Analysis

In this analysis we aim to show that QR can achieve similar, if not better, performance to OLS across various metrics. In order to make the comparisons fair, we will compare the 50th quantile QR, which corresponds to the median, to OLS regression as both the median and mean are measures of centrality. The power of QR is that it is able to produce similar results to OLS regression without having to meet the strict assumptions of OLS such as the assumption of normality. In fact, in this data set neither the response variable or predictor variables meet the assumptions of OLS, and therefore regardless of the performance of OLS it is invalid.

Visualization

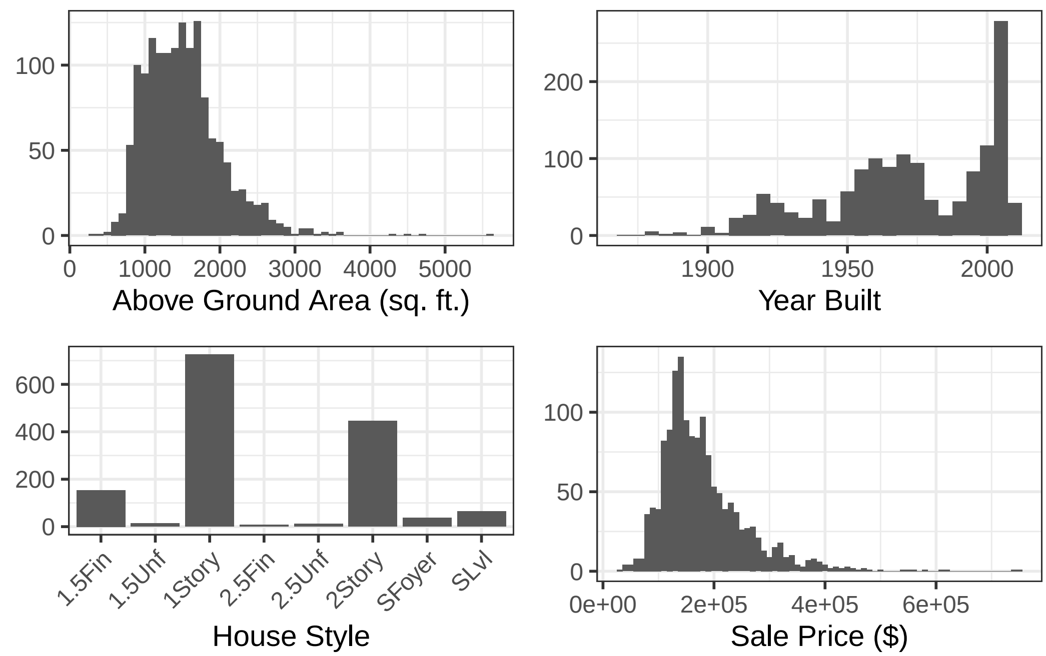
```
df <- read.csv("TrainData.csv") |>
  na.omit() |>
  distinct()
```

Visualizing data

There are many different kinds of predictor variables in this data set. For instance, there are continuous variables like GrLivArea, discrete/coutning variables likr YearBuilt, and categorical variables like HouseStyle. In all cases we cases we can see that the data is not normally distributed, including in the response variable, SalePrice. Thus, the assumptions of OLS are not met so it cannot be used to make predictions on the data. However, for the purposes of comparing the performance of OLS to QR. We will show that QR is able to give similar results for this data set to OLS, and because it does not require the same assumptions as OLS, one can actually use QR in practice for this kind of data, which is more common than normally distributed data in many important fields, like finance and epidemiology.

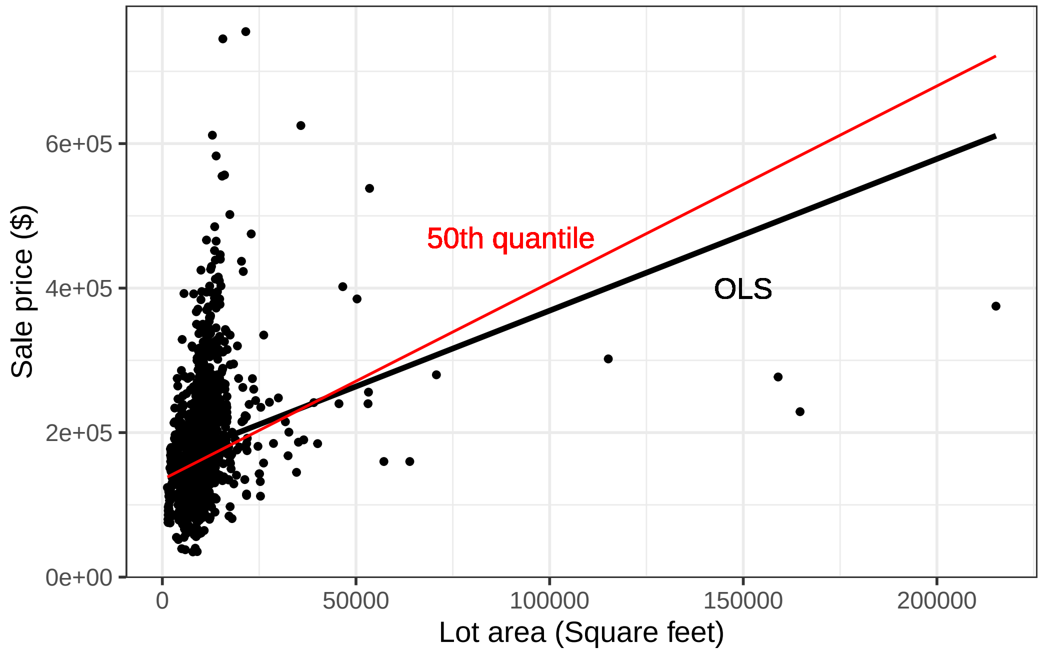
```
suppressWarnings({

p1 <- df |> ggplot(aes(x = GrLivArea)) +
  geom_histogram(binwidth = 100) +
```

```r
  theme_bw() +
  ylab(NULL) +
  xlab("Above Ground Area (sq. ft.)")

p2 <- df |> ggplot(aes(x = YearBuilt)) +
  geom_histogram(binwidth = 5) +
  theme_bw() +
  ylab(NULL) +
  xlab("Year Built")

p3 <- df |> ggplot(aes(x = HouseStyle)) +
  geom_histogram(stat="count") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylab(NULL) +
  xlab("House Style")

p4 <- df |> ggplot(aes(x = SalePrice)) +
  geom_histogram(binwidth = 10000) +
  theme_bw() +
  ylab(NULL) +
  xlab("Sale Price ($)")

grid.arrange(p1, p2, p3, p4, nrow = 2)

})
```

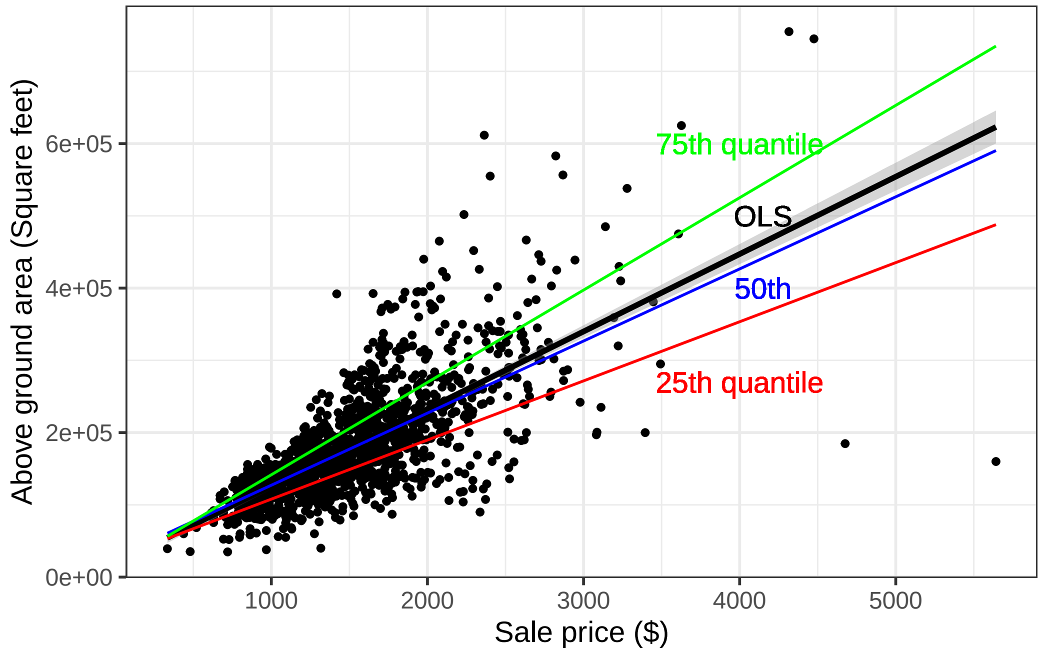


Visualizing quantile regression vs OLS

```
df |> ggplot(aes(y = SalePrice, x = LotArea)) +
  geom_point(size = 0.9) +
  geom_smooth(method = lm, se = F, color = "black") +
  geom_text(aes(y = 400000, x = 150000, label = "OLS"), color="black") +
  geom_quantile(quantiles=0.5, color="red") +
  geom_text(aes(y = 470000, x = 90000, label = "50th quantile"), color="red") +
  ylab("Sale price ($)") +
  xlab("Lot area (Square feet)") +
  theme_bw()
```




```
# df |> ggplot(aes(y = SalePrice, x = GrLivArea)) +
#   geom_boxplot()

df |> ggplot(aes(y = SalePrice, x = GrLivArea)) +
  geom_point(size = 0.9) +
  stat_smooth(method = lm, color = "black") +
  geom_text(aes(x = 4150, y = 500000, label = "OLS"), color="black") +
  geom_quantile(quantiles=0.25, color="red") +
  geom_text(aes(x = 4000, y = 270000, label = "25th quantile"), color="red") +
  geom_quantile(quantiles=0.5, color="blue") +
  geom_text(aes(x = 4150, y = 400000, label = "50th"), color="blue") +
  geom_quantile(quantiles=0.75, color="green") +
  geom_text(aes(x = 4000, y = 600000, label = "75th quantile"), color="green") +
  xlab("Sale price ($)") +
  ylab("Above ground area (Square feet)") +
  theme_bw()
```



Model creation

QR model

```
qr50 = rq(data=df, SalePrice ~ GrLivArea + LotArea + TotRmsAbvGrd + as.factor(LotShape) +
qr50_summary = summary(qr50)
qr50_summary
```

```
Call: rq(formula = SalePrice ~ GrLivArea + LotArea + TotRmsAbvGrd +
    as.factor(LotShape) + as.factor(Foundation), tau = 0.5, data = df)

tau: [1] 0.5

Coefficients:
                            Value          Std. Error    t value      Pr(>|t|)
(Intercept)                  36326.81296    3853.84854      9.42611      0.00000
GrLivArea                       96.66934       4.02708     24.00481      0.00000
LotArea                          0.99940       0.32815      3.04561      0.00236
TotRmsAbvGrd                 -6476.18114    1080.95132     -5.99119      0.00000
as.factor(LotShape)IR2       -5084.13375    7841.20685     -0.64839      0.51684
```

```
as.factor(LotShape)IR3      -21074.80675   7616.42154     -2.76702      0.00573
as.factor(LotShape)Reg      -11065.07360   2020.92512     -5.47525      0.00000
as.factor(Foundation)CBlock  21252.40678   1709.40460     12.43264      0.00000
as.factor(Foundation)PConc   53311.16094   2618.05941     20.36285      0.00000
as.factor(Foundation)Slab   -16867.20619   5378.30454     -3.13616      0.00175
as.factor(Foundation)Stone   14561.54748  13561.64146      1.07373      0.28312
as.factor(Foundation)Wood    -2008.81877   9022.14216     -0.22265      0.82384
```

OLS model

```r
ols = lm(data=df, SalePrice ~ GrLivArea + LotArea + TotRmsAbvGrd + as.factor(LotShape) + a
ols_summary = summary(ols)
ols_summary
```

```
Call:
lm(formula = SalePrice ~ GrLivArea + LotArea + TotRmsAbvGrd +
    as.factor(LotShape) + as.factor(Foundation), data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-422488  -26194    -805   20461  326538

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                  2.005e+04  7.267e+03   2.759  0.00587 **
GrLivArea                    9.893e+01  4.538e+00  21.801  < 2e-16 ***
LotArea                      9.173e-01  1.425e-01   6.438 1.64e-10 ***
TotRmsAbvGrd                -4.313e+03  1.396e+03  -3.089  0.00205 **
as.factor(LotShape)IR2      -2.009e+03  8.113e+03  -0.248  0.80446
as.factor(LotShape)IR3      -6.936e+04  1.603e+04  -4.328 1.61e-05 ***
as.factor(LotShape)Reg      -1.342e+04  2.809e+03  -4.777 1.96e-06 ***
as.factor(Foundation)CBlock  2.094e+04  4.497e+03   4.656 3.52e-06 ***
as.factor(Foundation)PConc   6.679e+04  4.541e+03  14.708  < 2e-16 ***
as.factor(Foundation)Slab   -1.426e+04  1.067e+04  -1.336  0.18170
as.factor(Foundation)Stone  -3.396e+03  2.021e+04  -0.168  0.86658
as.factor(Foundation)Wood   -5.553e+02  2.842e+04  -0.020  0.98441
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 48410 on 1448 degrees of freedom
Multiple R-squared:  0.6315,    Adjusted R-squared:  0.6287
```

```
F-statistic: 225.6 on 11 and 1448 DF,  p-value: < 2.2e-16
```

Model evaluation

Mean absolute error

```
olsMae = mae(predict(ols), df$SalePrice)
olsMae
```

```
[1] 32186.89
```

```
Qr50Mae = mae(predict(qr50), df$SalePrice)
Qr50Mae
```

```
[1] 31160.69
```

OLS MAE value: 32186.89.

And QR 50th MAE value: 31160.69.

QR for 50th quantile has a lower MAE therefore it is has more accurate predictions.

Root mean squared error

```
olsRmse = rmse(predict(ols), df$SalePrice)
olsRmse
```

```
[1] 48209.34
```

```
Qr50Rmse = rmse(predict(qr50), df$SalePrice)
Qr50Rmse
```

```
[1] 49434.81
```

OLS RMSE value: 48209.34.

And QR 50th RMSE value: 49434.81.

Since OLS algorithm's goal is to minimize RMSE, as expected it has a better (lower) value. But QR has a very similar value which shows how well QR model can keep up even if it is not focusing on optimizing RMSE.

Variance of error

```
ols_summary$df[2]
```

```
[1] 1448
```

```
qr50_summary$rdf
```

```
[1] 1448
```

The variance of error for OLS: 1448.

The variance of error for QR 50th: 1448.

Both have the same variance of error.

Min/max error

```
# Min OLS error
format(round(min(ols_summary$residuals), digits=0), scientific=F)
```

```
[1] "-422488"
```

```
# Absolute min OLS error
format(round(min(abs(ols_summary$residuals)), digits=0), scientific=F)
```

```
[1] "5"
```

```r
# Max OLS error
format(round(max(ols_summary$residuals), digits=0), scientific=F)
```

```
[1] "326538"
```

```r
# Absolute max OLS error
format(round(max(abs(ols_summary$residuals)), digits=0), scientific=F)
```

```
[1] "422488"
```

```r
# Min QR 50th error
format(round(min(qr50_summary$residuals), digits=0), scientific=F)
```

```
[1] "-440106"
```

```r
# Absolute min QR 50th error
format(round(min(abs(qr50_summary$residuals)), digits=0), scientific=F)
```

```
[1] "0"
```

```r
# Max QR 50th error
format(round(max(qr50_summary$residuals), digits=0), scientific=F)
```

```
[1] "351819"
```

```r
# Absolute max QR 50th error
format(round(max(abs(qr50_summary$residuals)), digits=0), scientific=F)
```

```
[1] "440106"
```

OLS

Min OLS error: -422488.

Absolute min OLS error: 5.

Max OLS error: 326538.

Absolute max OLS error: 422488.

QR

Min QR 50th error: -440106.

Absolute min QR 50th error: 0.

Max QR 50th error: 351819.

Absolute max QR 50th error: 440106.