PSET 03 - Linear Regression

S&DS 361

Part 1: Property values

The following data set contains information about properties in Branford, CT. Each row is a property, and the columns contain information about that property.

```r
b = read.csv('data/branford.csv')

## let's get rid of Mobile Homes
## and keep only the columns we will be working with
b = b %>%
  filter(style!='Mobile Home') %>%
  select(value, living, beds, baths, halfbaths, miles_to_coastline)
head(b,2)
```

```
##    value living beds baths halfbaths miles_to_coastline
## 1 247400   2194    3     2         1          0.6500547
## 2 177200   1200    3     1         1          0.4848638
```

The column `value` indicates the assessed value of that property. The column `living` indicates the square feet of the living area of the property, `beds` is the number of bedrooms, `baths` is the number of full bathrooms, and `halfbaths` is the number of half bathrooms.

1. Fit a linear regression model using `log(value)` as the outcome and `living` as a predictor, and another model with `log(living)` as a predictor. Which model do you think is better? Why?

```r
lm1 = lm(log(value) ~ living, data=b)
lm2 = lm(log(value) ~ log(living), data=b)
summary(lm1)
```

```
##
## Call:
## lm(formula = log(value) ~ living, data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.97639 -0.19853 -0.07320  0.09121  2.40703
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.167e+01  1.184e-02  985.84   <2e-16 ***
## living      3.955e-04  5.195e-06   76.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3417 on 4655 degrees of freedom
## Multiple R-squared:  0.5546, Adjusted R-squared:  0.5545
## F-statistic:  5795 on 1 and 4655 DF,  p-value: < 2.2e-16
```

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = log(value) ~ log(living), data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.17401 -0.22436 -0.09729  0.10060  2.72068
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.18045    0.08845   69.88   <2e-16 ***
## log(living)  0.83649    0.01171   71.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3537 on 4655 degrees of freedom
## Multiple R-squared:  0.5228, Adjusted R-squared:  0.5227
## F-statistic:  5099 on 1 and 4655 DF,  p-value: < 2.2e-16
```

From the above results, the first model is better because it has a higher R-squared value (0.55) compared to the second model (0.52). It is rather hard to say the first model is definitely better than the second because the R-squared values are so close.

2. Try adding `beds`, `baths`, and `halfbaths` as predictors to the model above. Do those predictors improve the model? Why or why not?

```r
lm3 = lm(log(value) ~ living + beds + baths + halfbaths, data=b)
summary(lm3)
```

```
##
## Call:
## lm(formula = log(value) ~ living + beds + baths + halfbaths,
##     data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.59874 -0.19289 -0.07465  0.08869  2.43700
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.158e+01  1.804e-02 642.003  < 2e-16 ***
## living      3.187e-04  8.912e-06  35.758  < 2e-16 ***
## beds        5.617e-03  6.495e-03   0.865    0.387
## baths       9.435e-02  9.496e-03   9.936  < 2e-16 ***
## halfbaths   7.294e-02  1.008e-02   7.240 5.23e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3369 on 4647 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.5673, Adjusted R-squared:  0.5669
## F-statistic:  1523 on 4 and 4647 DF,  p-value: < 2.2e-16
```

The R-squared value of the model with `living`, `beds`, `baths`, and `halfbaths` as predictors is 0.56, which is only slightly higher than the R-squared value of the model with only `living` as a predictor. This suggests that the predictors `beds`, `baths`, and `halfbaths` do not improve the model. This is likely because the predictors are not linearly related to the outcome, and the model is not linear.

3. Create a column `baths2` that is the sum of `baths` and 0.5*`halfbaths`. Fit a linear regression model with `living` and `beds` as before, but use `baths2` instead of `baths` and `halfbaths`. Is this model better, worse or similar? Why?

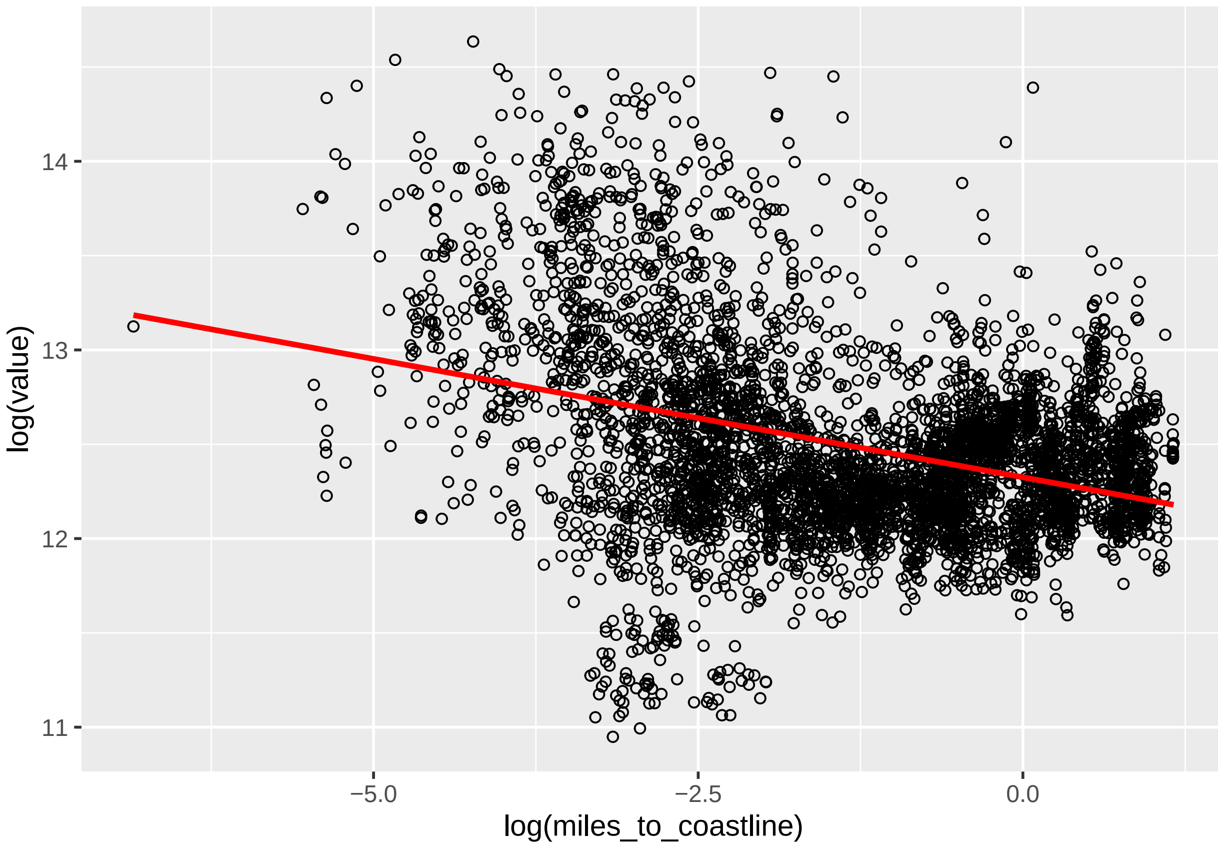
```
b = b %>% mutate(baths2 = baths + 0.5*halfbaths)
lm4 = lm(log(value) ~ living + beds + baths2, data=b)
summary(lm4)
```

```
##
## Call:
## lm(formula = log(value) ~ living + beds + baths2, data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.58726 -0.19212 -0.07353  0.08977  2.43110
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.158e+01  1.806e-02 641.582   <2e-16 ***
## living      3.215e-04  8.856e-06  36.305   <2e-16 ***
## beds        5.983e-03  6.498e-03   0.921    0.357
## baths2      9.737e-02  9.438e-03  10.317   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3372 on 4648 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.5666, Adjusted R-squared:  0.5663
## F-statistic:  2025 on 3 and 4648 DF,  p-value: < 2.2e-16
```

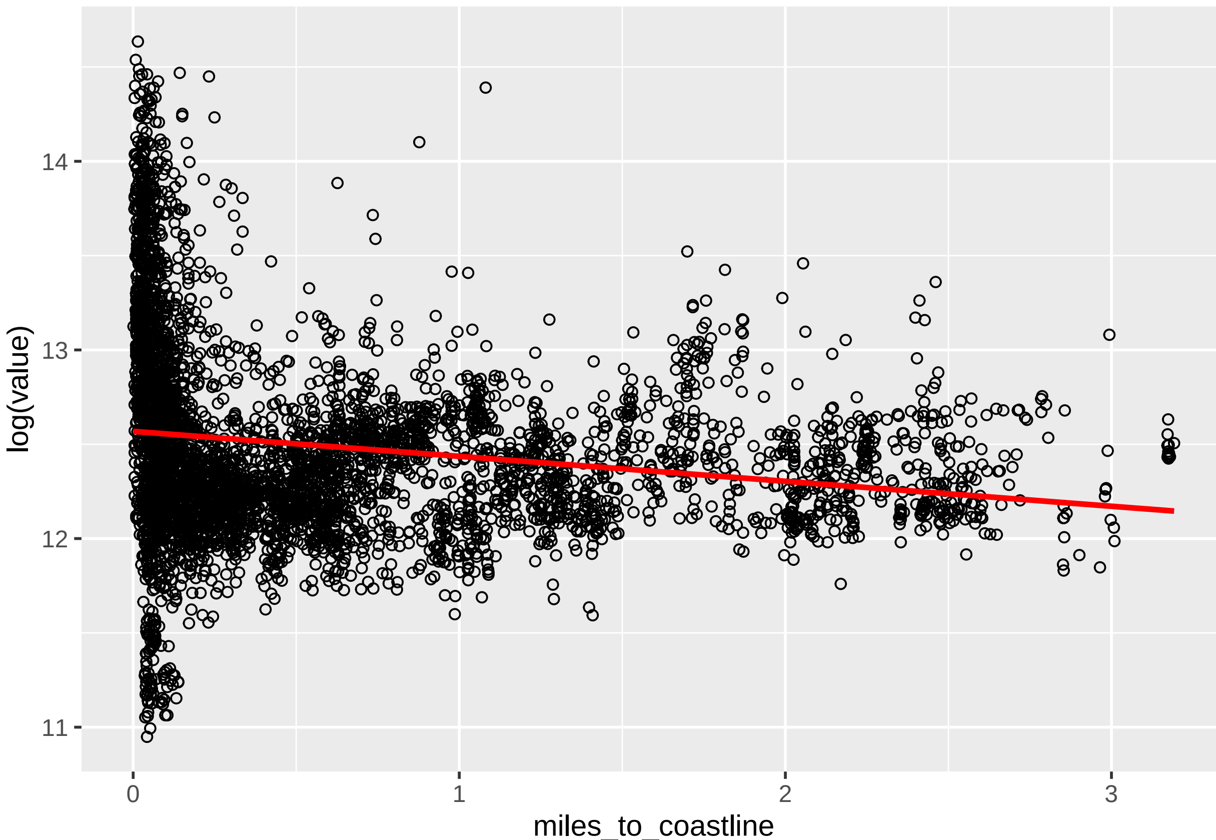
The R-squared value of the model with `living`, `beds`, and `baths2` as predictors is 0.56, which is the same as the R-squared value of the model with `living`, `beds`, and `baths` as predictors. This suggests that the model with `baths2` is not better than the model with `baths` and `halfbaths` as predictors. The two models have a similar performance.

4. Plot `log(value)` vs `log(miles_to_coastline)`. Does this relationship look more linear than `log(value)` vs `miles_to_coastline`?

```
ggplot(b, aes(x=log(miles_to_coastline), y=log(value))) +
  geom_point(shape = 1, color = 'black') +
  geom_smooth(method = 'lm', formula = y ~ x, se = F, color = 'red')
```




```r
ggplot(b, aes(x=miles_to_coastline, y=log(value))) +
  geom_point(shape = 1, color = 'black') +
  geom_smooth(method = 'lm', formula = y ~ x, se = F, color = 'red')
```



The relationship between `log(value)` and `log(miles_to_coastline)` looks more linear than the relationship between `log(value)` and `miles_to_coastline`. This is because the scatter plot of `log(value)` vs `log(miles_to_coastline)` has a more linear shape than the scatter plot of `log(value)` vs `miles_to_coastline`.

5. Fit a model like your model above but with `miles_to_coastline` as an additional predictor, and one like your model above but with `log(miles_to_coastline)` as an additional predictor. Which model is better? Why or why not? Explain what you mean by "better".

```
lm5 = lm(log(value) ~ living + beds + baths2 + miles_to_coastline, data=b)
lm6 = lm(log(value) ~ living + beds + baths2 + log(miles_to_coastline), data=b)
summary(lm5)
```

```
##
## Call:
## lm(formula = log(value) ~ living + beds + baths2 + miles_to_coastline,
##     data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53457 -0.18859 -0.04584  0.12820  2.35429
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.164e+01  1.708e-02 681.646  < 2e-16 ***
## living             3.237e-04  8.301e-06  38.998  < 2e-16 ***
## beds               2.333e-02  6.128e-03   3.807 0.000143 ***
## baths2             9.082e-02  8.850e-03  10.262  < 2e-16 ***
```

```
## miles_to_coastline -1.672e-01  6.590e-03 -25.374  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.316 on 4647 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.6193, Adjusted R-squared:  0.619
## F-statistic:  1890 on 4 and 4647 DF,  p-value: < 2.2e-16
```

```r
summary(lm6)
```

```
##
## Call:
## lm(formula = log(value) ~ living + beds + baths2 + log(miles_to_coastline),
##     data = b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.29780 -0.15700 -0.00948  0.14088  2.26293
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.136e+01  1.589e-02 714.528  < 2e-16 ***
## living                   3.130e-04  7.390e-06  42.358  < 2e-16 ***
## beds                     3.966e-02  5.471e-03   7.249  4.9e-13 ***
## baths2                   8.456e-02  7.877e-03  10.734  < 2e-16 ***
## log(miles_to_coastline) -1.299e-01  2.881e-03 -45.095  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2812 on 4647 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.6985, Adjusted R-squared:  0.6983
## F-statistic:  2692 on 4 and 4647 DF,  p-value: < 2.2e-16
```

The R-squared value of the model with `living`, `beds`, `baths2`, and `miles_to_coastline` as predictors is 0.61, which is lower than the R-squared value of the model with `living`, `beds`, `baths2`, and `log(miles_to_coastline)` as predictors (0.69). This suggests that the model with `log(miles_to_coastline)` as a predictor is better than the model with `miles_to_coastline` as a predictor. By adding the `log(miles_to_coastline)` as a predictor, the model is able to explain more of the variation in the outcome. It also makes every one the of the predictors to be statistically significant, which is not the case for previous linear models.

Part 2: Census Data

The following data set contains information from the US Census. Each row is a census tract, and the columns contain information about that census tract.

```r
census = readRDS('data/tracts.and.census.with.EV.stations.rds')
census = census[census$state=='CT',]
df = census@data ## just the data frame, without the polygons
head(df,2)
```

```
##       STATEFP COUNTYFP TRACTCE             AFFGEOID       GEOID NAME LSAD
## 12844      09      001  090300 1400000US09001090300 09001090300  903   CT
## 12845      09      001  090400 1400000US09001090400 09001090400  904   CT
##         ALAND AWATER  meters    miles state tract           county  state.full
```

```
## 12844 4764507      0 4764507 1.839586    CT   903 Fairfield County Connecticut
## 12845 7347827      0 7347827 2.837012    CT   904 Fairfield County Connecticut
##        pop male female  age male.age female.age white black indian.alaskan
## 12844 4611 2333   2278 42.9     42.7       43.1  4230    24              0
## 12845 6518 3355   3163 40.6     36.5       45.7  4742   654             78
##       asian pacific other two.or.more white.not.hisp hisp white.hisp black.hisp
## 12844   180       0    66         111           3888  435        342          0
## 12845   524      24    88         408           4324  613        418          0
##       households i10orless i10to14 i15to19 i20to24 i25to29 i30to34 i35to39
## 12844       1550        56       8      22      34      16      12      31
## 12845       2140        24      18      85       0      89      36      49
##       i40to44 i45to49 i50to59 i60to74 i75to99 i100to124 i125to149 i150to199
## 12844      19      43      59      67     231       204       197       261
## 12845       0      94      24     128     219       343       233       421
##       i200ormore hh.income house.value   male.p female.p  white.p    black.p
## 12844        290    118819      372100 50.59640 49.40360 91.73715  0.5204945
## 12845        377    121802      344600 51.47284 48.52716 72.75238 10.0337527
##        asian.p   hisp.p white.not.hisp.p white.hisp.p black.hisp.p  other.p
## 12844 3.903709 9.433962         84.32010     7.417046            0 3.838647
## 12845 8.039276 9.404725         66.33937     6.413010            0 9.174593
##       rescaled.house.value hh.income.and.house tot.hh.income tot.house.value
## 12844             80487.25            99653.12     184169450       576755000
## 12845             76759.97            99280.99     260656280       737444000
##       tot.hh.income.and.house pop.density hh.density income.density
## 12844               154462344    2506.542   842.5807      100114594
## 12845               212461309    2297.488   754.3148       91877050
##       house.value.density house.and.income.density lev2 lev3
## 12844           313524273                 83965798    2    4
## 12845           259936875                 74889116   NA   NA
```

```r
lm7 = lm(house.value ~ hh.income, data=df)
summary(lm7)
```

6. Build a simple linear regression model that describes the relationship between median household income `hh.income` and median housing value `house.value` by census tract. Explain how you decided which was the independent and which was the dependent variable. Describe any potential issues with your choice, if any.

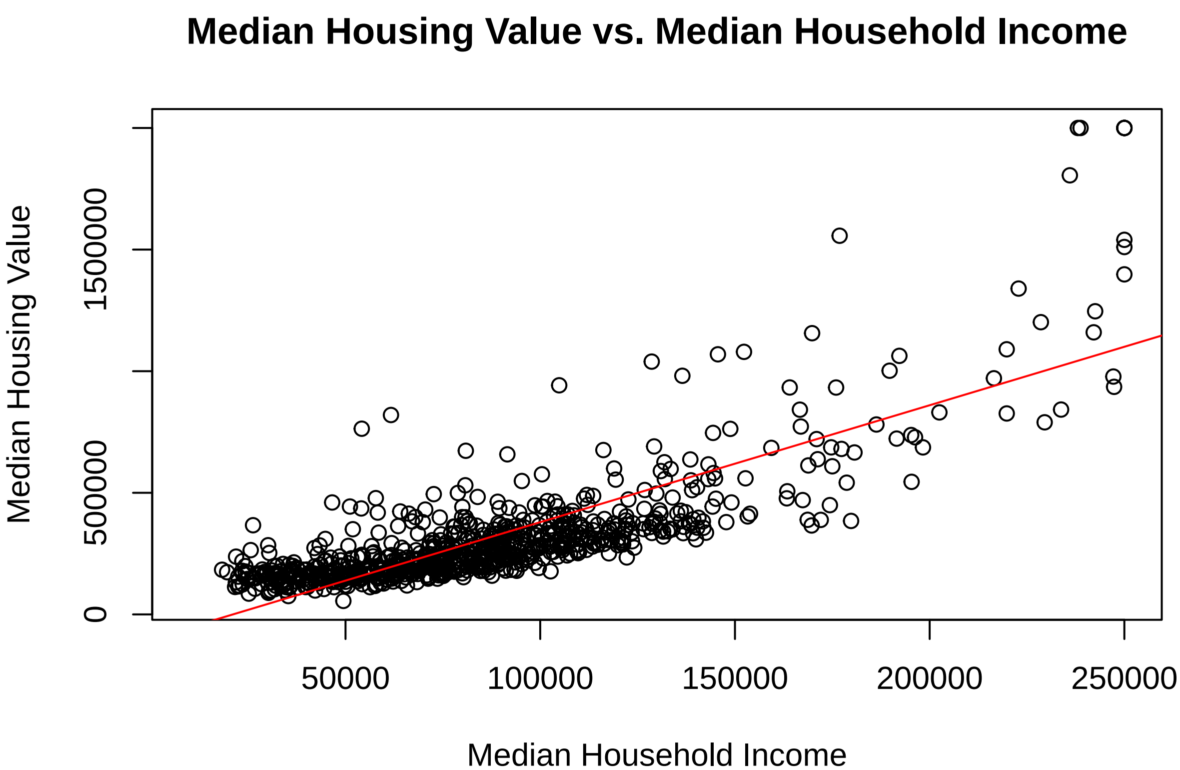
```
##
## Call:
## lm(formula = house.value ~ hh.income, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -378663  -92742  -17749   54430  957048
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.014e+05  1.219e+04  -8.319  3.7e-16 ***
## hh.income    4.807e+00  1.250e-01  38.471  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 156500 on 815 degrees of freedom
```

```
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.6449, Adjusted R-squared:  0.6444
## F-statistic:  1480 on 1 and 815 DF,  p-value: < 2.2e-16
```

The independent variable is `hh.income` and the dependent variable is `house.value`. This is because intuitively how much one earns is the major factor of to what values of house one would buy. The potential issue with this choice is that the relationship between `hh.income` and `house.value` may not be linear, and the model may not be able to capture the relationship between the two variables.

7. For which census tracts in CT does your model perform well? Not as well? What do those census tracts have in common?

```r
plot(df$hh.income, df$house.value,
      xlab = "Median Household Income",
      ylab = "Median Housing Value",
      main = "Median Housing Value vs. Median Household Income")
abline(lm7, col="red")
```



The model performs well for census tracts with median household income below 150k. The model does not perform well for census tracts with median household income above 150k. The census tracts that the model performs well have in common that they have a linear relationship between `hh.income` and `house.value`. The census tracts that the model does not perform well have in common that they have a non-linear relationship between `hh.income` and `house.value`.

One common feature of both groups is that in general `hh.income` and `house.value` are positively correlated.

Part 3: Super Bowl predictions

The Superbowl is the championship game of the National Football League (NFL). The game will likely be the most watched broadcast in the US this year (in 2023, the Superbowl has twice the viewership of the next most watched broadcast, according to https://www.sportico.com/law/analysis/2024/super-bowl-security-1234765332/).

Let's use our regression skills to predict the outcome of this year's Superbowl.

```r
g = readRDS('data/games.rds')

g = g %>%
  filter(lg=='nfl', season %in% 2023) %>%
  select(date, away, home, ascore, hscore, season, gid)

da = g %>% select(date, away, ascore, home, hscore, season, gid) %>% mutate(ha = 'away')
dh = g %>% select(date, home, hscore, away, ascore, season, gid) %>% mutate(ha = 'home')
colnames(da) = c('date', 'team', 'score',  'opp', 'opp.score', 'season', 'gid', 'ha')
colnames(dh) = c('date', 'team', 'score',  'opp', 'opp.score', 'season', 'gid', 'ha')
dd = bind_rows(da, dh) %>%
  arrange(date, gid)
head(dd)
```

```
##         date team score opp opp.score season        gid   ha
## 1 2023-09-07  DET    21  KC        20   2023 2023090700 away
## 2 2023-09-07   KC    20 DET        21   2023 2023090700 home
## 3 2023-09-10  CAR    10 ATL        24   2023 2023091000 away
## 4 2023-09-10  ATL    24 CAR        10   2023 2023091000 home
## 5 2023-09-10  HOU     9 BAL        25   2023 2023091001 away
## 6 2023-09-10  BAL    25 HOU         9   2023 2023091001 home
```

```r
## Super Bowl, and games before the Super Bowl
sb = dd %>% filter(date=='2024-02-11')
dd = dd %>% filter(date< '2024-02-11')
```

```r
lm8 = lm(score ~ ha + team + opp, data=dd)
summary(lm8)
```

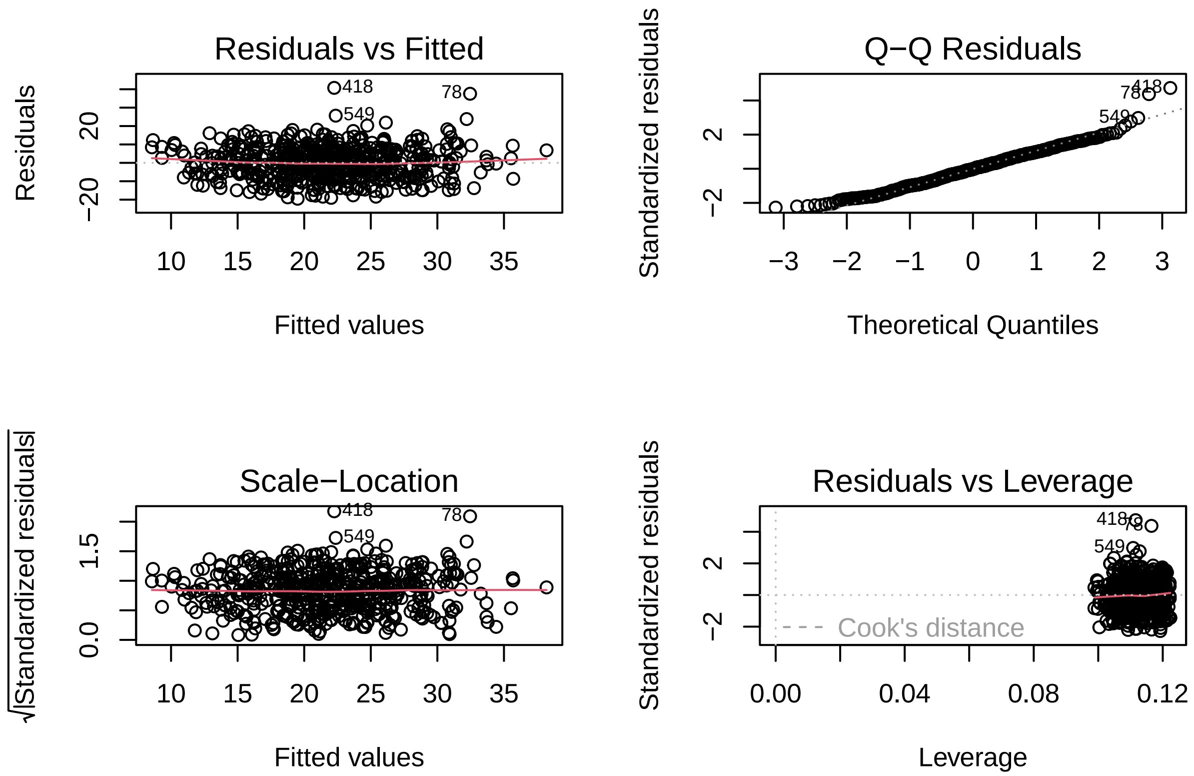
8. Build a model with `score` as the outcome and `ha` (home or away), `team`, and `opp` as the predictors, using only games played before the Super Bowl (the data frame `dd`). Are the linear regression assumptions satisfied?

```
##
## Call:
## lm(formula = score ~ ha + team + opp, data = dd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.504  -6.309  -0.112   6.046  40.747
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.24338    3.24371   6.857 2.06e-11 ***
## hahome       2.66738    0.77234   3.454 0.000600 ***
## teamATL     -1.25092    3.25404  -0.384 0.700828
## teamBAL      8.05584    3.11056   2.590 0.009880 **
## teamBUF      6.76826    3.16195   2.141 0.032791 *
## teamCAR     -5.69691    3.23153  -1.763 0.078521 .
## teamCHI      1.16895    3.25508   0.359 0.719659
## teamCIN      3.26333    3.19270   1.022 0.307213
## teamCLE      3.96663    3.14407   1.262 0.207668
## teamDAL      8.61169    3.15756   2.727 0.006608 **
## teamDEN      1.48577    3.25657   0.456 0.648416
```

```
## teamDET       8.82164    3.09485   2.850 0.004545 **
## teamGB        4.94716    3.13254   1.579 0.114900
## teamHOU       3.30407    3.15532   1.047 0.295537
## teamIND       4.53877    3.22099   1.409 0.159415
## teamJAX       3.43024    3.22085   1.065 0.287381
## teamKC        2.21439    3.14040   0.705 0.481055
## teamLA        3.66896    3.12808   1.173 0.241385
## teamLAC       1.67440    3.26231   0.513 0.607998
## teamLV       -0.09769    3.26182  -0.030 0.976119
## teamMIA       8.74196    3.20905   2.724 0.006670 **
## teamMIN       0.76522    3.23369   0.237 0.813031
## teamNE       -6.45841    3.24885  -1.988 0.047363 *
## teamNO        3.99730    3.23196   1.237 0.216735
## teamNYG      -5.15055    3.20156  -1.609 0.108294
## teamNYJ      -4.71637    3.23739  -1.457 0.145782
## teamPHI       4.83071    3.16154   1.528 0.127149
## teamPIT      -1.01437    3.14864  -0.322 0.747465
## teamSEA       1.59758    3.17144   0.504 0.614663
## teamSF        8.77545    3.08896   2.841 0.004681 **
## teamTB        1.03394    3.15082   0.328 0.742935
## teamTEN      -1.87100    3.22049  -0.581 0.561523
## teamWAS      -1.01325    3.19005  -0.318 0.750899
## oppATL       -2.77576    3.25404  -0.853 0.394053
## oppBAL      -10.39974    3.11056  -3.343 0.000889 ***
## oppBUF       -5.55687    3.16195  -1.757 0.079453 .
## oppCAR       -2.22122    3.23153  -0.687 0.492173
## oppCHI       -3.44268    3.25508  -1.058 0.290729
## oppCIN       -4.15792    3.19270  -1.302 0.193402
## oppCLE       -3.48241    3.14407  -1.108 0.268557
## oppDAL       -4.82471    3.15756  -1.528 0.127143
## oppDEN       -1.19733    3.25657  -0.368 0.713278
## oppDET       -2.27039    3.09485  -0.734 0.463532
## oppGB        -4.51203    3.13254  -1.440 0.150383
## oppHOU       -4.15649    3.15532  -1.317 0.188338
## oppIND       -0.21593    3.22099  -0.067 0.946577
## oppJAX       -4.14901    3.22085  -1.288 0.198277
## oppKC        -9.88976    3.14040  -3.149 0.001734 **
## oppLA        -5.03642    3.12808  -1.610 0.108009
## oppLAC       -2.55979    3.26231  -0.785 0.433025
## oppLV        -5.74340    3.26182  -1.761 0.078880 .
## oppMIA       -0.76844    3.20905  -0.239 0.810845
## oppMIN       -5.30877    3.23369  -1.642 0.101274
## oppNE        -4.42008    3.24885  -1.361 0.174278
## oppNO        -4.73419    3.23196  -1.465 0.143598
## oppNYG       -2.77802    3.20156  -0.868 0.385966
## oppNYJ       -4.77384    3.23739  -1.475 0.140945
## oppPHI        0.25693    3.16154   0.081 0.935263
## oppPIT       -7.14235    3.14864  -2.268 0.023727 *
## oppSEA       -2.81391    3.17144  -0.887 0.375358
## oppSF        -7.77431    3.08896  -2.517 0.012152 *
## oppTB        -7.23836    3.15082  -2.297 0.022011 *
## oppTEN       -4.67008    3.22049  -1.450 0.147648
## oppWAS        4.67595    3.19005   1.466 0.143328
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.132 on 504 degrees of freedom
## Multiple R-squared:  0.2669, Adjusted R-squared:  0.1753
## F-statistic: 2.913 on 63 and 504 DF,  p-value: 3.685e-11
```

```r
par(mfrow = c(2,2))
plot(lm8)
```



From the residual vs fitted plot, the residuals are generally normally randomly distributed around 0, which means the gaussian noise assumption is fulfilled.

9. Use this model to predict the outcome of the Super Bowl.

```r
sb_selected_feature = sb %>% select(ha, team, opp)
head(sb_selected_feature)
```

```
##     ha team opp
## 1 away   SF  KC
## 2 home   KC  SF
```

```r
predict(lm8, newdata=sb_selected_feature, interval = "confidence", level = 0.95)
```

```
##        fit      lwr      upr
## 1 21.12907 15.09577 27.16238
## 2 19.35083 13.31753 25.38414
```

The model predicts that the away team (SF) will score 21.12 points, and the home team (KC) will score 19.35 points. SF is expected to win the game.

10. Suppose the betting odds say that the San Francisco 49ers (SF) are favored to win by 1.5 points. If you bet on the 49ers, and they win by 2 or more points, you win money, and if they win by 1 point or lose, you lose money. Should you bet on the 49ers? If you do this assignment after the Super Bowl, ignore the

outcome when answering this question. :)

No, because the expected mean values of the scores are 21.12 and 19.35 for SF and KC, respectively. The expected difference is 1.77, which is lower than 2.