorical variables




```r
coef(model4)
```

```
##   (Intercept)   sqft_living conditionfair conditiongood
##  -164.1643864     0.3986409  -144.4708793   -49.9370479
```

Non-linearity

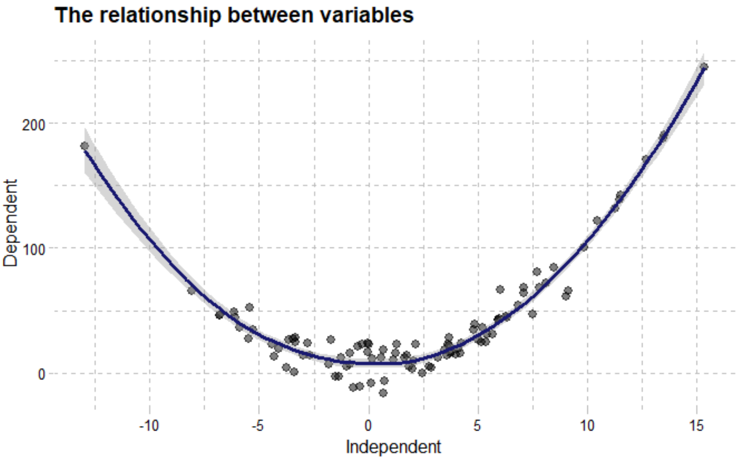
- Non-linear by x

```r
set.seed(27)
x1 = rnorm(100, mean = 2, sd = 5)
y1 = x1^2 + rnorm(100, mean = 5, sd = 10)
y2 = x1^3 + x1^2 + x1 + rnorm(100, mean = 5, sd = 100)

poly_df <- data.frame(x1, y1, y2)

g9 <- ggplot(poly_df, aes(x = x1, y = y1)) +
  geom_point(size = 2.5, alpha = 0.5) +
  ggtitle("The relationship between variables") +
  xlab("Independent") +
  ylab("Dependent") +
  geom_smooth(method = "auto", col = "midnightblue", size = 1.2) +
  theme_pander()
```

Non-linearity

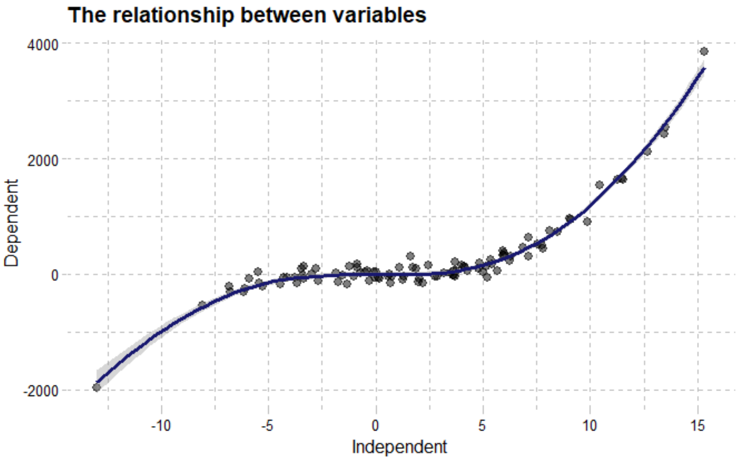


Non-linearity in R

```r
poly_1 <- lm(formula = y1 ~ I(x1^2), data = poly_df)
summary(poly_1)
```

```
##
## Call:
## lm(formula = y1 ~ I(x1^2), data = poly_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.6027  -5.6713   0.4195   6.1071  24.3730
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.08906    1.21250   5.022  2.3e-06 ***
## I(x1^2)      1.00146    0.02133  46.940  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.818 on 98 degrees of freedom
## Multiple R-squared:  0.9574, Adjusted R-squared:  0.957
## F-statistic:  2203 on 1 and 98 DF,  p-value: < 2.2e-16
```

Non-linearity in R



Non-linearity in R

```r
poly_2 <- lm(formula = y2 ~ poly(x1, 3), data = poly_df)
summary(poly_2)
```

```
##
## Call:
## lm(formula = y2 ~ poly(x1, 3), data = poly_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -245.502  -75.148   -8.598   69.365  305.287
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    237.74      10.57   22.50   <2e-16 ***
## poly(x1, 3)1  5478.08     105.66   51.84   <2e-16 ***
## poly(x1, 3)2  2563.99     105.66   24.27   <2e-16 ***
## poly(x1, 3)3  3139.94     105.66   29.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105.7 on 96 degrees of freedom
```

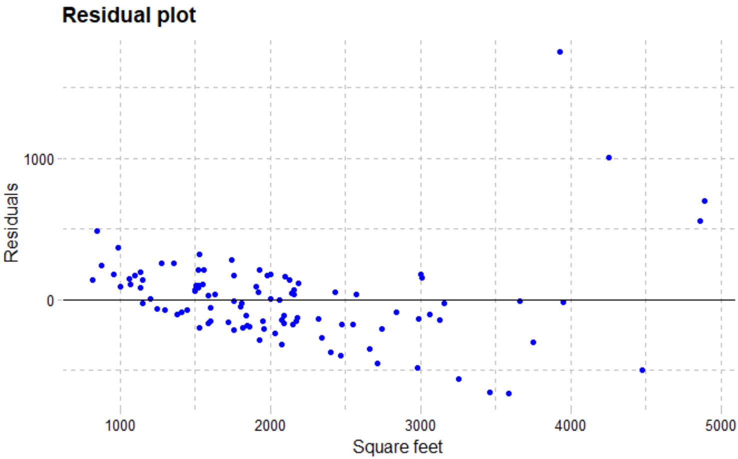
Heteroskedasticity

- Residual plot

```
g11 <- ggplot(data = hd, aes(y = model1$residuals, x = sqft_living)) +
  geom_point(col = 'blue') +
  geom_abline(slope = 0) +
  xlab("Square feet") +
  ylab("Residuals") +
  theme_pander() +
  ggtitle("Residual plot")
```

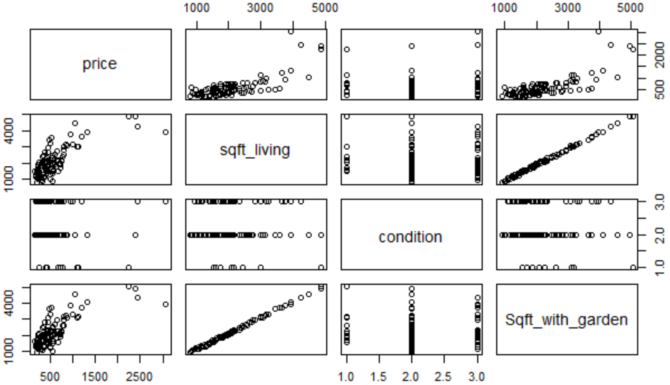
There is more variation in price for houses with a more large square of area.

Heteroskedasticity



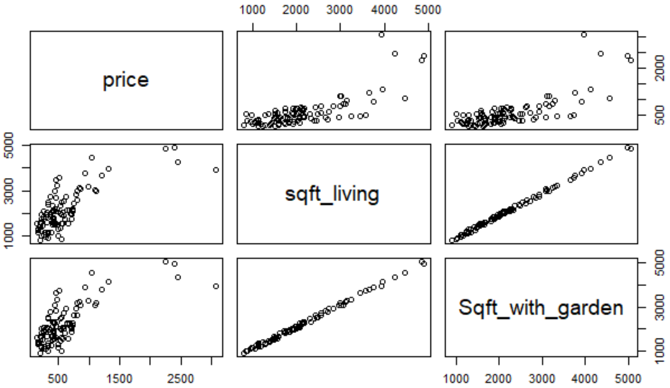
Multicollinearity

- Suspicion of multicollinearity
- Meaningful correlation with numerics and price
- pairs(hd)



Multicollinearity

- pairs(hd[, c("price","sqft_living", "Sqft_with_garden")])



Detecting the multicollinearity

```r
cor(hd[ , c("price","sqft_living", "Sqft_with_garden")])
```

```
##                      price sqft_living Sqft_with_garden
## price            1.0000000   0.7552458        0.7518162
## sqft_living      0.7552458   1.0000000        0.9982481
## Sqft_with_garden 0.7518162   0.9982481        1.0000000
```

```r
cor(hd[ , c("price","sqft_living", "Sqft_with_garden")])[2,3]
```

```
## [1] 0.9982481
```

```r
model3 <- lm(price ~ sqft_living + Sqft_with_garden, data = hd)
model3_sub <- lm(price ~ sqft_living, data = hd)
stargazer(model3, model3_sub,
  title = "Multicollinearity",
  out.header = FALSE,
  type = "latex",
  header=FALSE,
  covariate.labels = c(
    "Square feet",
    "With garder"))
```

Table 3: Multicollinearity

	Dependent variable:	
	price	
	(1)	(2)
Square feet	0.727	0.405^{***}
	(0.599)	(0.035)
With garder	-0.320	
	(0.594)	
Constant	-247.245^{**}	-275.659^{***}
	(96.426)	(80.419)
Observations	101	101
R^2	0.572	0.570
Adjusted R^2	0.563	0.566
Residual Std. Error	312.706 (df = 98)	311.582 (df = 99)
F Statistic	65.396^{***} (df = 2; 98)	131.445^{***} (df = 1; 99)
Note:		$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

Detecting the multicollinearity: VIF

```
vif(model3)
```

```
 sqft_living Sqft_with_garden
    285.6559         285.6559
```

```
mod_vif <- lm(sqft_living ~ Sqft_with_garden, data = hd)
stargazer(mod_vif, type = "latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Wed, Mar 11, 2020 - 11:02:57 AM

Table 4

	Dependent variable:
	sqft_living
Sqft_with_garden	0.990^{***}
	(0.006)
Constant	-80.571^{***}
	(14.005)

Model Selection

- Stepwise regression Based on AIC

```
set.seed(2708)
sample <- sample(nrow(hd), round(nrow(hd)*0.8))
Train <- hd[sample, ]
Test <- hd[-sample, ]
model_full <- lm(price ~ ., data = Train)
```

```r
model_step <- stepAIC(model_full, direction = "backward")
```

```
## Start:  AIC=939.91
## price ~ sqft_living + condition + Sqft_with_garden
##
##                    Df Sum of Sq     RSS    AIC
## - Sqft_with_garden  1     53748 7894711 938.47
## - sqft_living       1    184013 8024976 939.79
## <none>                          7840963 939.91
## - condition         2    398527 8239490 939.93
##
## Step:  AIC=938.47
## price ~ sqft_living + condition
##
##               Df Sum of Sq      RSS     AIC
## <none>                      7894711  938.47
## - condition    2    428167  8322877  938.75
## - sqft_living  1  11687723 19582434 1010.05
```

Model Selection

- `938.47` - AIC if you take out the variable Sqft_with_garden
- `<none>  7840963 939.91` - The model with all the variables
- `<none> 7894711 938.47` - The model, where none of the variables is excluded has the lowest AIC, thus no more changes.

Stepwise regression

```r
summary(model_step)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + condition, data = Train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -668.47 -153.60  -28.82  143.68 1589.04
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -222.07940  149.67519  -1.484    0.142
## sqft_living      0.43153    0.04042  10.677   <2e-16 ***
## conditionfair -143.64465  123.35731  -1.164    0.248
## conditiongood    7.12732  130.40833   0.055    0.957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 320.2 on 77 degrees of freedom
## Multiple R-squared:  0.6209, Adjusted R-squared:  0.6061
## F-statistic: 42.03 on 3 and 77 DF,  p-value: 3.406e-16
```

RMSE

```
pred_model_step <- predict(model_step, newdata = Test)
RMSE1 <- sqrt(mean((pred_model_step - Test$price)^2))
RMSE1
```

```
## [1] 289.2494
```

```
pred_model_3 <- predict(model3, newdata = Test)
RMSE2 <- sqrt(mean((pred_model_3 - Test$price)^2))
RMSE2
```

```
## [1] 242.2185
```

```
pred_model_full <- predict(model_full, newdata = Test)
RMSE3 <- sqrt(mean((pred_model_full - Test$price)^2))
RMSE3
```

```
## [1] 293.4834
```