GAM – Piecewise, MARS, LOESS, and Splines - R Version

Goal: Predict `Sale_Price` from a focused set of Ames features using increasingly flexible methods in the GAM framework.

1. Setup and Data

```r
set.seed(4321)

library(tidyverse)
library(dplyr)
library(tidyr)
library(AmesHousing)   # make_ordinal_ames()
library(earth)         # MARS
library(segmented)     # piecewise (segmented) regression
library(splines)       # regression splines (bs, ns) - kept for reference
library(mgcv)          # smoothing splines via GAM
library(caret)         # data splitting / utilities
library(Metrics)       # rmse, mse helpers (plus we'll compute R2 manually)
theme_set(theme_minimal())
```

```r
# Limit to requested columns
cols <- c("Sale_Price","Bedroom_AbvGr","Year_Built","Mo_Sold","Lot_Area","Street",
          "Central_Air","First_Flr_SF","Second_Flr_SF","Full_Bath","Half_Bath",
          "Fireplaces","Garage_Area","Gr_Liv_Area","TotRms_AbvGrd")

ames <- make_ordinal_ames() |>
  dplyr::select(dplyr::all_of(cols)) |>
  tidyr::drop_na()

```

```r
# Train/test split
set.seed(4321)
idx <- sample.int(nrow(ames), size = floor(0.7*nrow(ames)))
train <- ames[idx,]
test  <- ames[-idx,]

glimpse(train)
```

```
Rows: 2,051
Columns: 15
$ Sale_Price    <int> 274000, 75200, 329900, 145400, 108000, 184000, 176000, 1~
$ Bedroom_AbvGr <int> 3, 2, 4, 3, 2, 2, 3, 3, 3, 3, 3, 3, 3, 4, 3, 2, 3, 3, 2,~
$ Year_Built    <int> 2001, 1922, 2005, 1926, 1949, 1999, 1962, 1915, 1999, 19~
$ Mo_Sold       <int> 1, 9, 8, 5, 5, 6, 5, 6, 9, 6, 8, 5, 7, 7, 4, 8, 5, 11, 5~
$ Lot_Area      <int> 9720, 3672, 11643, 7000, 8777, 5858, 19296, 8094, 3768, ~
$ Street        <fct> Pave, Pave, Pave, Pave, Pave, Pave, Pave, Pave, Pave, Pa~
$ Central_Air   <fct> Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y,~
$ First_Flr_SF  <int> 1366, 816, 1544, 861, 1126, 1337, 1382, 1048, 713, 792, ~
$ Second_Flr_SF <int> 581, 0, 814, 424, 0, 0, 0, 720, 739, 725, 0, 1151, 695, ~
$ Full_Bath     <int> 2, 1, 2, 1, 2, 2, 1, 2, 2, 1, 1, 2, 2, 2, 1, 2, 2, 1, 1,~
$ Half_Bath     <int> 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0,~
$ Fireplaces    <int> 1, 0, 1, 0, 0, 1, 1, 0, 0, 2, 0, 1, 2, 1, 0, 1, 0, 0, 1,~
$ Garage_Area   <dbl> 725, 100, 784, 506, 520, 511, 884, 576, 506, 400, 0, 434~
$ Gr_Liv_Area   <int> 1947, 816, 2358, 1285, 1126, 1337, 1382, 1768, 1452, 151~
$ TotRms_AbvGrd <int> 7, 5, 10, 6, 5, 5, 6, 8, 6, 7, 5, 8, 7, 5, 6, 7, 6, 6, 5~
```

```r
summary(train$Sale_Price)
```

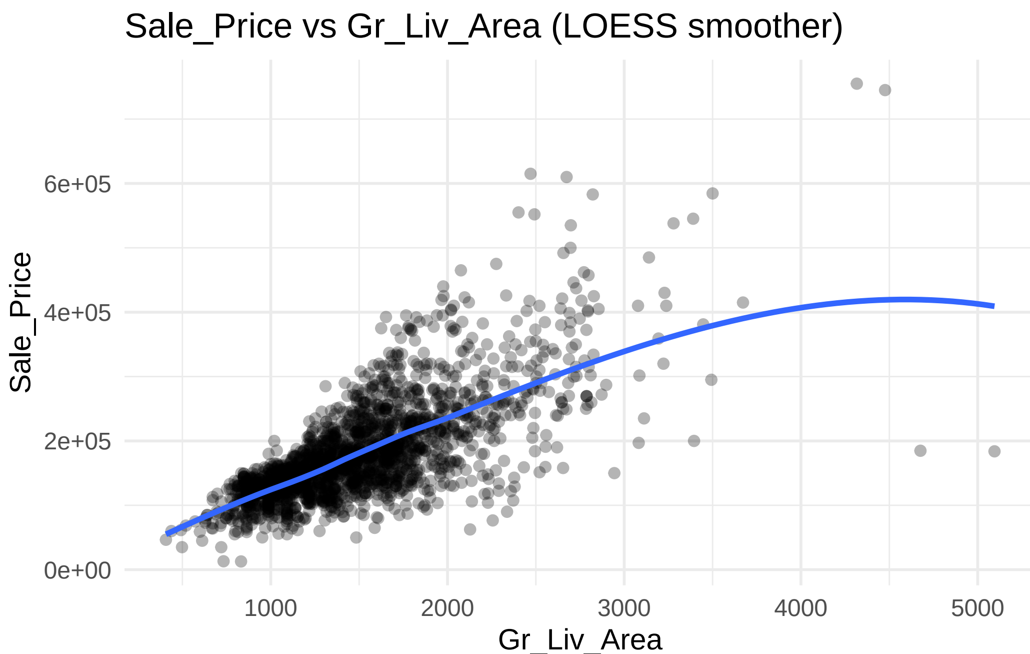
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  12789  130000  161000  180897  215000  755000 
```

---

2. Quick Visual: Nonlinearity is Common

```r
ggplot(train, aes(Gr_Liv_Area, Sale_Price)) +
  geom_point(alpha=.3) +
  geom_smooth(method="loess", se=FALSE) +
  labs(title="Sale_Price vs Gr_Liv_Area (LOESS smoother)")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Takeaway: The relationship bends—nonlinear methods can help.

---

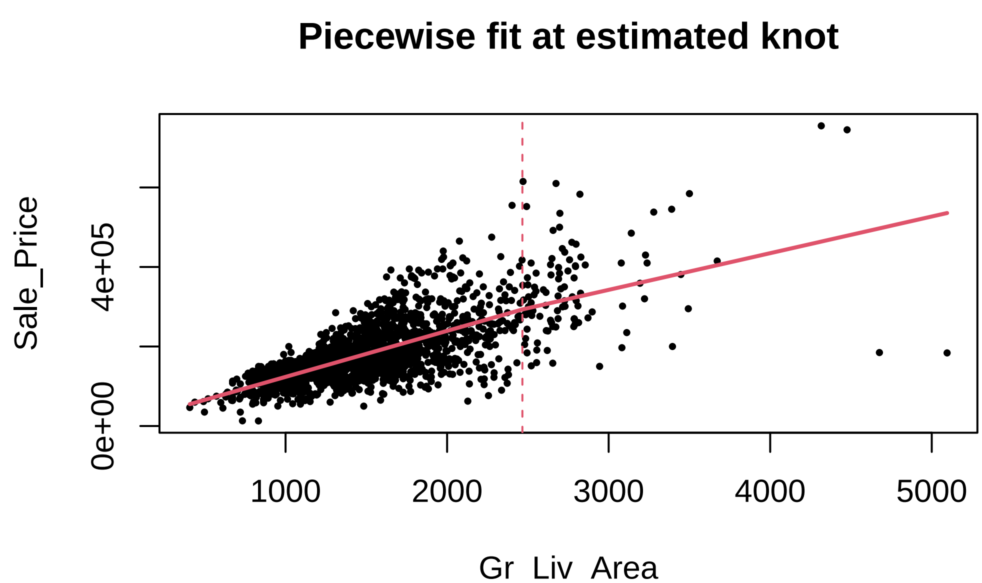
3. Piecewise Regression

```r
m_lin <- lm(Sale_Price ~ Gr_Liv_Area, data=train)
m_seg_init <- lm(Sale_Price ~ Gr_Liv_Area, data=train)
m_seg <- segmented(m_seg_init, seg.Z = ~ Gr_Liv_Area, psi = list(Gr_Liv_Area = 2000))
summary(m_seg)
```

```
	***Regression Model with Segmented Relationship(s)***

Call:
segmented.lm(obj = m_seg_init, seg.Z = ~Gr_Liv_Area, psi = list(Gr_Liv_Area = 2000))

Estimated Break-Point(s):
```

```
                  Est.  St.Err
psi1.Gr_Liv_Area 2466 332.881

Coefficients of the linear terms:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    7901.186   4698.641   1.682   0.0928 .
Gr_Liv_Area     115.705      3.171  36.488   <2e-16 ***
U1.Gr_Liv_Area  -23.479     12.436  -1.888       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 55110 on 2047 degrees of freedom
Multiple R-Squared: 0.5124,  Adjusted R-squared: 0.5116

Boot restarting based on 6 samples. Last fit:
Convergence attained in 1 iterations (rel. change 0)
```

```r
plot(train$Gr_Liv_Area, train$Sale_Price, pch=16, cex=.5,
     xlab="Gr_Liv_Area", ylab="Sale_Price", main="Piecewise fit at estimated knot")
plot(m_seg, add=TRUE, col=2, lwd=2)
abline(v = m_seg$psi[,"Est."], lty=2, col=2)
```



Interpretation (piecewise): Two linear slopes before/after a knot. Useful when you expect a threshold effect.

---

4. MARS (earth) – From Simple to Interactions

MARS = *Multivariate Adaptive Regression Splines*. It builds **hinge** basis functions like `h(x - c) = max(0, x - c)` and can **interact** them.
- A term such as `h(Gr_Liv_Area - 1500)` means *once* `Gr_Liv_Area` exceeds 1500, the fitted line's slope can change.
- An interaction like `h(Gr_Liv_Area - 1500) * Central_AirY` means the slope change applies when `Central_Air == "Y"`.

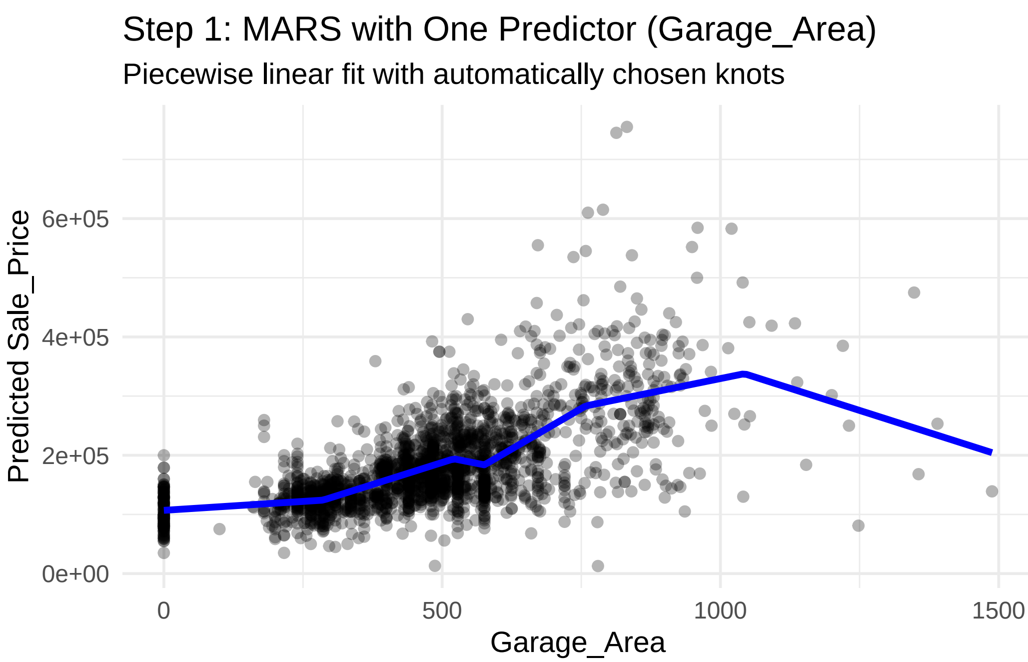
4A. Univariate MARS (Garage_Area)

```r
mars1 <- earth(Sale_Price ~ Garage_Area, data=train)
summary(mars1)
```

```
Call: earth(formula=Sale_Price~Garage_Area, data=train)

                    coefficients
(Intercept)           124159.039
h(286-Garage_Area)       -60.257
h(Garage_Area-286)       297.277
h(Garage_Area-521)      -483.642
h(Garage_Area-576)       733.859
h(Garage_Area-758)      -356.460
h(Garage_Area-1043)     -490.873

Selected 7 of 7 terms, and 1 of 1 predictors
Termination condition: RSq changed by less than 0.001 at 7 terms
Importance: Garage_Area
Number of terms at each degree of interaction: 1 6 (additive model)
GCV 3427475346    RSS 6.94092e+12    GRSq 0.4492014    RSq 0.4556309
```

```r
grid <- tibble(Garage_Area = seq(min(train$Garage_Area), max(train$Garage_Area), length.out=
grid$pred <- predict(mars1, newdata=grid)

ggplot(train, aes(Garage_Area, Sale_Price)) +
```

```r
  geom_point(alpha=.3) +
  geom_line(data=grid, aes(Garage_Area, pred), color="blue", linewidth=1.2) +
  labs(title="Step 1: MARS with One Predictor (Garage_Area)",
       subtitle="Piecewise linear fit with automatically chosen knots",
       y="Predicted Sale_Price")
```



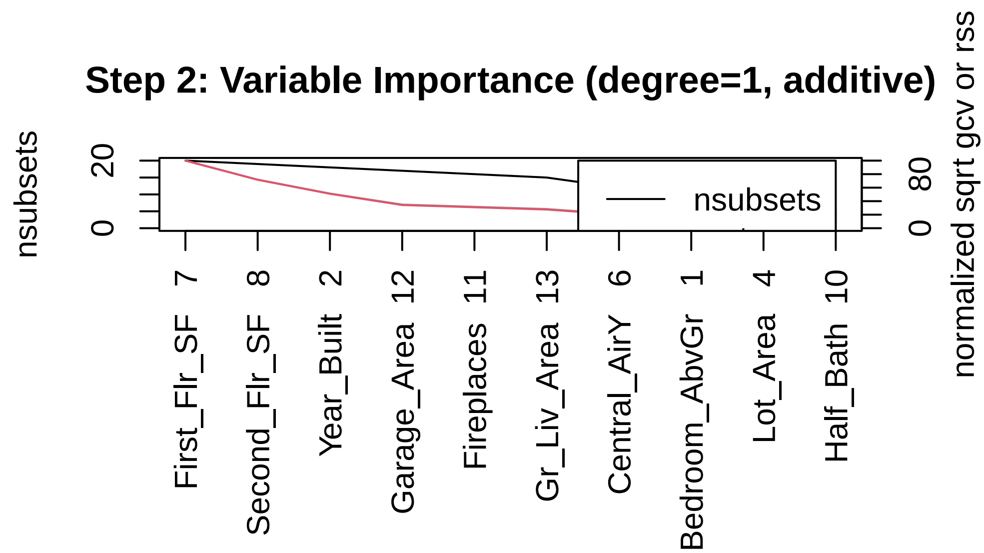
4B. Additive MARS (degree = 1; no interactions)

```r
mars2 <- earth(Sale_Price ~ Bedroom_AbvGr + Year_Built + Mo_Sold + Lot_Area +
                 Street + Central_Air + First_Flr_SF + Second_Flr_SF + Full_Bath +
                 Half_Bath + Fireplaces + Garage_Area + Gr_Liv_Area + TotRms_AbvGrd,
               data=train, degree=1, nfold=5)
summary(mars2)
```

```
Call: earth(formula=Sale_Price~Bedroom_AbvGr+Year_Built+Mo_Sold+Lot_Ar...),
            data=train, degree=1, nfold=5)

                      coefficients
(Intercept)              319493.46
Central_AirY              20289.49
```

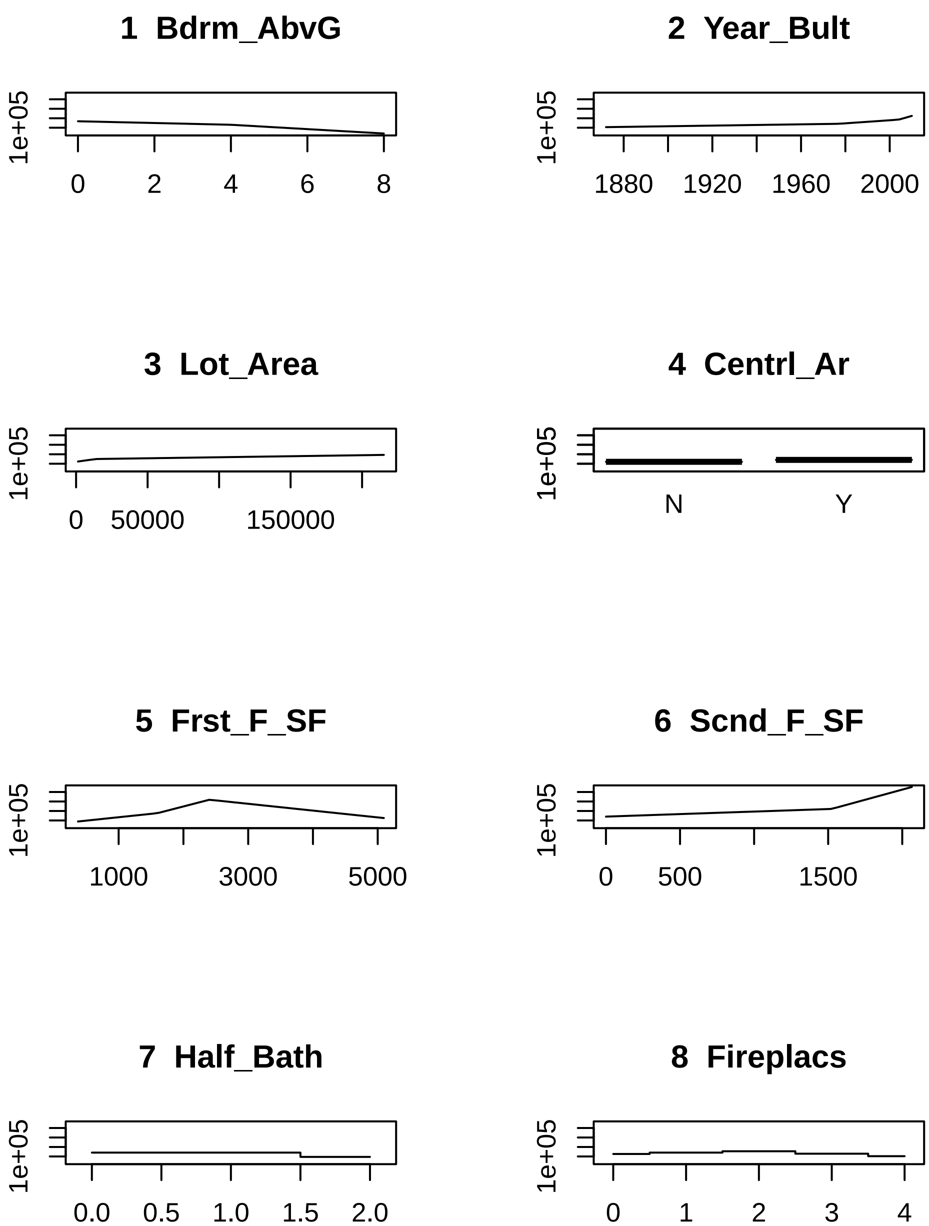
```
h(4-Bedroom_AbvGr)         9214.66
h(Bedroom_AbvGr-4)       -23009.05
h(Year_Built-1977)         1275.57
h(2004-Year_Built)         -336.64
h(Year_Built-2004)         5315.57
h(13869-Lot_Area)            -2.09
h(Lot_Area-13869)             0.22
h(First_Flr_SF-1600)        104.91
h(2402-First_Flr_SF)        -71.56
h(First_Flr_SF-2402)       -176.61
h(1523-Second_Flr_SF)       -53.13
h(Second_Flr_SF-1523)       426.63
h(Half_Bath-1)           -45378.31
h(2-Fireplaces)          -14408.56
h(Fireplaces-2)          -26072.58
h(Garage_Area-539)          101.97
h(Garage_Area-1043)        -294.30
h(Gr_Liv_Area-2049)          65.21
h(Gr_Liv_Area-3194)        -159.79

Selected 21 of 24 terms, and 10 of 14 predictors
Termination condition: Reached nk 29
Importance: First_Flr_SF, Second_Flr_SF, Year_Built, Garage_Area, ...
Number of terms at each degree of interaction: 1 20 (additive model)
GCV 1033819964  RSS 2.036439e+12  GRSq 0.8338641  RSq 0.8402842  CVRSq 0.7978671

Note: the cross-validation sd's below are standard deviations across folds

Cross validation:   nterms 21.00 sd 0.71    nvars 9.20 sd 0.84

     CVRSq    sd     MaxErr     sd
     0.798 0.043    -428941 282429
```

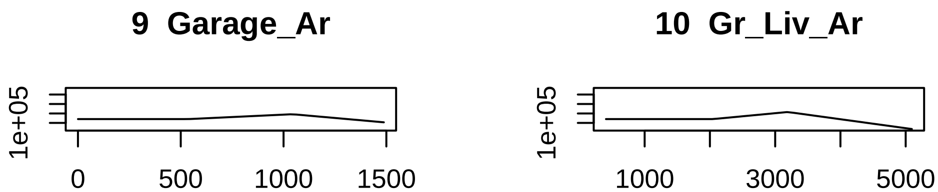
```r
ev2 <- evimp(mars2); plot(ev2, main="Step 2: Variable Importance (degree=1, additive)")
```




```r
par(mfrow=c(2,2)); plotmo(mars2, type="response", nresponse=1, do.par=FALSE); par(mfrow=c(1,
```

```
 plotmo grid:    Bedroom_AbvGr Year_Built Mo_Sold Lot_Area Street Central_Air
                             3       1974       6     9480   Pave           Y
 First_Flr_SF Second_Flr_SF Full_Bath Half_Bath Fireplaces Garage_Area
         1092             0         2         0          1         480
 Gr_Liv_Area TotRms_AbvGrd
        1442             6
```


1 Bdrm_AbvG
1e+05
0
2
4
6
8
2 Year_Bult
1e+05
1880
1920
1960
2000
3 Lot_Area
1e+05
0
50000
150000
4 Centrl_Ar
1e+05
N
Y
5 Frst_F_SF
1e+05
1000
3000
5000
6 Scnd_F_SF
1e+05
0
500
1500
7 Half_Bath
1e+05
0.0
0.5
1.0
1.5
2.0
8 Fireplacs
1e+05
0
1
2
3
4



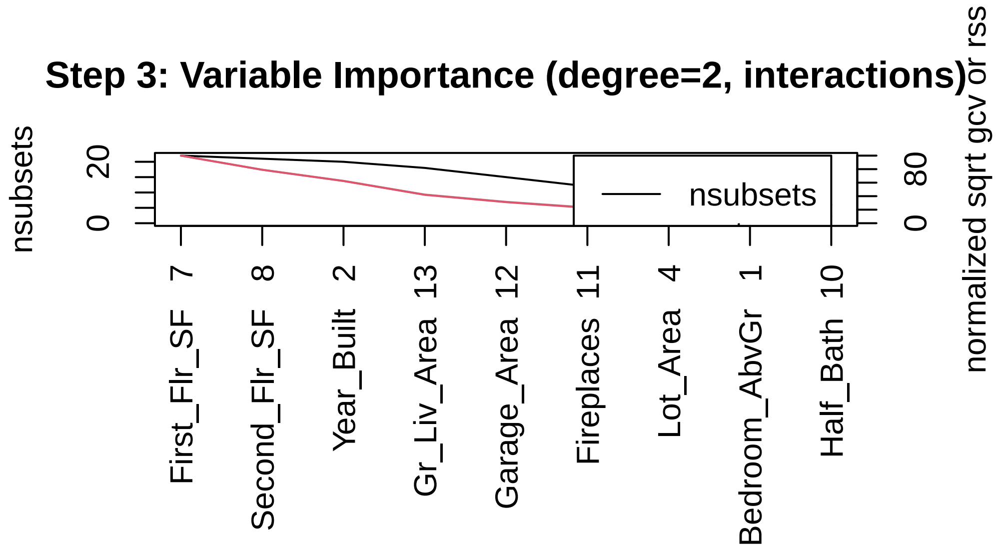
4C. MARS with 2-way Interactions (degree = 2)

```r
mars3 <- earth(Sale_Price ~ Bedroom_AbvGr + Year_Built + Mo_Sold + Lot_Area +
                 Street + Central_Air + First_Flr_SF + Second_Flr_SF + Full_Bath +
                 Half_Bath + Fireplaces + Garage_Area + Gr_Liv_Area + TotRms_AbvGrd,
               data=train, degree=2, nfold=5)
summary(mars3)
```

```
Call: earth(formula=Sale_Price~Bedroom_AbvGr+Year_Built+Mo_Sold+Lot_Ar...),
            data=train, degree=2, nfold=5)

                                          coefficients
(Intercept)                                  411949.27
h(Year_Built-2004)                           169869.37
h(14803-Lot_Area)                                -1.75
h(Lot_Area-14803)                                 0.48
h(1570-First_Flr_SF)                           -150.79
h(First_Flr_SF-1570)                            224.90
h(1523-Second_Flr_SF)                           -86.91
h(Second_Flr_SF-1523)                           156.02
h(1-Half_Bath)                                -7743.19
```

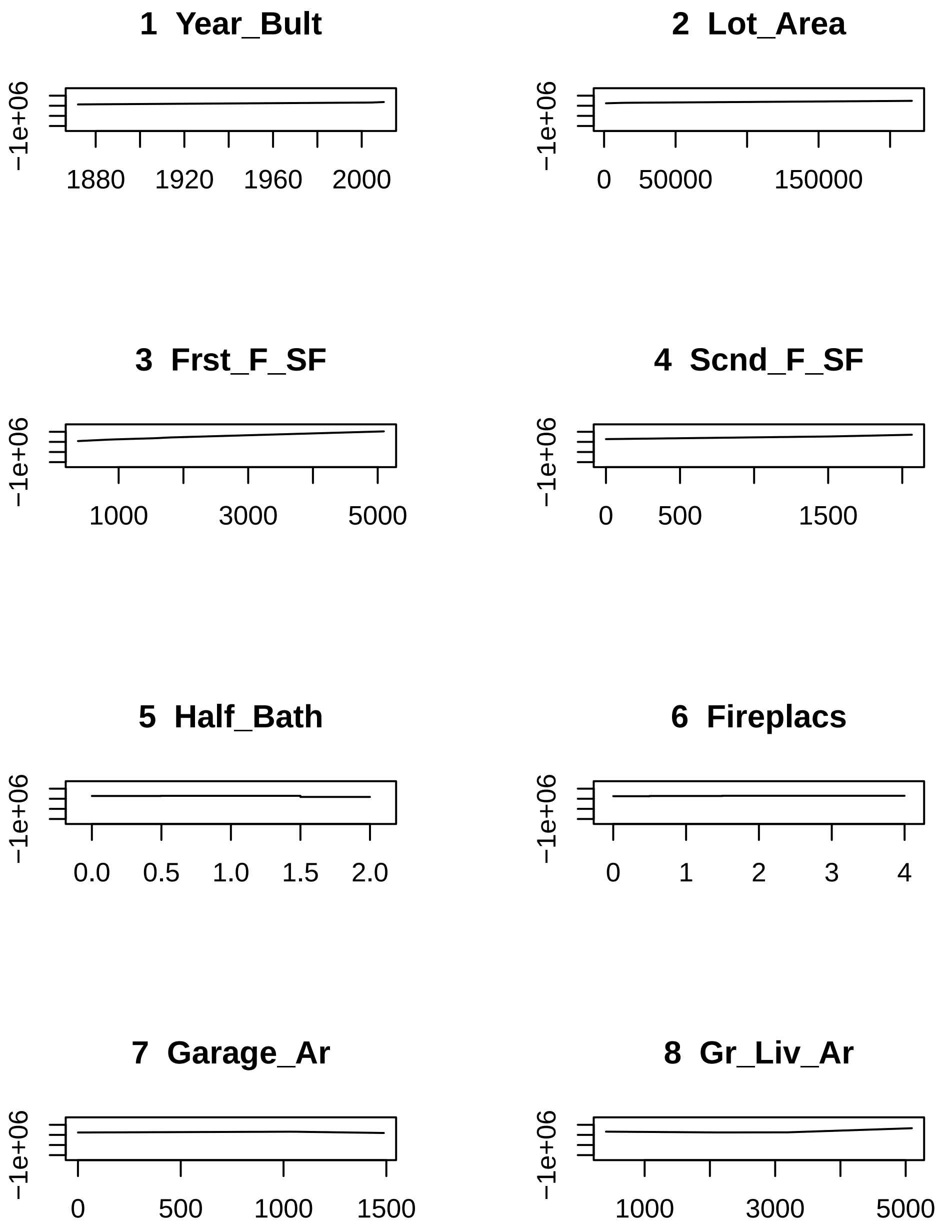
```
h(Half_Bath-1)                                   -54819.24
h(2-Fireplaces)                                  -10516.92
h(1043-Garage_Area)                                 -70.97
h(Garage_Area-1043)                                -141.86
h(Gr_Liv_Area-3194)                                 106.27
h(3-Bedroom_AbvGr) * h(1523-Second_Flr_SF)            7.01
h(Bedroom_AbvGr-3) * h(1523-Second_Flr_SF)           -9.61
h(2004-Year_Built) * h(First_Flr_SF-876)             -1.86
h(2004-Year_Built) * h(3194-Gr_Liv_Area)             -0.17
h(Year_Built-2004) * h(Gr_Liv_Area-2320)           -159.80
h(Year_Built-2004) * h(2320-Gr_Liv_Area)            156.64
h(Year_Built-2004) * h(3194-Gr_Liv_Area)           -173.05
h(First_Flr_SF-1778) * h(2-Fireplaces)              -78.19
h(1043-Garage_Area) * h(2122-Gr_Liv_Area)             0.05

Selected 23 of 29 terms, and 9 of 14 predictors
Termination condition: Reached nk 29
Importance: First_Flr_SF, Second_Flr_SF, Year_Built, Gr_Liv_Area, ...
Number of terms at each degree of interaction: 1 13 9
GCV 795873462  RSS 1.544416e+12  GRSq 0.8721024  RSq 0.8788731  CVRSq 0.7824362

Note: the cross-validation sd's below are standard deviations across folds

Cross validation:   nterms 22.60 sd 1.82    nvars 9.40 sd 1.14

     CVRSq    sd     MaxErr     sd
     0.782 0.114    -945502 422348
```

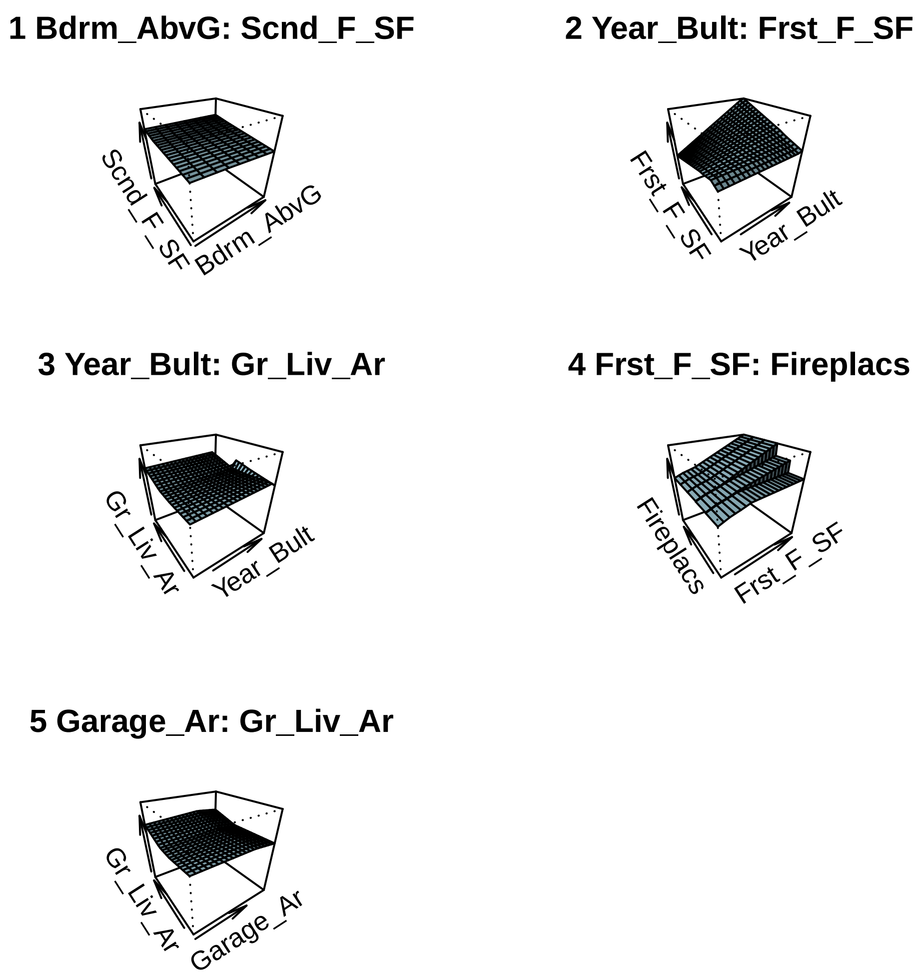
```
ev3 <- evimp(mars3); plot(ev3, main="Step 3: Variable Importance (degree=2, interactions)")
```




```
par(mfrow=c(2,2)); plotmo(mars3, type="response", nresponse=1, do.par=FALSE); par(mfrow=c(1,1
```

```
plotmo grid:    Bedroom_AbvGr Year_Built Mo_Sold Lot_Area Street Central_Air
                            3       1974       6     9480   Pave           Y
First_Flr_SF Second_Flr_SF Full_Bath Half_Bath Fireplaces Garage_Area
        1092                0         2         0          1         480
Gr_Liv_Area TotRms_AbvGrd
       1442             6
```

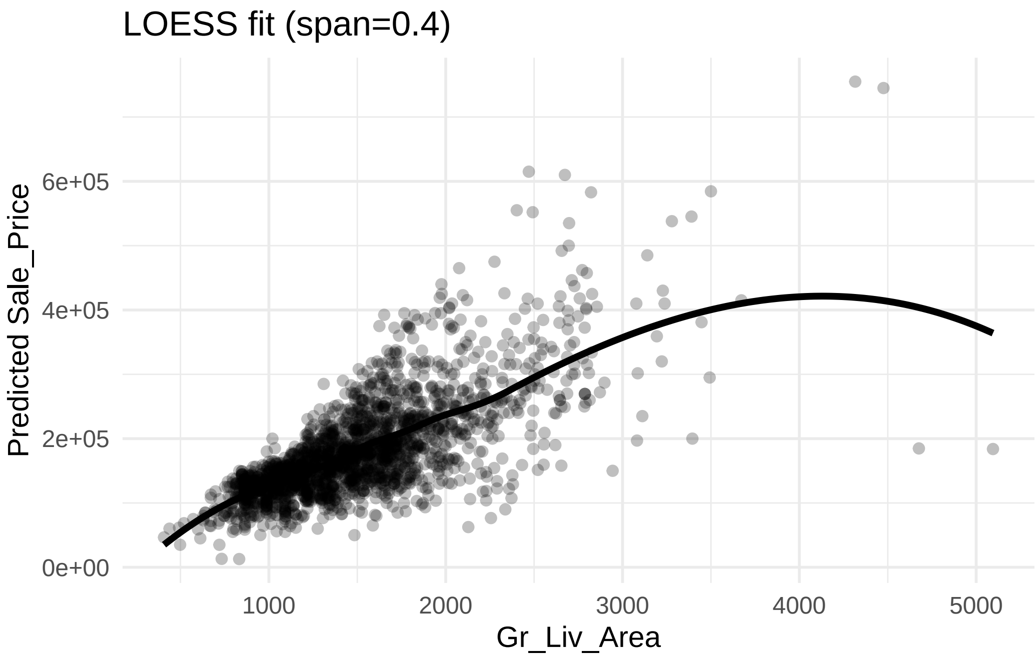

1 Year_Bult
−1e+06
1880 1920 1960 2000
2 Lot_Area
−1e+06
0 50000 150000
3 Frst_F_SF
−1e+06
1000 3000 5000
4 Scnd_F_SF
−1e+06
0 500 1500
5 Half_Bath
−1e+06
0.0 0.5 1.0 1.5 2.0
6 Fireplacs
−1e+06
0 1 2 3 4
7 Garage_Ar
−1e+06
0 500 1000 1500
8 Gr_Liv_Ar
−1e+06
1000 3000 5000



Interpreting MARS terms:
Look for `h()` pieces and products. Coefficients on `h(x-c)` indicate how slope changes after the knot `c`. Interaction products mean “the slope change depends on another variable”.

5. LOESS (Visual local regression)

```
loess_fit <- loess(Sale_Price ~ Gr_Liv_Area, data=train, span=0.4)
grid_lo <- tibble(Gr_Liv_Area = seq(min(train$Gr_Liv_Area), max(train$Gr_Liv_Area), length.ou
grid_lo$pred <- predict(loess_fit, newdata=grid_lo)

ggplot() +
  geom_point(data=train, aes(Gr_Liv_Area, Sale_Price), alpha=.25) +
  geom_line(data=grid_lo, aes(Gr_Liv_Area, pred), linewidth=1.2) +
  labs(title="LOESS fit (span=0.4)", y="Predicted Sale_Price")
```



We keep LOESS as a visualization-oriented smoother (not used in the multivariate comparison below).

6. GAM Splines with `mgcv` – Simple → Complex

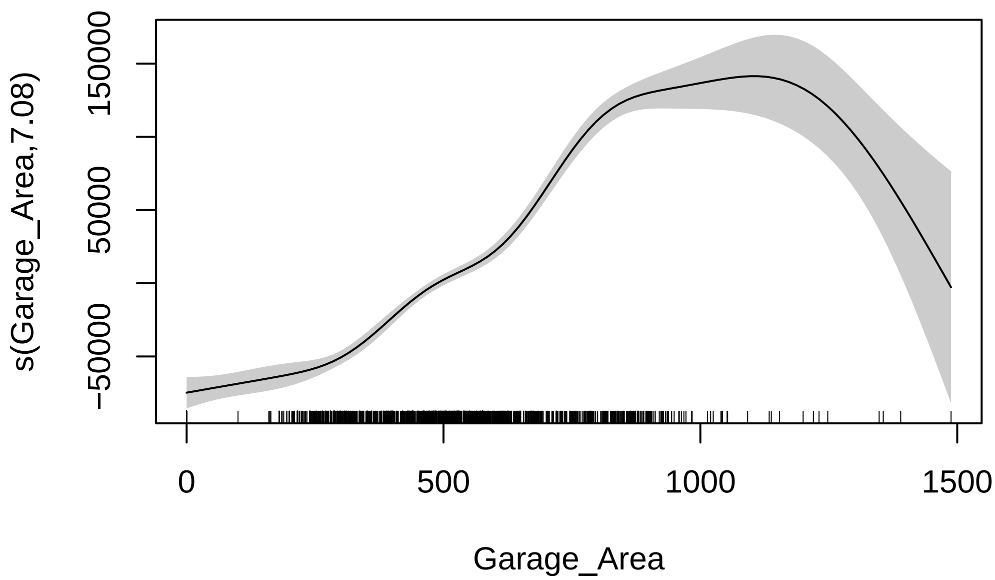
GAM smooths are written `s(x)` and estimated with penalized splines. The **edf** (effective degrees of freedom) in the summary tells you the smooth's complexity (larger edf wigglier function).

6A. Univariate GAM: `Sale_Price ~ s(Garage_Area)`

```r
gam1 <- gam(Sale_Price ~ s(Garage_Area, k=9), data=train, method="REML")
summary(gam1)  # inspect edf and significance
```

```

Family: gaussian
Link function: identity

Formula:
Sale_Price ~ s(Garage_Area, k = 9)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   180897       1292     140   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                 edf Ref.df     F p-value
s(Garage_Area) 7.078  7.719 217.1  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =   0.45   Deviance explained = 45.2%
-REML =  25418  Scale est. = 3.4232e+09  n = 2051
```

```r
plot(gam1, pages=1, shade=TRUE, scheme=1, scale=0)
```



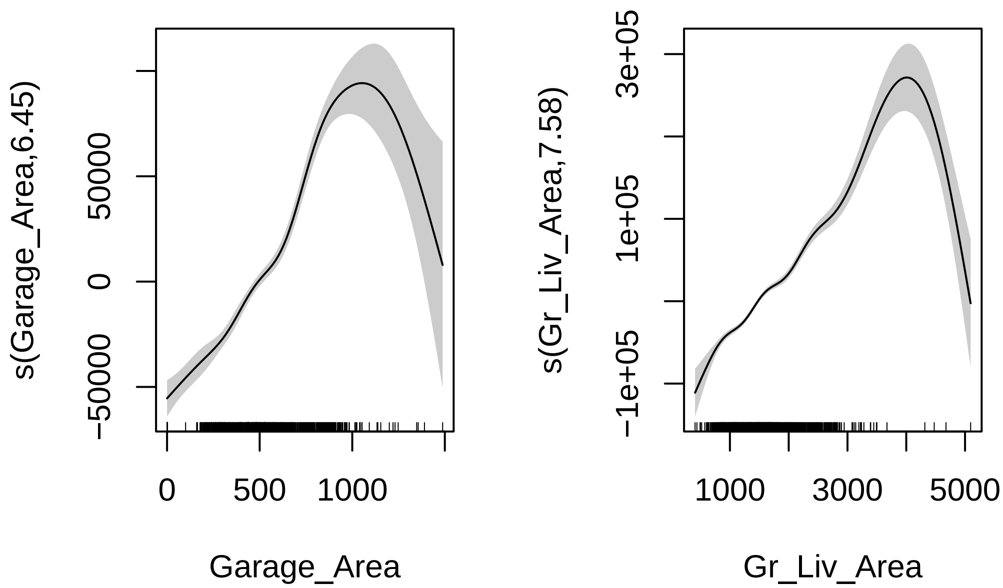
6B. Two-smooth Additive GAM: `s(Garage_Area) + s(Gr_Liv_Area)`

```
gam2 <- gam(Sale_Price ~ s(Garage_Area, k=9) + s(Gr_Liv_Area, k=9), data=train, method="REML"
summary(gam2)
```

```
Family: gaussian
Link function: identity

Formula:
Sale_Price ~ s(Garage_Area, k = 9) + s(Gr_Liv_Area, k = 9)

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   180897       1026   176.2   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                 edf Ref.df     F p-value
s(Garage_Area) 6.446  7.293 103.3  <2e-16 ***
```

```
s(Gr_Liv_Area) 7.585  7.941 152.6  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.653   Deviance explained = 65.5%
-REML =  24949  Scale est. = 2.1608e+09  n = 2051
```

```r
plot(gam2, pages=1, shade=TRUE, scheme=1, scale=0)
```



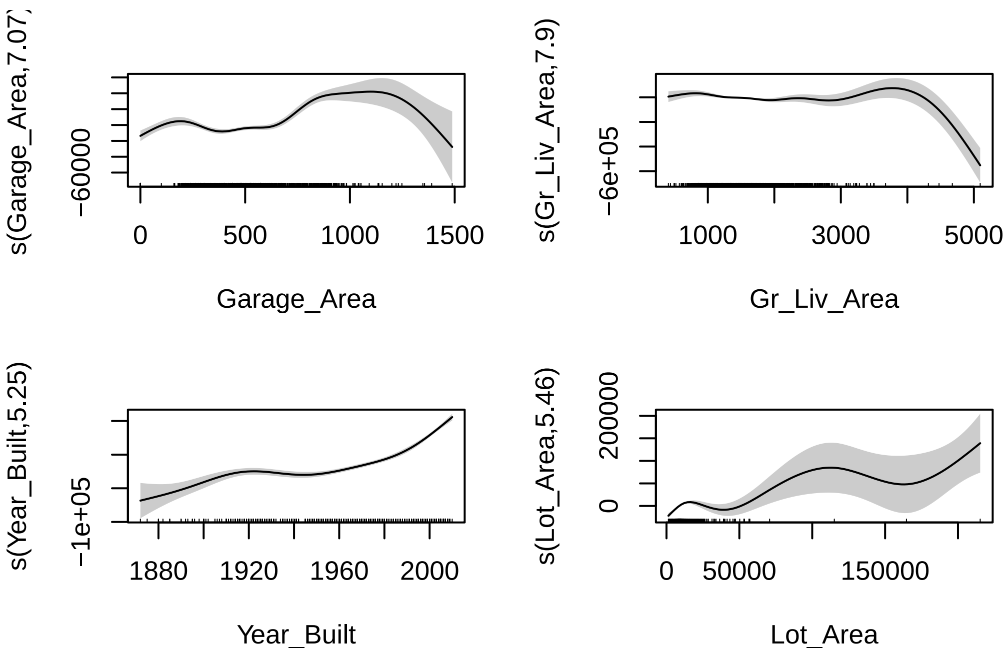
6C. Full Multivariate GAM (selected smooths + factors)

```r
gam3 <- gam(Sale_Price ~
              s(Garage_Area, k=9) +
              s(Gr_Liv_Area, k=9) +
              s(Year_Built, k=7) +
              s(Lot_Area, k=7) +
              Bedroom_AbvGr + Mo_Sold +
              Street + Central_Air +
              First_Flr_SF + Second_Flr_SF +
              Full_Bath + Half_Bath +
```

```r
                Fireplaces + TotRms_AbvGrd,
              data=train, method="REML")
summary(gam3)
```

```
Family: gaussian
Link function: identity

Formula:
Sale_Price ~ s(Garage_Area, k = 9) + s(Gr_Liv_Area, k = 9) +
    s(Year_Built, k = 7) + s(Lot_Area, k = 7) + Bedroom_AbvGr +
    Mo_Sold + Street + Central_Air + First_Flr_SF + Second_Flr_SF +
    Full_Bath + Half_Bath + Fireplaces + TotRms_AbvGrd

Parametric coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   -21237.92   30106.84  -0.705   0.4806
Bedroom_AbvGr -10350.20    1382.38  -7.487 1.05e-13 ***
Mo_Sold         -409.12     270.24  -1.514   0.1302
StreetPave     30589.21   13486.61   2.268   0.0234 *
Central_AirY   17189.68    3366.03   5.107 3.59e-07 ***
First_Flr_SF     140.27      17.44   8.044 1.47e-15 ***
Second_Flr_SF    101.93      17.36   5.871 5.06e-09 ***
Full_Bath      -4548.94    2211.69  -2.057   0.0398 *
Half_Bath       2355.98    2175.40   1.083   0.2789
Fireplaces     13066.61    1371.59   9.527  < 2e-16 ***
TotRms_AbvGrd  -1640.87     948.67  -1.730   0.0838 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                 edf Ref.df      F p-value
s(Garage_Area) 7.067  7.712  24.96  <2e-16 ***
s(Gr_Liv_Area) 7.899  7.996  46.44  <2e-16 ***
s(Year_Built)  5.255  5.764 128.05  <2e-16 ***
s(Lot_Area)    5.465  5.883  16.44  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.827   Deviance explained =   83%
-REML =  24167  Scale est. = 1.0786e+09  n = 2051
```

```r
plot(gam3, pages=1, shade=TRUE, scheme=1, scale=0)
```



Interpreting GAM splines:
A smooth `s(x)` shows how the expected outcome varies with `x` *holding other terms constant*. The edf indicates complexity; wide confidence bands suggest greater uncertainty in those regions.

---

7. Model Comparison (R^2, RMSE, MSE)

```r
# Helper for R^2 on test
rsq <- function(y, yhat) {
  1 - sum((y - yhat)^2) / sum((y - mean(y))^2)
}

# Predictions
p_piece <- predict(m_seg, newdata=test)
p_mars  <- predict(mars3, newdata=test)   # final MARS with interactions
p_gam   <- predict(gam3, newdata=test)    # full multivariate GAM
```

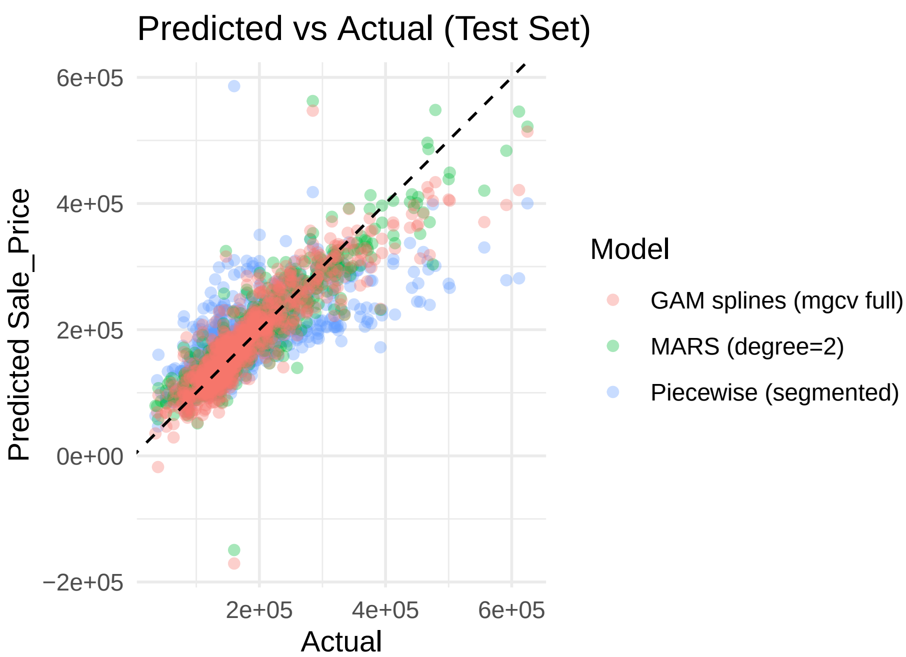
```r

# Metrics
tbl <- tibble(
  Model = c("Piecewise (segmented)", "MARS (degree=2)", "GAM splines (mgcv full)"),
  RMSE  = c(rmse(test$Sale_Price, p_piece),
            rmse(test$Sale_Price, p_mars),
            rmse(test$Sale_Price, p_gam)),
  MSE   = c(mse(test$Sale_Price, p_piece),
            mse(test$Sale_Price, p_mars),
            mse(test$Sale_Price, p_gam)),
  R2    = c(rsq(test$Sale_Price, p_piece),
            rsq(test$Sale_Price, p_mars),
            rsq(test$Sale_Price, p_gam))
) |> arrange(desc(R2))

tbl
```

```
# A tibble: 3 x 4
  Model                      RMSE         MSE    R2
  <chr>                     <dbl>       <dbl> <dbl>
1 MARS (degree=2)          33028. 1090840972. 0.839
2 GAM splines (mgcv full) 36796. 1353918715. 0.800
3 Piecewise (segmented)   59034. 3484963067. 0.484
```

---

8. Final Plot: Predicted vs Actual (Test Set)

```r
plot_df <- bind_rows(
  tibble(Model="Piecewise (segmented)", Actual=test$Sale_Price, Pred=p_piece),
  tibble(Model="MARS (degree=2)", Actual=test$Sale_Price, Pred=p_mars),
  tibble(Model="GAM splines (mgcv full)", Actual=test$Sale_Price, Pred=p_gam)
)

ggplot(plot_df, aes(Actual, Pred, color=Model)) +
  geom_point(alpha=.35) +
  geom_abline(intercept=0, slope=1, linetype="dashed") +
  labs(title="Predicted vs Actual (Test Set)", y="Predicted Sale_Price") +
  coord_equal()
```



9. How to Explain

- **Piecewise**: "There's a threshold where the effect changes." Simple story, limited flexibility.

- **MARS**: "The model finds breakpoints and sometimes *when* they matter (interactions)." Explain `h(x-c)` as *extra slope after c*.

- **GAM (splines)**: "We model smooth curves for key variables; edf shows wiggliness. More edf more flexibility."

- **Tradeoff**: More flexibility usually improves error (RMSE/MSE) and R^2, but reduce interpretability. Validate with cross-validation and keep the story aligned with business intuition.