Code

ADA2: Class 14, Ch 07b, Analysis of Covariance

Name Here

March 10, 2022

This is a challenging dataset, in part because it's real and messy. I will guide you through a simplified sensible analysis, but other models are possible.

Note that I needed to set `cache=FALSE` to assure all output was updated.

1 ANCOVA model: Albuquerque NM 87108, House and Apartment listing prices

Prof Erhardt constructed a dataset of listing prices for dwellings (homes and apartments) for sale from Zillow.com (http://www.zillow.com/homes/for_sale/Albuquerque-NM-87108/95303_rid/any_days/35.095087-106.52167835.035021-106.633258_rect/13_zm/0_mmm/) on Feb 26, 2016 at 1 PM for Albuquerque NM 87108. In this assignment we'll develop a model to help understand which qualities that contribute to a **typical dwelling's listing price**. We will then also predict the listing prices of new listings posted on the following day, Feb 27, 2016 by 2 PM.

Because we want to model a *typical dwelling*, it is completely reasonable to remove "unusual" dwellings from

the dataset. Dwellings have a distribution with a long tail (https://en.wikipedia.org/wiki/Long_tail)!

1.1 Unusual assignment, not top-down, but up-down-up-down

This is an unusual assignment because the workflow of this assignment isn't top-down; instead, you'll be scrolling up and down as you make decisions about the data and model you're fitting. Yes, I have much of the code worked out for you. However, there are data decisions to make early in the code (such as excluding observations, transforming variables, etc.) that depend on the analysis (model checking) later. Think of it as a "choose your own adventure" that I've written for you.

1.1.1 Keep a record of your decisions

It is always desirable to make your work reproducible, either by someone else or by your future self. For each step you take, keep a diary of (a) what the next minor goal is, (b) what evidence/information you have, (c) what decision you make, and (d) what the outcome was.

For example, here's the first couple steps of your diary:

1. Include only "typical dwellings". Based on scatterplot, remove extreme observations. Keep only HOUSE and APARTMENT.
2. Exclude a few variables to reduce multicollinearity between predictor variables. Exclude `Baths` and `LotSize` .d
3. etc.

1.2 (2 p) (Step 1) Restrict data to "typical" dwellings

Step 1: After looking at the scatterplot below, identify what you consider to be a "typical dwelling" and exclude observations far from that range. For example, there are only a couple `TypeSale` that are common enough to model; remember to run `factor()` again to remove factor levels that no longer appear.

Hide

```
library(erikmisc)
library(tidyverse)
library(car)

# First, download the data to your computer,
#   save in the same folder as this Rmd file.

# read the data, skip the first two comment lines of the data file
dat_abq <-
  read_csv("~/Dropbox/3_Education/Courses/stat_528_ada2/ADA2_CL_14_HomePricesZi
        llow_Abq87108.csv", skip=2) %>%
  mutate(
    id = 1:n()
  , TypeSale = factor(TypeSale)
    # To help scale the intercept to a more reasonable value
    #   Scaling the x-variables are sometimes done to the mean of each x.
    # center year at 1900 (negative values are older, -10 is built in 1890)
  , YearBuilt_1900 = YearBuilt - 1900
  , logPriceList = log(PriceList, 10)
  , logSizeSqft = log(Size_sqft, 10)
  ) %>%
  select(
    id, everything()
    , -Address, -YearBuilt
  )

head(dat_abq)
```

```
# A tibble: 6 × 11
     id TypeSale  PriceList  Beds Baths Size_sqft LotSize DaysListed
  <int> <fct>         <dbl> <dbl> <dbl>     <dbl>   <dbl>      <dbl>
1     1 HOUSE        186900     3     2      1305    6969          0
2     2 APARTMENT    305000     1     1      2523    6098          0
3     3 APARTMENT    244000     1     1      2816    6098          0
4     4 CONDO        108000     3     2      1137      NA          0
5     5 CONDO         64900     2     1      1000      NA          1
6     6 HOUSE        275000     3     3      2022    6098          1
# … with 3 more variables: YearBuilt_1900 <dbl>, logPriceList <dbl>,
#   logSizeSqft <dbl>
```

Hide

```
## RETURN HERE TO SUBSET THE DATA

dat_abq <-
  dat_abq %>%
  filter(
    TypeSale %in% c("APARTMENT", "HOUSE"),
    PriceList < 6e5
  ) %>%
  mutate(across(TypeSale, ~factor(TypeSale))) %>%
  select(-c(Baths, LotSize))
# note, if you remove a level from a categorical variable, then run factor() ag
        ain

  # SOLUTION
  # these deletions are based only on the scatter plot in order to have
  #  "typical" dwellings

str(dat_abq)
```

```
tibble [129 × 9] (S3: tbl_df/tbl/data.frame)
 $ id            : int [1:129] 1 2 3 6 7 9 10 12 13 14 ...
 $ TypeSale      : Factor w/ 2 levels "APARTMENT","HOUSE": 2 1 1 2 2 2 2 1 2 2
...
 $ PriceList     : num [1:129] 186900 305000 244000 275000 133000 ...
 $ Beds          : num [1:129] 3 1 1 3 2 3 3 1 4 2 ...
 $ Size_sqft     : num [1:129] 1305 2523 2816 2022 1440 ...
 $ DaysListed    : num [1:129] 0 0 0 1 1 1 2 2 6 6 ...
 $ YearBuilt_1900: num [1:129] 54 48 89 52 52 58 52 49 41 53 ...
 $ logPriceList  : num [1:129] 5.27 5.48 5.39 5.44 5.12 ...
 $ logSizeSqft   : num [1:129] 3.12 3.4 3.45 3.31 3.16 ...
```

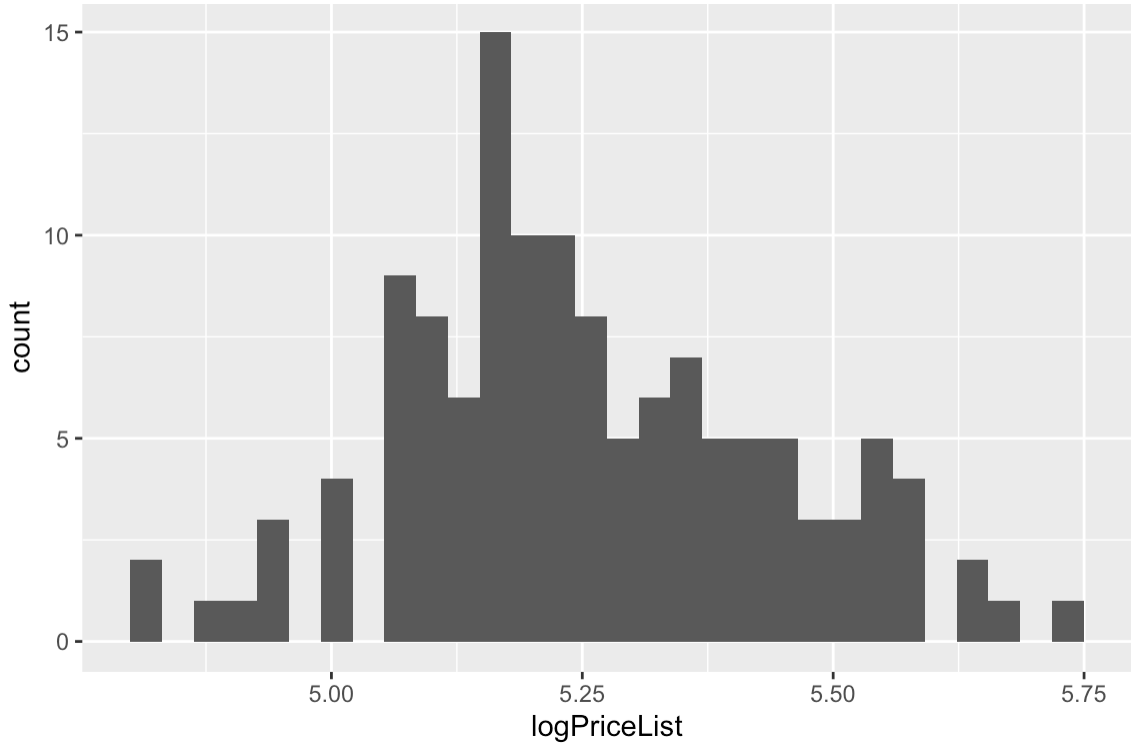
Hide

```
table(dat_abq$TypeSale)
```

```

APARTMENT     HOUSE
       40        89
```

Hide

```
dat_abq %>%
  ggplot() +
  geom_histogram(aes(x = logPriceList))
```



1.3 (2 p) (Step 3) Transform response, if necessary.

Step 3: Does the response variable require a transformation? If so, what transformation is recommended from the model diagnostic plots (Box-Cox)?

1.3.1 Solution

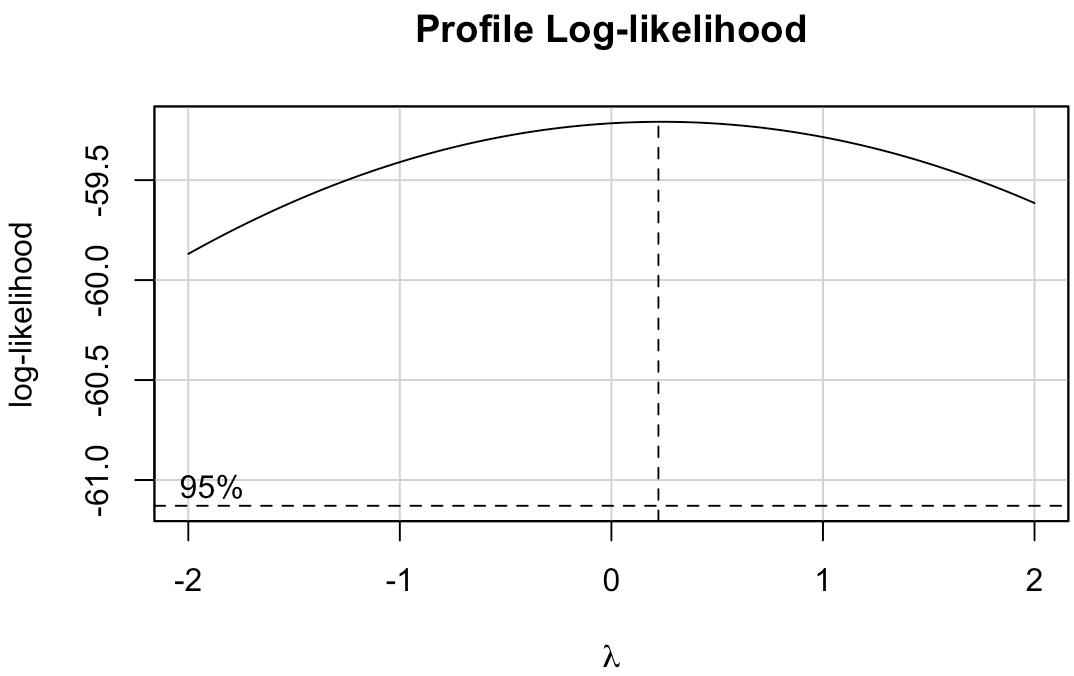
Hide

```
dat_abq <-
  dat_abq %>%
  mutate(
    # Price in units of $1000
    PriceListK = PriceList / 1000

    # SOLUTION
  ) %>%
  select(
    -PriceList
  )

str(dat_abq)
```

```
tibble [129 × 9] (S3: tbl_df/tbl/data.frame)
 $ id            : int [1:129] 1 2 3 6 7 9 10 12 13 14 ...
 $ TypeSale      : Factor w/ 2 levels "APARTMENT","HOUSE": 2 1 1 2 2 2 2 1 2 2
...
 $ Beds          : num [1:129] 3 1 1 3 2 3 3 1 4 2 ...
 $ Size_sqft     : num [1:129] 1305 2523 2816 2022 1440 ...
 $ DaysListed    : num [1:129] 0 0 0 1 1 1 2 2 6 6 ...
 $ YearBuilt_1900: num [1:129] 54 48 89 52 52 58 52 49 41 53 ...
 $ logPriceList  : num [1:129] 5.27 5.48 5.39 5.44 5.12 ...
 $ logSizeSqft   : num [1:129] 3.12 3.4 3.45 3.31 3.16 ...
 $ PriceListK    : num [1:129] 187 305 244 275 133 ...
```

Hide

```
mod1 <- lm(logPriceList ~ Beds + Size_sqft + DaysListed + YearBuilt_1900,
           data = dat_abq)
boxCox(mod1)
```



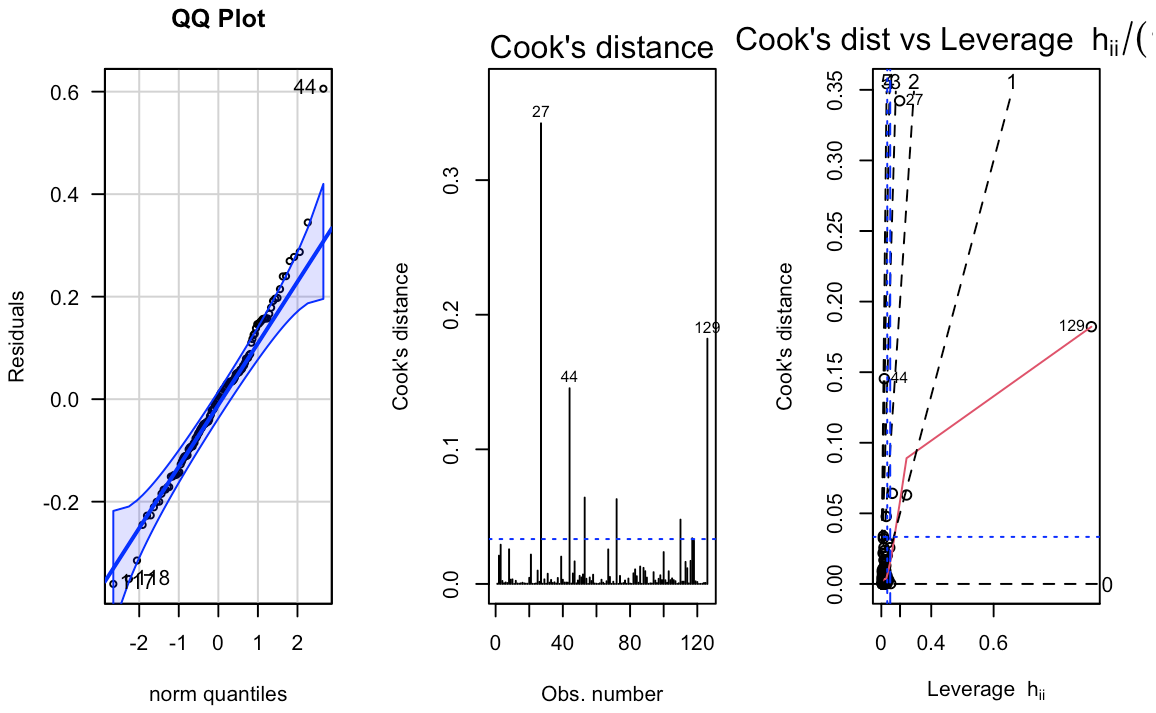
The log transformation is clearly supported by the Box-Cox profile.

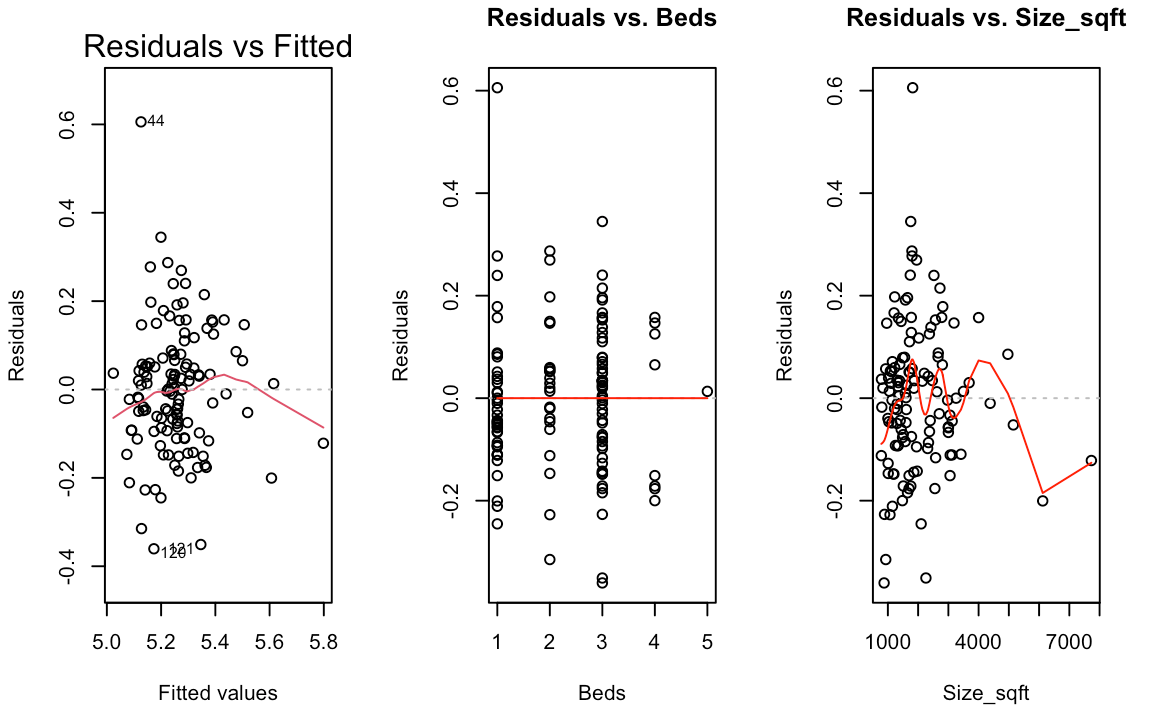
1.4 (2 p) (Step 4) Remove extremely influential observations.

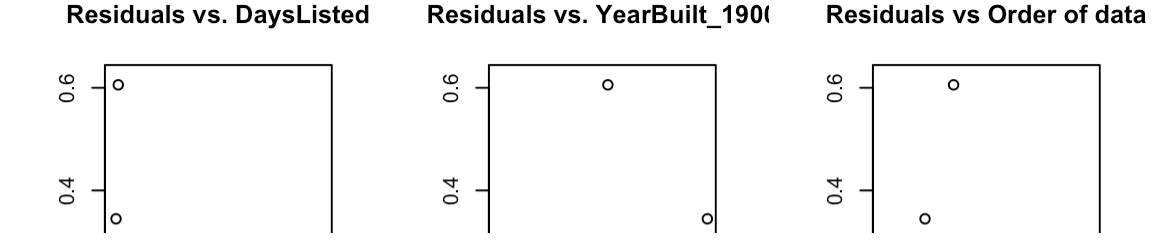
Step 4: The goal is to develop a model that will work well for the typical dwellings. If an observation is highly influential, then it's unusual.

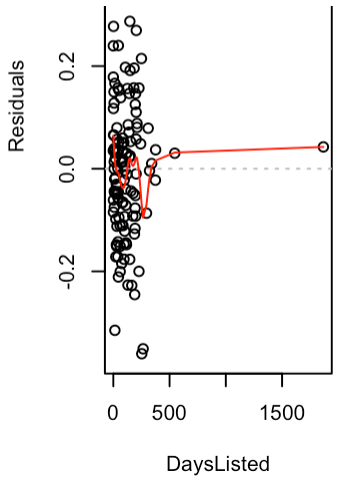
```
e_plot_lm_diagostics(mod1)
```

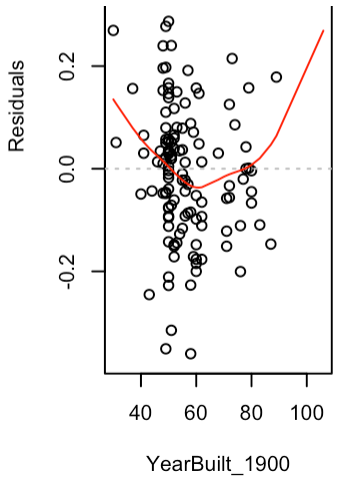
Hide

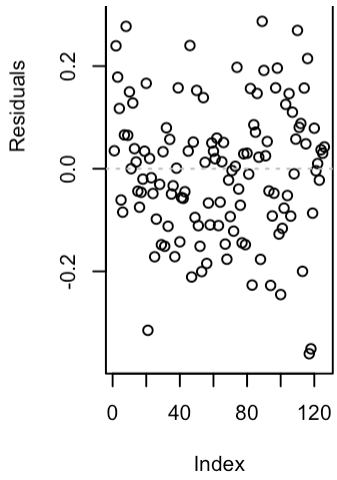




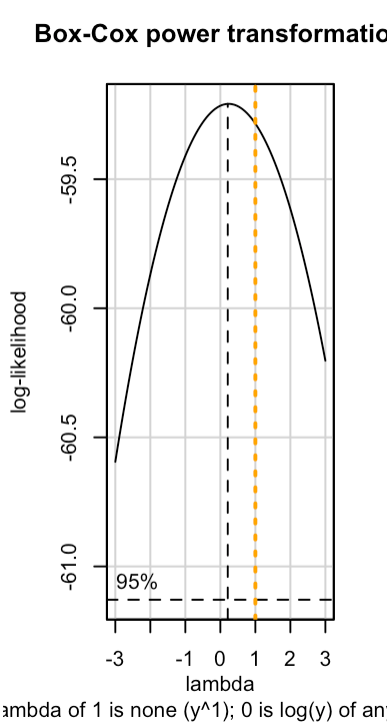


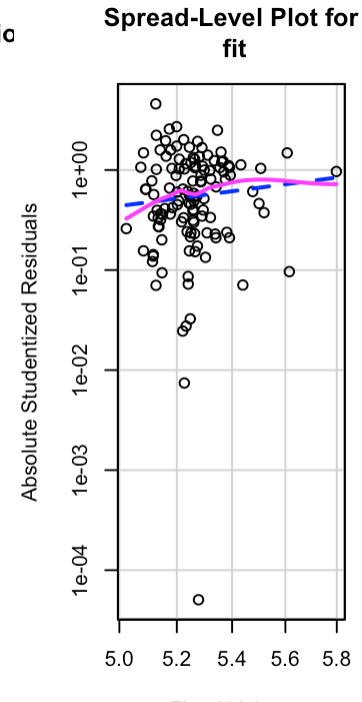


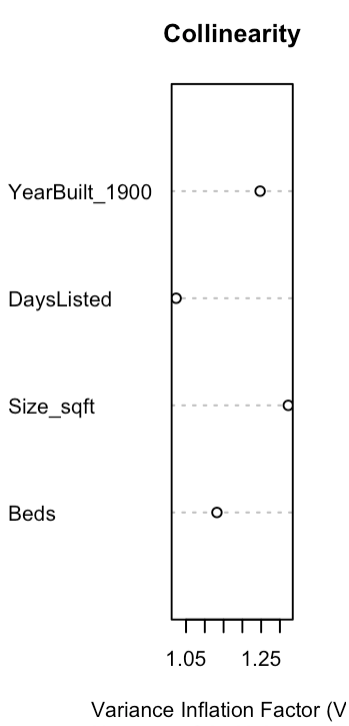


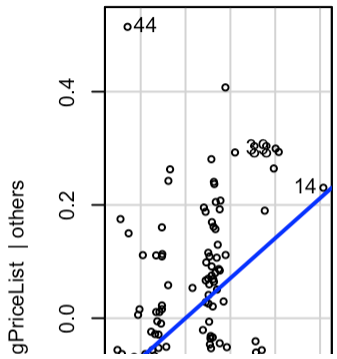


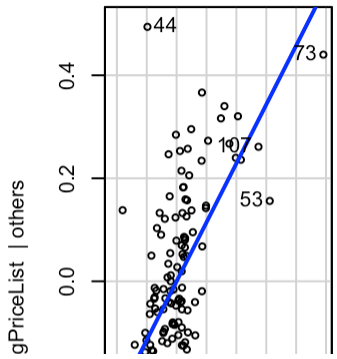

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.593903, Df = 1, p = 0.20677
```

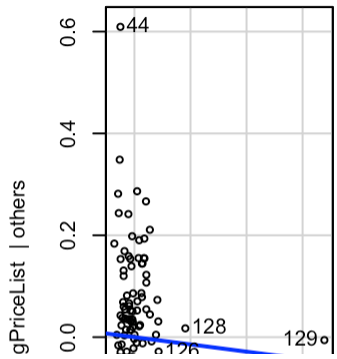


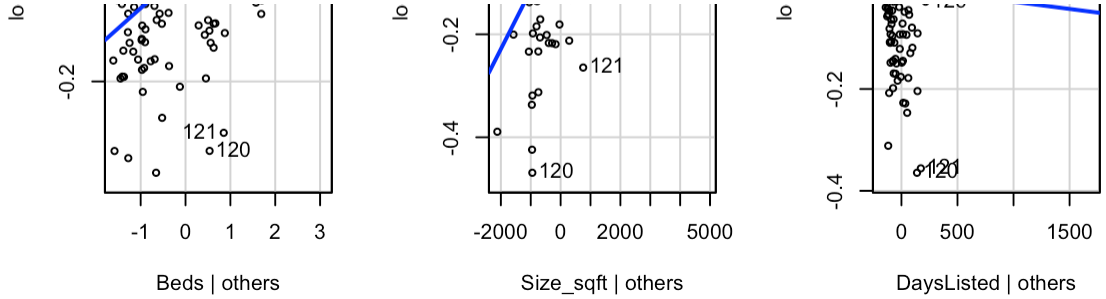


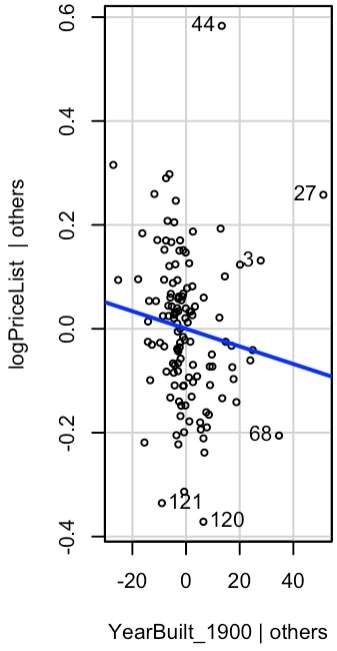








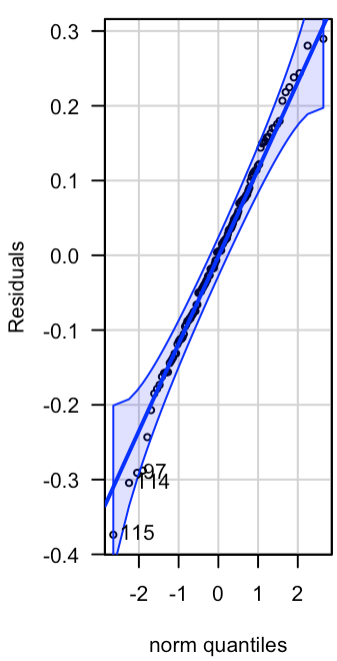


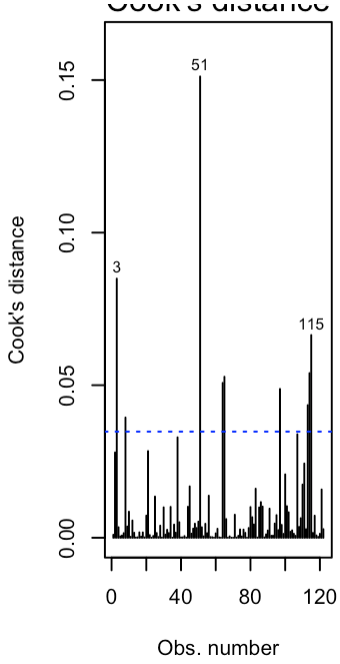


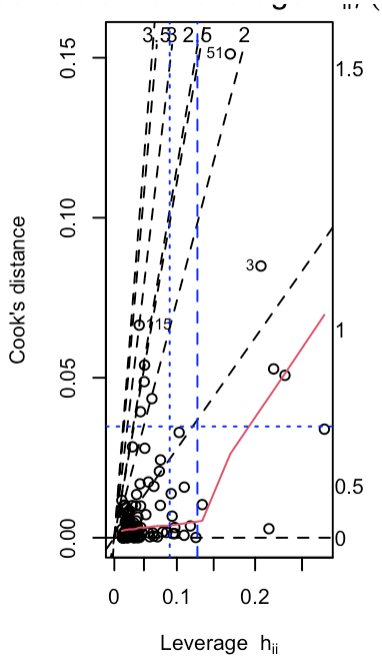
Hide

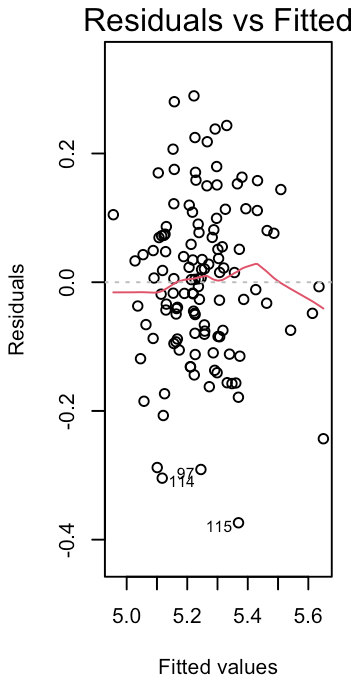
```
## Remove influential observation
  dat_abq <-
    dat_abq %>%
    filter(
      YearBuilt_1900 < 100,
      PriceListK < 539,
      DaysListed < 1000,
      Size_sqft < 7000
    )

mod1 <- lm(logPriceList ~ Beds + Size_sqft + DaysListed + YearBuilt_1900 + I(Ye
        arBuilt_1900^2),
           data = dat_abq)

e_plot_lm_diagostics(mod1)
```

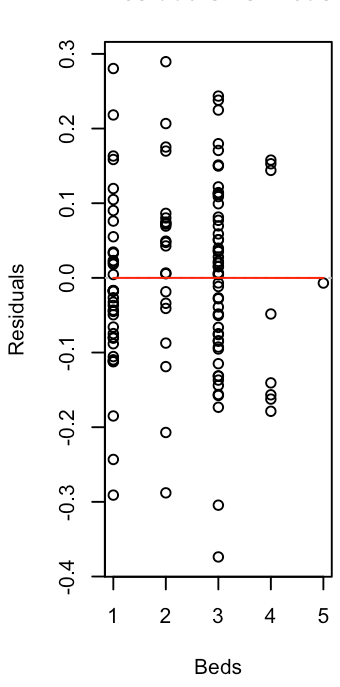
QQ Plot Cook's distance Cook's dist vs Leverage $h_{ii}/($

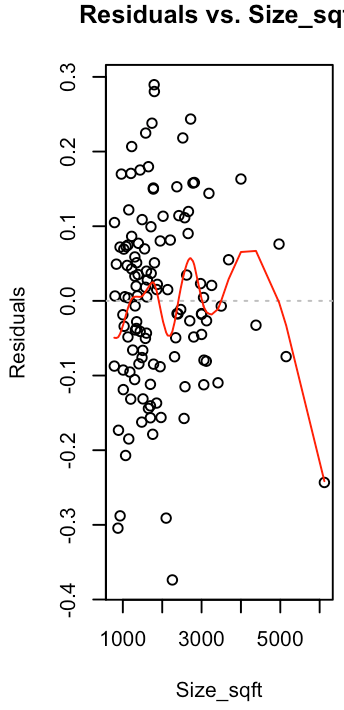

Residuals
norm quantiles
97
114
115

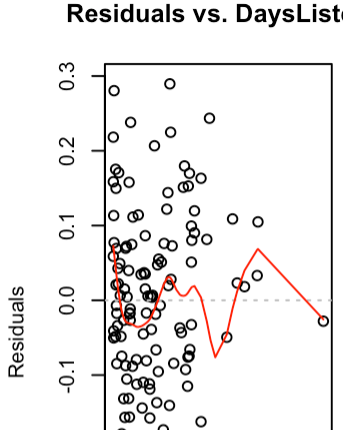

Cook's distance
Obs. number
51
3
115

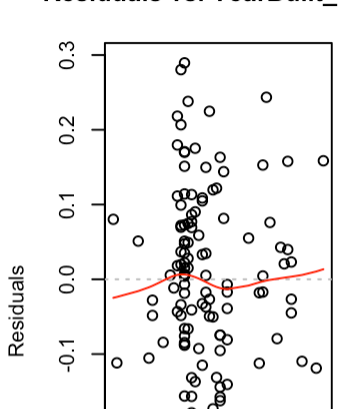

Cook's distance
Leverage h_ii
51
30
115

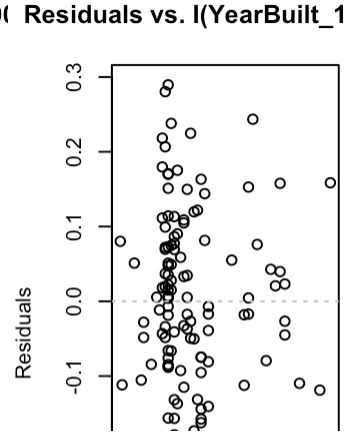

Residuals vs Fitted
Residuals
Fitted values
97
114
115

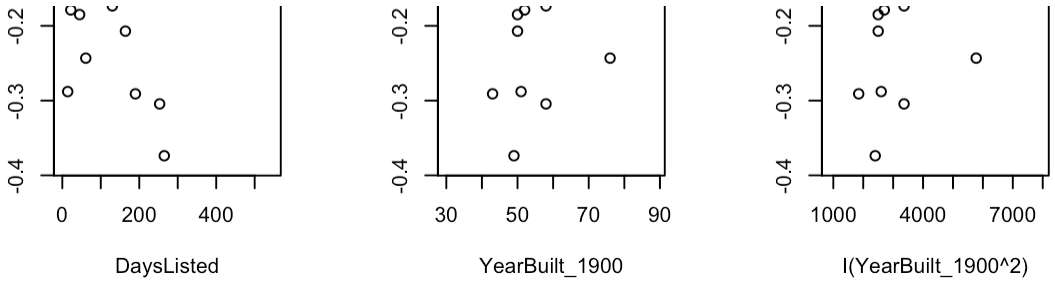

Residuals vs. Beds
Residuals
Beds


Residuals vs. Size_sqft
Residuals
Size_sqft

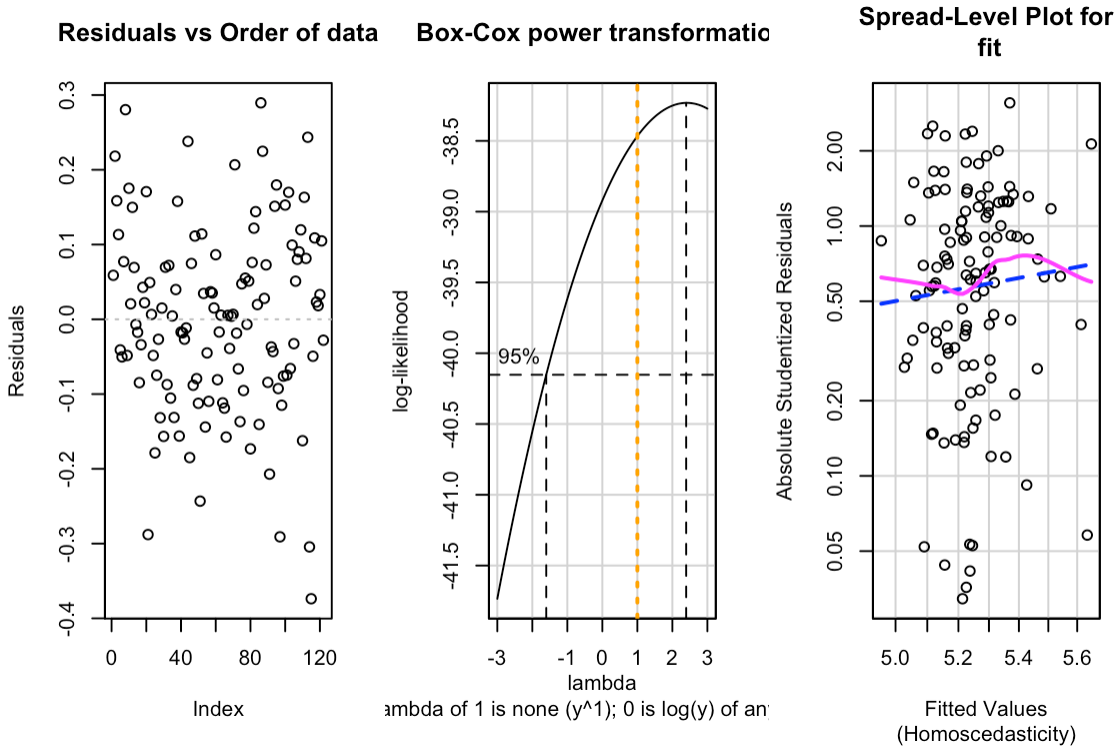

Residuals vs. DaysListed
Residuals

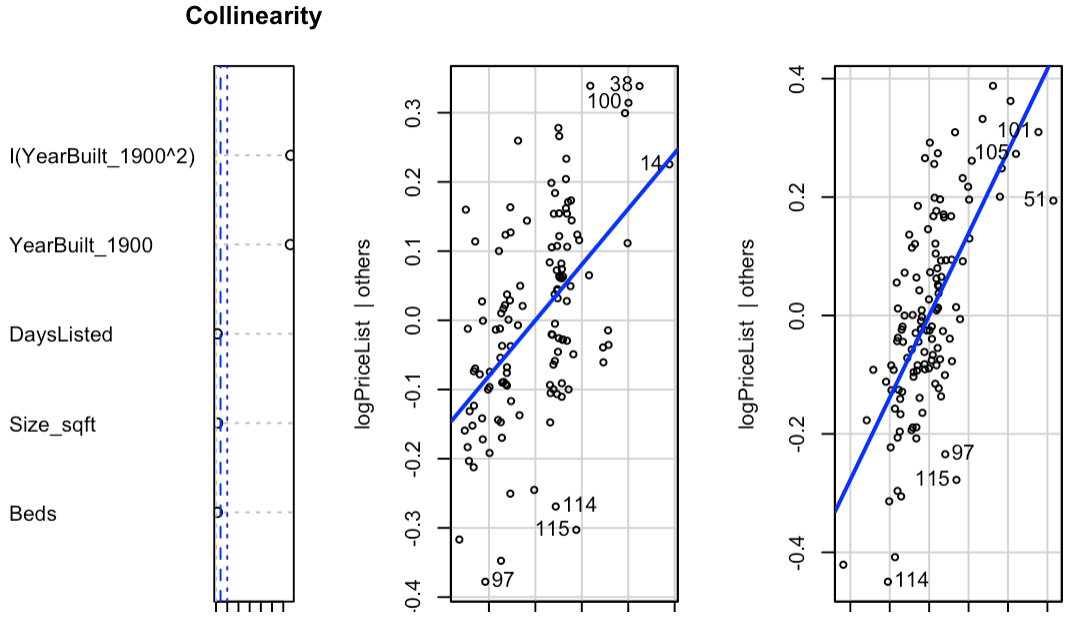

Residuals vs. YearBuilt_190(
Residuals

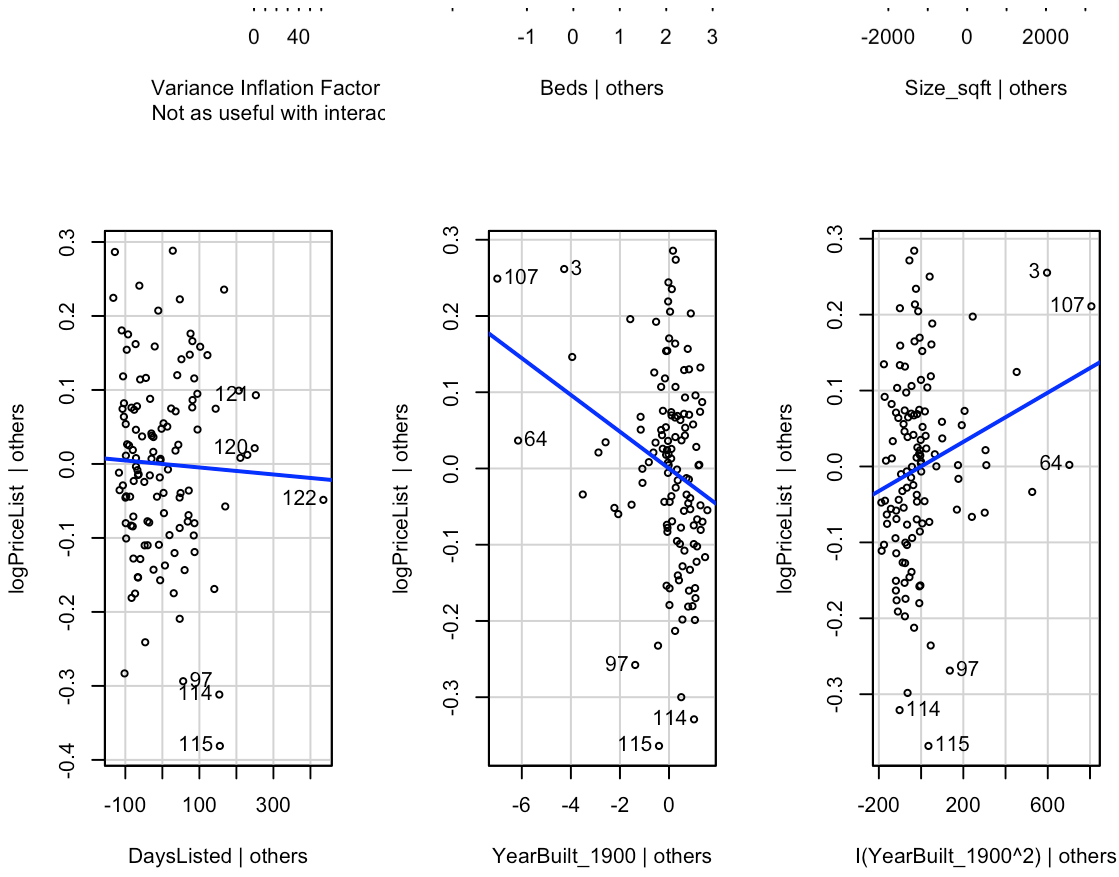

Residuals vs. I(YearBuilt_1900
Residuals




```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.2940893, Df = 1, p = 0.58761
```







This looks pretty good. Adding a quadratic term for year built seems to work. The 70s were really bad. Just awful.

1.5 Subset data for model building and prediction

Create a subset of the data for building the model, and another subset for prediction later on.

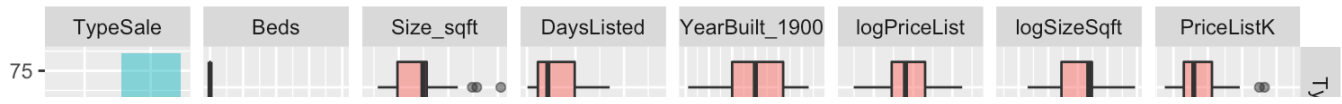
Hide

```
# remove observations with NAs
dat_abq <-
  dat_abq %>%
  na.omit()

# the data subset we will use to build our model
dat_sub <-
  dat_abq %>%
  filter(
    DaysListed > 0
  )

# the data subset we will predict from our model
dat_pred <-
  dat_abq %>%
  filter(
    DaysListed == 0
  ) %>%
  mutate(
    # the prices we hope to predict closely from our model
    PriceListK_true = logPriceList
    # set them to NA to predict them later
  , PriceListK = NA
  )
```

Scatterplot of the model-building subset.

Hide

```
# NOTE, this plot takes a long time if you're repeadly recompiling the documen
        t.
# comment the "print(p)" line so save some time when you're not evaluating this
        plot.
library(GGally)
library(ggplot2)
p <-
  ggpairs(
    dat_sub %>% select(-id)
  , mapping = ggplot2::aes(colour = TypeSale, alpha = 0.5)
  , lower = list(continuous = "points")
  , upper = list(continuous = "cor")
  , progress = FALSE
  )
print(p)
```



All the apartments appear to have only one bed, yet they are larger? Something fishy there.

For both number of beds (houses only) and square footage, we see clear positive relationships with listing price.

There may be an outlier in the number of days listed … but it wasn't apparent in the diagnostics, so I'm going to roll with it.

The log-transformed data look more normally distributed than the raw data.

1.5.1 Solution

[answer]

Features of data:

1.6 (2 p) (Step 2) Fit full two-way interaction model.

You'll revisit this section after each modification of the data above.

Step 2: Let's fit the full two-way interaction model and assess the assumptions. However, some of the predictor variables are highly correlated. Recall that the interpretation of a beta coefficient is "the expected increase in the response for a 1-unit increase in \(x\) with all other predictors held constant". It's hard to hold one variable constant if it's correlated with another variable you're increasing. Therefore, we'll make a decision to retain some variables but not others depending on their correlation values. (In the PCA chapter, we'll see another strategy.)

Somewhat arbitrarily, let's exclude `Baths` (since highly correlated with `Beds` and `Size_sqft`). Let's also exclude `LotSize` (since highly correlated with `Size_sqft`). Modify the code below. Notice that because APARTMENTs don't have more than 1 Beds or Baths, those interaction terms need to be excluded from the model; I show you how to do this manually using the `update()` function.

Note that the formula below `y ~ (x1 + x2 + x3)^2` expands into all main effects and two-way interactions.

Hide

```
  ## SOLUTION
dat_sub <- dat_sub %>%
  filter(! id %in% c(59, 74, 75))

  lm_full <-
    lm(
      logPriceList ~ (TypeSale + Beds + logSizeSqft + DaysListed + YearBuilt_19
        00)^2
    , data = dat_sub
    )
  #lm_full <-
  #  lm(
  #    PriceListK ~ (Beds + Baths + Size_sqft + LotSize + DaysListed + YearBuil
        t_1900)^2
  #  , data = dat_sub
  #  )
  lm_full
```

```
Call:
lm(formula = logPriceList ~ (TypeSale + Beds + logSizeSqft + 
    DaysListed + YearBuilt_1900)^2, data = dat_sub)

Coefficients:
                  (Intercept)                  TypeSaleHOUSE  
                    6.296e+00                      9.084e-01  
                         Beds                    logSizeSqft  
                   -3.532e-01                     -1.804e-01  
                   DaysListed                 YearBuilt_1900  
                    1.681e-04                     -7.652e-02  
          TypeSaleHOUSE:Beds     TypeSaleHOUSE:logSizeSqft  
                           NA                     -2.058e-01  
    TypeSaleHOUSE:DaysListed  TypeSaleHOUSE:YearBuilt_1900  
                   -1.700e-04                      2.454e-04  
            Beds:logSizeSqft               Beds:DaysListed  
                    7.051e-02                      4.533e-05  
         Beds:YearBuilt_1900        logSizeSqft:DaysListed  
                    2.121e-03                     -1.814e-04  
  logSizeSqft:YearBuilt_1900     DaysListed:YearBuilt_1900  
                    1.996e-02                      7.739e-06  
```

Hide

```
  library(car)
  try(Anova(lm_full, type=3))
```

```
Error in Anova.III.lm(mod, error, singular.ok = singular.ok, ...) : 
  there are aliased coefficients in the model
```

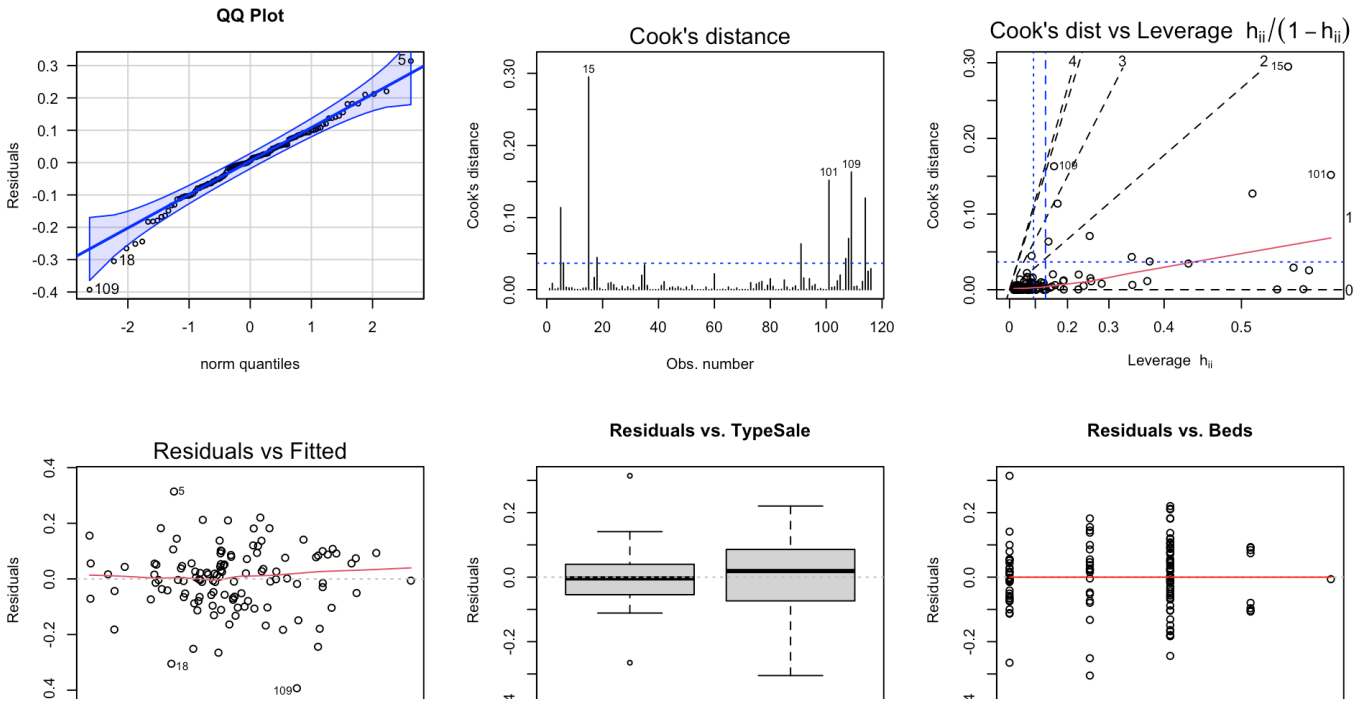
Hide

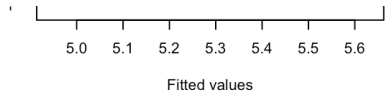
```
  ## Note that this doesn't work because APARTMENTs only have 1 bed and 1 bath.
  ## There isn't a second level of bed or bath to estimate the interaction.
  ## Therefore, remove those two terms
  lm_full <-
    update(
      lm_full
    , . ~ . - TypeSale:Beds
    )
  library(car)
  try(Anova(lm_full, type=3))
```

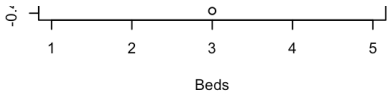
```
Anova Table (Type III tests)

Response: logPriceList
                            Sum Sq  Df F value   Pr(>F)
(Intercept)                0.12448   1  8.6120 0.004134 **
TypeSale                   0.01167   1  0.8077 0.370942
Beds                       0.01060   1  0.7330 0.393924
logSizeSqft                0.00107   1  0.0742 0.785847
DaysListed                 0.00004   1  0.0025 0.960456
YearBuilt_1900             0.05185   1  3.5869 0.061099 .
TypeSale:logSizeSqft       0.00652   1  0.4512 0.503320
TypeSale:DaysListed        0.00137   1  0.0946 0.759030
TypeSale:YearBuilt_1900    0.00002   1  0.0011 0.973877
Beds:logSizeSqft           0.00438   1  0.3031 0.583165
Beds:DaysListed            0.00064   1  0.0442 0.833968
Beds:YearBuilt_1900        0.00831   1  0.5752 0.449982
logSizeSqft:DaysListed     0.00037   1  0.0257 0.872840
logSizeSqft:YearBuilt_1900 0.03996   1  2.7644 0.099483 .
DaysListed:YearBuilt_1900  0.00475   1  0.3287 0.567706
Residuals                  1.45986 101
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

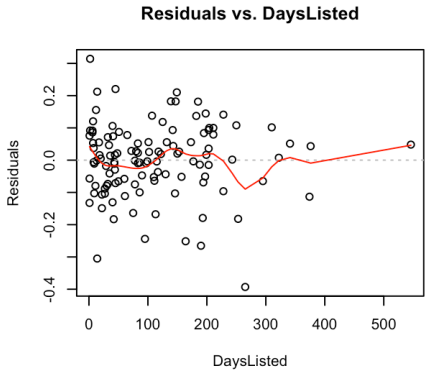
Hide

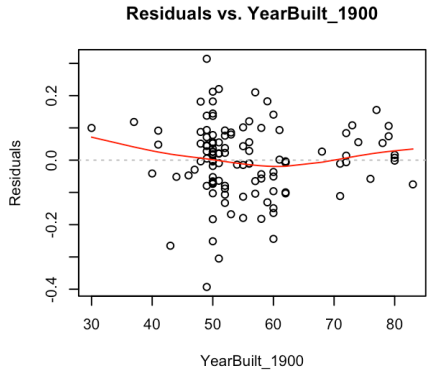
```
## Uncomment this line when you're ready to assess the model assumptions
# plot diagnostics
e_plot_lm_diagostics(lm_full)
```



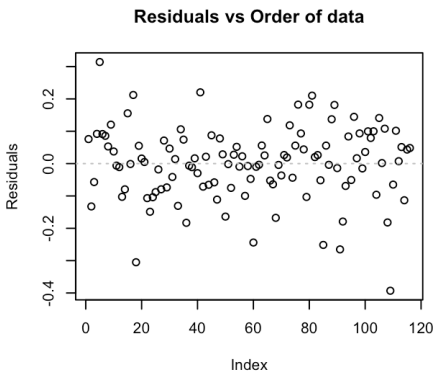


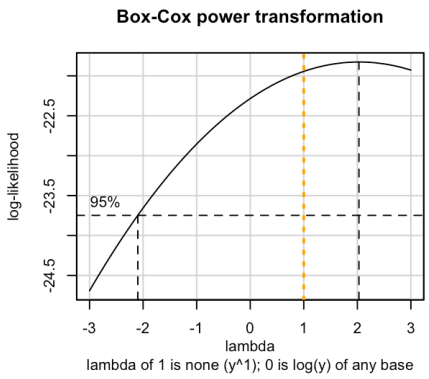


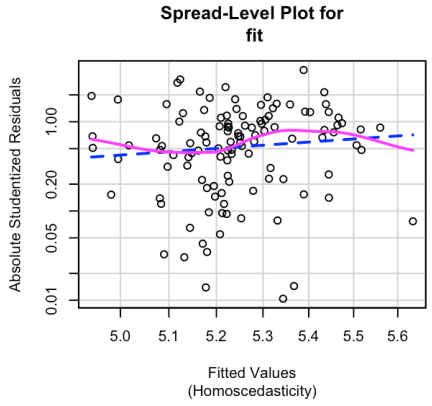


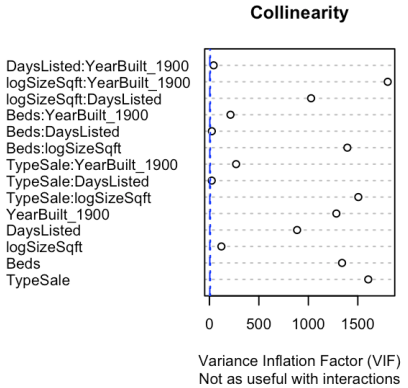


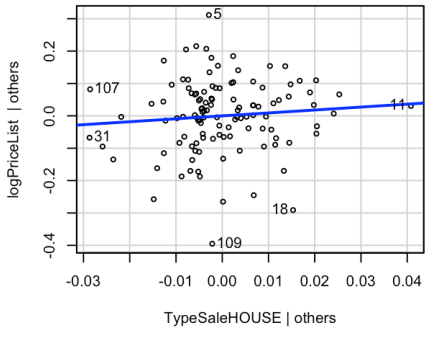

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.3603047, Df = 1, p = 0.54834
```

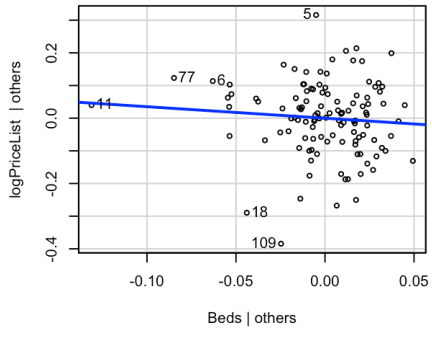


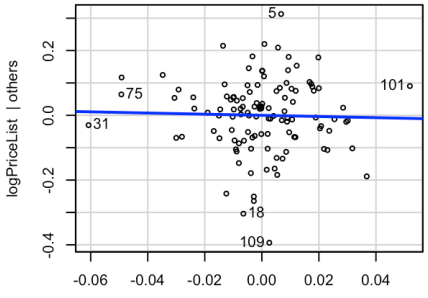


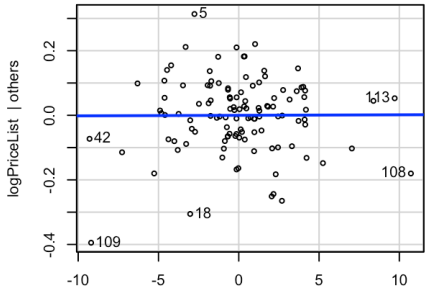


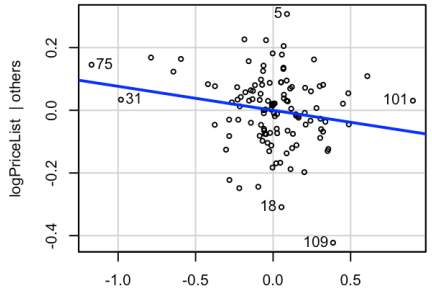


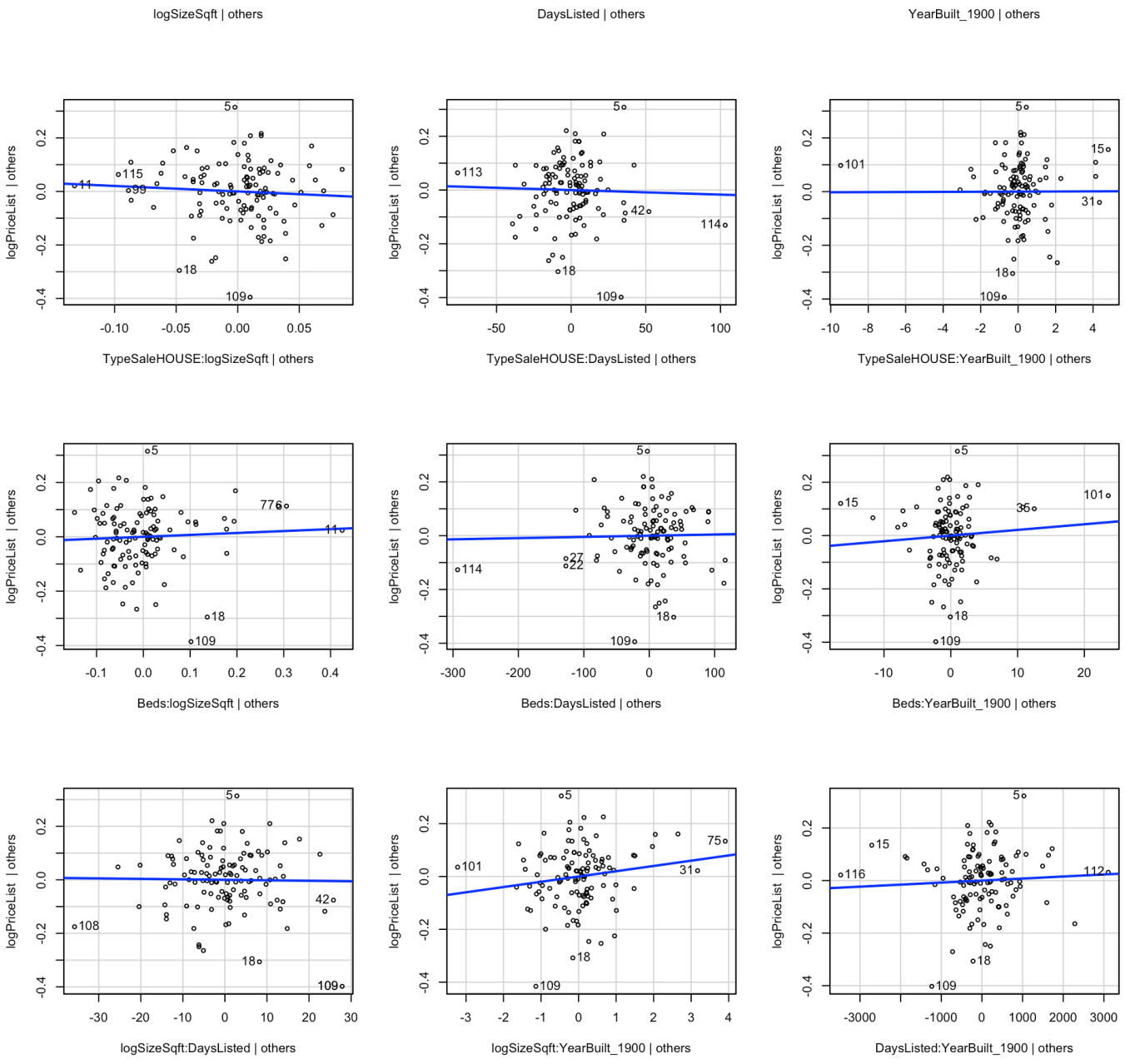












Hide

```
# List the row numbers with id numbers
#   The row numbers appear in the residual plots.
#   The id number can be used to exclude values in code above.
dat_sub %>% select(id) %>% print(n = Inf)
```

```
# A tibble: 116 × 1
       id
    <int>
  1     6
  2     7
  3     9
  4    10
  5    12
  6    13
  7    14
  8    15
  9    16
 10    17
 11    19
 12    20
 13    21
 14    22
 15    23
 16    24
 17    25
 18    26
 19    27
 20    28
 21    29
 22    30
 23    31
 24    33
 25    34
 26    35
 27    36
 28    38
 29    39
 30    40
 31    41
 32    42
 33    43
 34    44
 35    45
 36    46
 37    47
 38    48
 39    49
 40    51
 41    52
 42    53
 43    54
 44    55
```

```
45    56
46    57
47    58
48    60
49    61
50    62
51    64
52    65
53    67
54    68
55    69
56    70
57    71
58    72
59    73
60    76
61    77
62    78
63    79
64    81
65    83
66    84
67    85
68    86
69    87
70    88
71    89
72    91
73    92
74    94
75    95
76    97
77    98
78    99
79   100
80   101
81   102
82   103
83   104
84   105
85   106
86   108
87   109
88   110
89   111
90   112
91   113
92   114
```

```
 93   115
 94   116
 95   117
 96   118
 97   119
 98   121
 99   122
100   123
101   124
102   126
103   127
104   128
105   129
106   131
107   132
108   133
109   134
110   135
111   136
112   137
113   138
114   140
115   141
116   142
```

After Step 2, interpret the residual plots. What are the primary issues in the original model?

See discussion above. There were a lot of outliers influencing the data, and the response needed a transformation. AFter removing outliers, the data don't seem all that normally distributed anymore. They're pretty close though, so we'll just keep moving along.

1.6.1 Solution

[answer]

1.7 (2 p) (Step 5) Model selection, check model assumptions.

Using `step(..., direction="both")` with the BIC criterion, perform model selection.

1.7.1 Solution

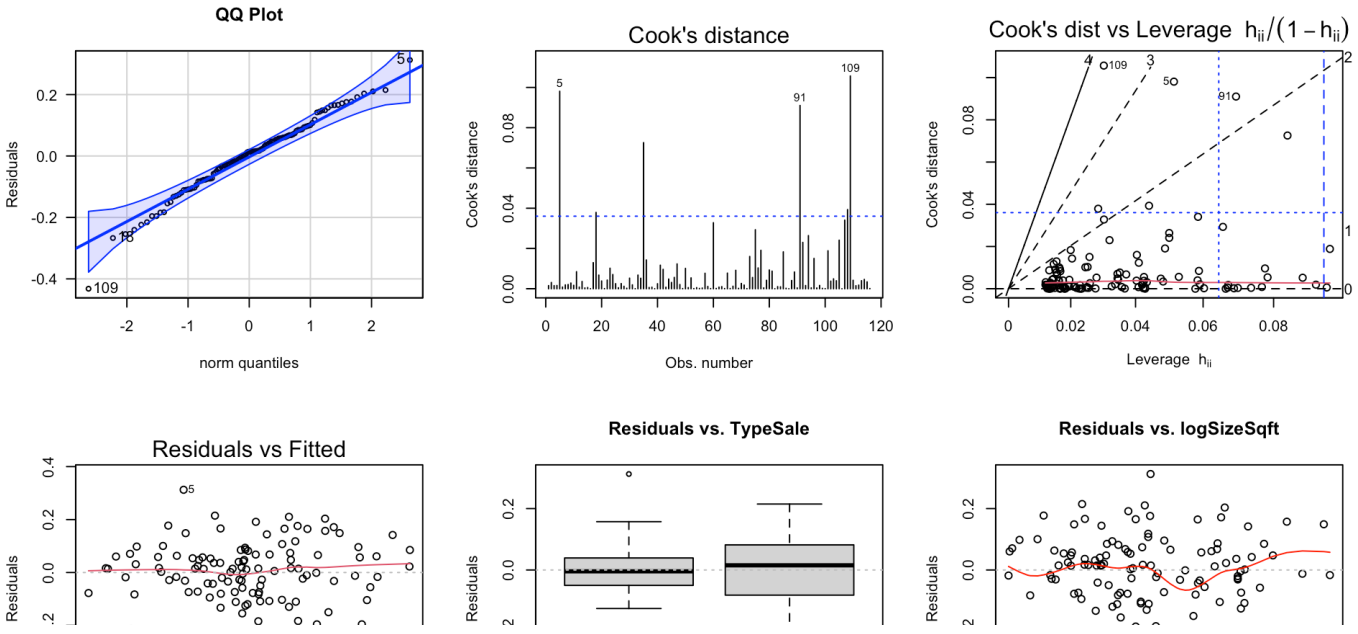
Hide

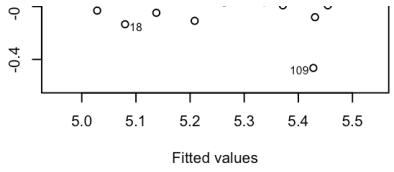
```
## BIC
# option: test="F" includes additional information
#           for parameter estimate tests that we're familiar with
# option: for BIC, include k=log(nrow( [data.frame name] ))
lm_red_BIC <-
  step(
    lm_full
  , direction = "both"
  , test = "F"
  , trace = 0
  , k = log(nrow(dat_sub))
  )
lm_final <- lm_red_BIC
lm_red_BIC
```

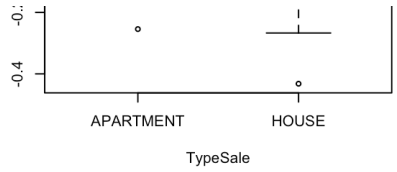
```

Call:
lm(formula = logPriceList ~ TypeSale + logSizeSqft + YearBuilt_1900,
    data = dat_sub)

Coefficients:
   (Intercept)   TypeSaleHOUSE     logSizeSqft  YearBuilt_1900
      2.451636        0.215648        0.881960       -0.004028
```

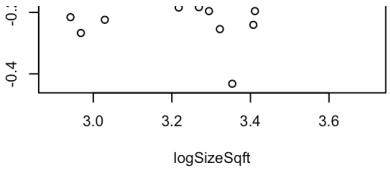
Hide

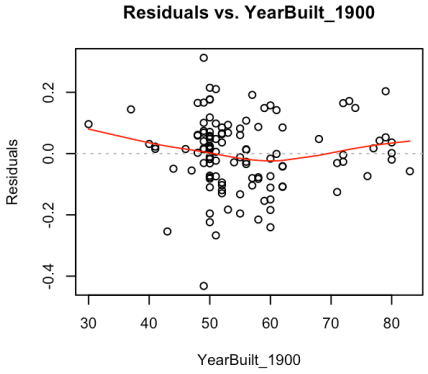
```
## Uncomment this line when you're ready to assess the model assumptions
# plot diagnostics
e_plot_lm_diagostics(lm_final)
```

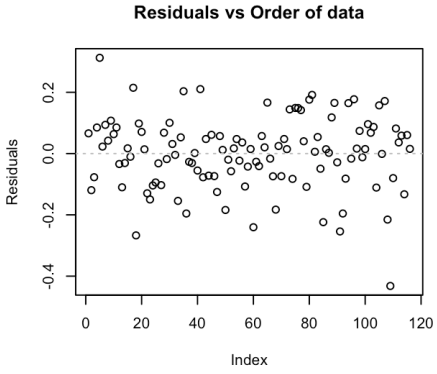


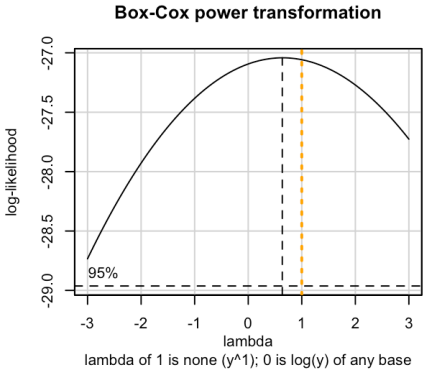




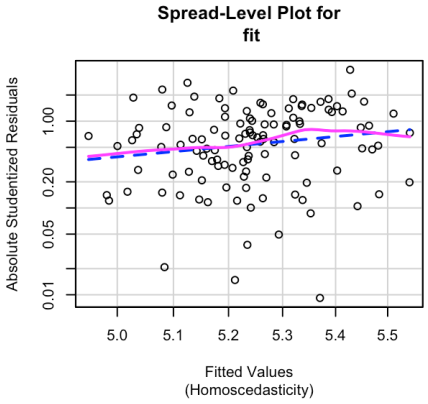


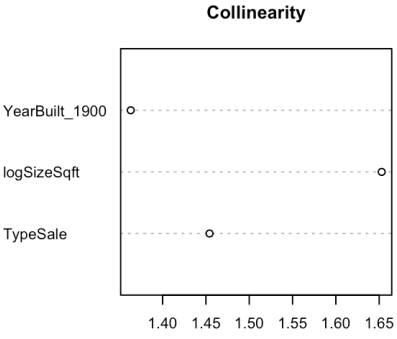


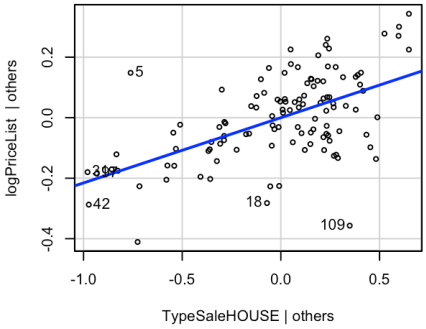


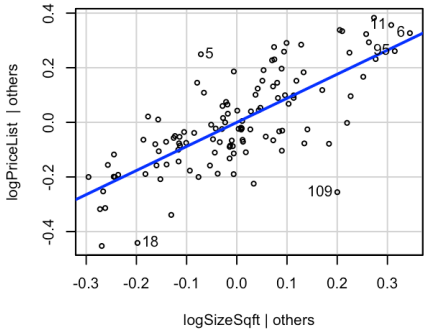


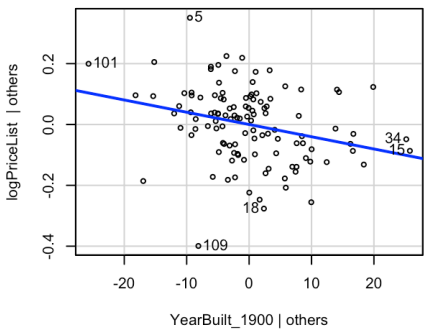

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 3.055127, Df = 1, p = 0.080483
```







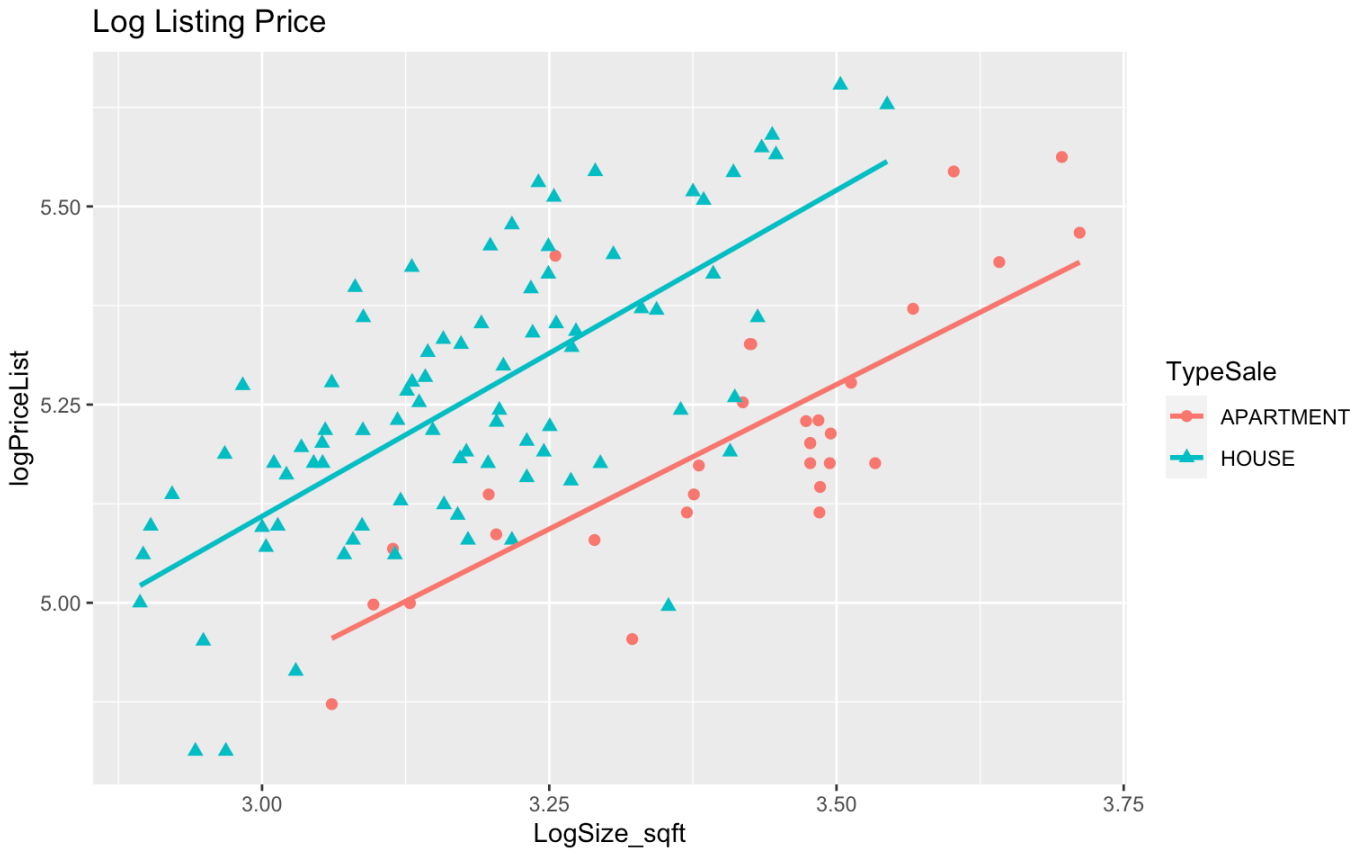




So, the model that included a quadratic term for year built was selected by stepwise regression, but it resulted in a dubious, complex model. I simplified things by removing the quadratic, leading to an additive model with good diagnostics. There do not appear to be outliers, the residuals are (mostly) normally distributed, and variance is well stabilized.

1.8 (4 p) (Step 6) Plot final model, interpret coefficients.

If you arrived at the same model I did, then the code below will plot it. Eventually (after Step 7), the fitted model equations will describe the each dwelling `TypeSale` and interpret the coefficients.



Hide

```
library(car)
#Anova(lm_final, type=3)
summary(lm_final)
```

```
Call:
lm(formula = logPriceList ~ TypeSale + logSizeSqft + YearBuilt_1900,
    data = dat_sub)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43211 -0.07393  0.01398  0.06793  0.31249

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.451636   0.244447  10.029  < 2e-16 ***
TypeSaleHOUSE  0.215648   0.030189   7.143 9.78e-11 ***
logSizeSqft    0.881960   0.076863  11.474  < 2e-16 ***
YearBuilt_1900 -0.004028   0.001307  -3.082  0.00259 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1193 on 112 degrees of freedom
Multiple R-squared:  0.5507,    Adjusted R-squared:  0.5386
F-statistic: 45.76 on 3 and 112 DF,  p-value: < 2.2e-16
```

Fitted model equation is $$ = 2.45 + 0.216 (TypeSale = House) + 0.882 LogSqFt - .004 YearBuilt

$$

1.8.1 Solution

After Step 7, return and intepret the model coefficients above.

The intercept isn't worth interpreting, since it's relevant when Size (sqft) is 0, a ridiculous and terrifying idea

The significant effect of TypeSale indicates that houses list for higher prices than apartment. The significant effect of logSizeSqft indicates a positive relationship between listing price and size for both apartments and houses. Lastly, the significant negative relationship with YearBuilt indicates older houses list for higher prices.

1.9 (2 p) (Step 7) Transform predictors.

We now have enough information to see that a transformation of a predictor can be useful. See the curvature with `Size_sqft` ? This is one of the headaches of regression modelling, *everything depends on everything else* and you learn as you go. Return to the top and transform `Size_sqft` and `LotSize` .

A nice feature of this transformation is that the model interaction goes away. Our interpretation is now on the log scale, but it's a simpler model.

I don't know how to get points here. I did it.

1.10 (4 p) (Step 8) Predict new observations, interpret model's predictive ability.

Using the `predict()` function, we'll input the data we held out to predict earlier, and use our final model to predict the `PriceListK` response. Note that `10^lm_pred` is the table of values on the scale of "thousands of dollars".

Interpret the predictions below the output.

How well do you expect this model to predict? Justify your answer.

Hide

```
# predict new observations, convert to data frame
lm_pred <-
  as.data.frame(
    predict(
      lm_final
    , newdata = dat_pred
    , interval = "prediction"
    )
  ) %>%
  mutate(
    # add column of actual list prices
    PriceListK = dat_pred$PriceListK_true
  )
lm_pred
```

```
       fit      lwr      upr PriceListK
1 5.197594 4.959557 5.435631   5.271609
2 5.258627 5.015824 5.501430   5.484300
3 5.135546 4.885871 5.385221   5.387390
```

Hide

```
# on "thousands of dollars" scale
10^lm_pred
```

```
       fit       lwr      upr PriceListK
1 157613.8  91108.19 272665.9     186900
2 181395.8 103710.81 317271.1     305000
3 136630.1  76890.25 242784.8     244000
```

Hide

```
# attributes of the three predicted observations
dat_pred %>% print(n = Inf, width = Inf)
```

```
# A tibble: 3 × 10
     id TypeSale   Beds Size_sqft DaysListed YearBuilt_1900 logPriceList
  <int> <fct>     <dbl>     <dbl>      <dbl>          <dbl>        <dbl>
1     1 HOUSE         3      1305          0             54         5.27
2     2 APARTMENT     1      2523          0             48         5.48
3     3 APARTMENT     1      2816          0             89         5.39
  logSizeSqft PriceListK PriceListK_true
        <dbl> <lgl>                <dbl>
1        3.12 NA                    5.27
2        3.40 NA                    5.48
3        3.45 NA                    5.39
```

1.10.1 Solution

It didn't do so great, but we don't have a benchmark in any case. Personally, I think we ought not to have chosen the 0 days listed properties, which typically will list for higher than older properties. Indeed, we see that we are under-predicting the property listing values here. A better approach would be to randomly sample a handfull (20%?) of properties for prediction, and use DaysListed as a feature in the prediction process.