Model Comparison – Example 2 Wrap-Up

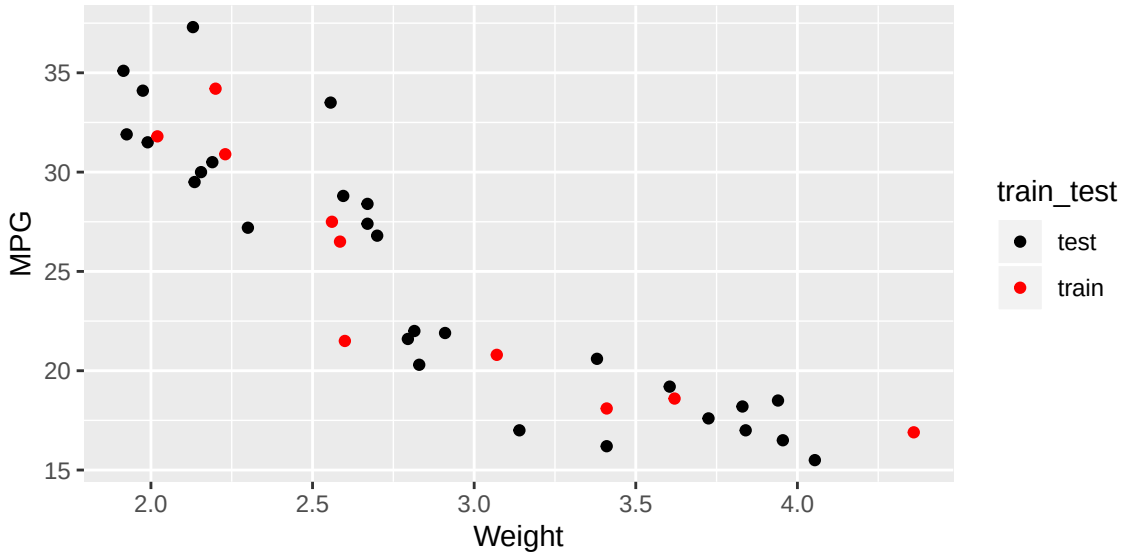
Recall we fit a line, a parabola, and a degree 9 polynomial to model the relationship between a car's weight (`Weight`) and its fuel efficiency (`MPG`).

We fit these models to 10 cars that were selected from a larger data set of 38 cars. These 10 cars are the **training set**: they were used to **train** the model, or estimate the model parameters.

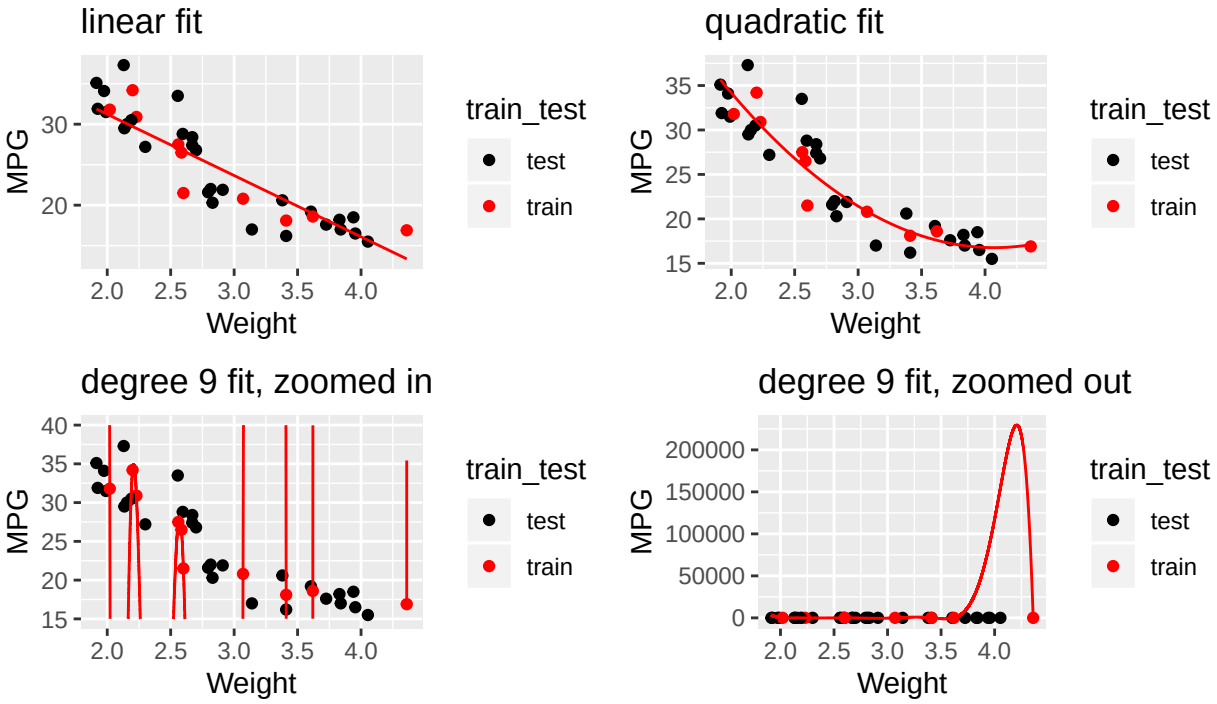
The remaining 28 cars can be used as a **test set**: a set of observations that were *not* used in model estimation, and can therefore be used to independently check the quality of the model fit.

The train and test set are labeled as such in the plot below.

```
library(dplyr) # for data manipulation functions
library(tidyr) # for data manipulation functions
library(readr) # for read_csv, which can read csv files from the internet
library(ggplot2) # for making plots
library(gridExtra) # for grid.arrange, which arranges the plots next to each other
library(polynom) # for obtaining the third polynomial fit below

cars <- read_csv("http://www.evanlray.com/data/sdm4/Cars.csv")
cars$train_test <- "test"
cars$train_test[c(1, 6, 8, 14, 15, 16, 21, 32, 33, 37)] <- "train"

ggplot() +
  geom_point(data = cars, mapping = aes(x = Weight, y = MPG, color = train_test)) +
  scale_color_manual(breaks = c("test", "train"), values = c("black", "red"))
```



Here are the plots again, over both the training and test sets. The estimated curves are shown in red, to indicate that they were fit to the training data set.



Below is R code for calculating and plotting:

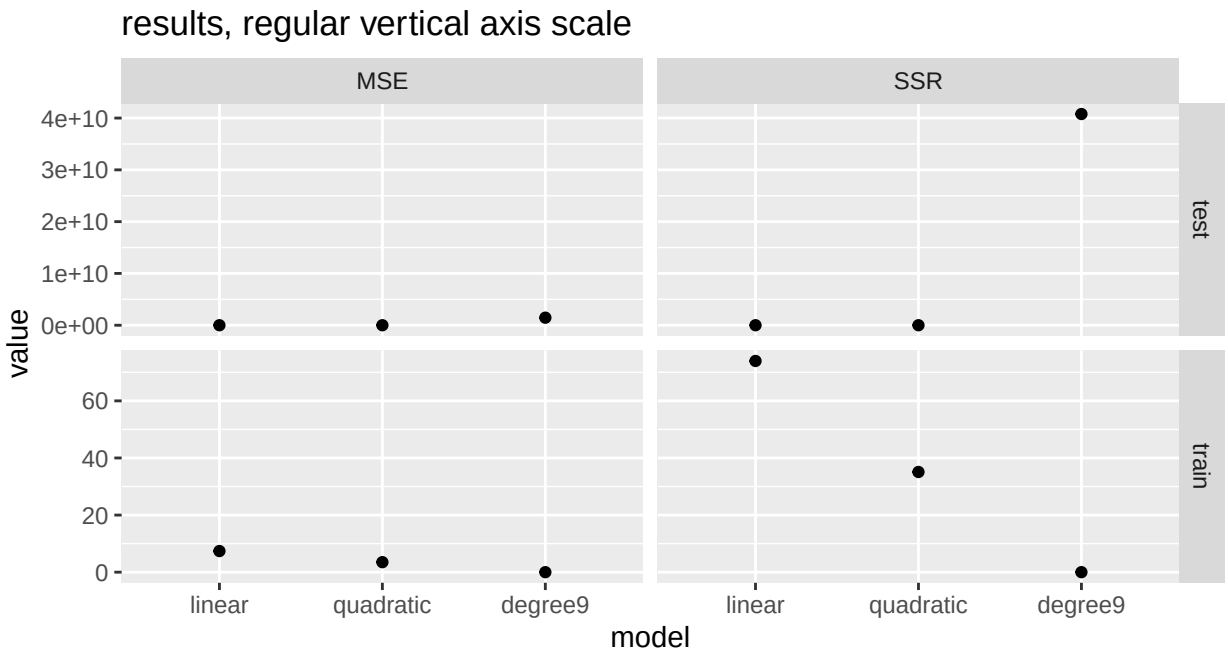
- the mean squared error, separately for the train and test sets for each of our three candidate models; and
- estimates of the residual standard error based on the train and test sets for each of our three candidate models.

```
#' Calculate the Sum of Squared Residuals, SSR
#'
#' @param x a vector of residuals
#'
#' @return sum of squared residuals
SSR <- function(x) {
  sum(x^2)
}

#' Calculate the Mean Squared Error (MSE)
#'
#' @param x a vector of residuals
#'
#' @return mean of squared residuals
MSE <- function(x) {
  mean(x^2)
}

model_residual_summaries <- cars %>%
  transmute(
    train_test = train_test,
    residual_linear = MPG - predict_1(Weight),
    residual_quadratic = MPG - predict_2(Weight),
    residual_degree9 = MPG - predict_9(Weight)
  ) %>%
```

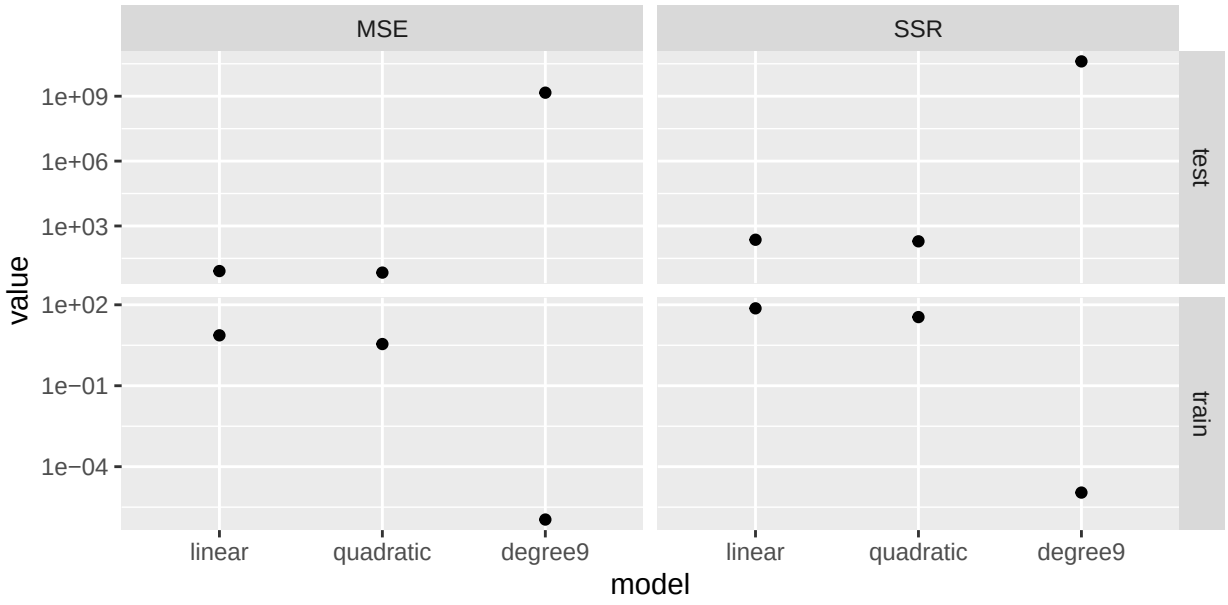
```
  group_by(train_test) %>%
  summarize_all(
    .funs = c("SSR", "MSE")
  ) %>%
  gather("condition", "value", -train_test) %>%
  mutate(
    condition = substr(condition, 10, nchar(condition))
  ) %>%
  separate(col = condition, into = c("model", "summary"), sep = "_") %>%
  mutate(
    model = factor(model, levels = c("linear", "quadratic", "degree9"), ordered = TRUE)
  )

model_residual_summaries
```

```
## # A tibble: 12 x 4
##    train_test model     summary    value
##    <chr>      <ord>     <chr>      <dbl>
##  1 test       linear    SSR     2.30e+ 2
##  2 train      linear    SSR     7.40e+ 1
##  3 test       quadratic SSR     1.95e+ 2
##  4 train      quadratic SSR     3.51e+ 1
##  5 test       degree9   SSR     4.08e+10
##  6 train      degree9   SSR     1.10e- 5
##  7 test       linear    MSE     8.23e+ 0
##  8 train      linear    MSE     7.40e+ 0
##  9 test       quadratic MSE     6.97e+ 0
## 10 train      quadratic MSE     3.51e+ 0
## 11 test       degree9   MSE     1.46e+ 9
## 12 train      degree9   MSE     1.10e- 6
```

```
ggplot(data = model_residual_summaries) +
  geom_point(mapping = aes(x = model, y = value)) +
  facet_grid(train_test ~ summary, scales = "free") +
  ggtitle("results, regular vertical axis scale")
```




```
ggplot(data = model_residual_summaries) +
  geom_point(mapping = aes(x = model, y = value)) +
  facet_grid(train_test ~ summary, scales = "free") +
  scale_y_log10() +
  ggtitle("results, logarithmic vertical axis scale")
```



Additionally, here are the R^2 values for these three models as evaluated on the training data (a "testing data R^2" doesn't really make sense – why?):

```
TSS <- sum((cars$MPG - mean(cars$MPG))^2)

# R^2 linear regression
RSS_linear <- model_residual_summaries %>%
  filter(model == "linear", summary == "SSR", train_test == "train") %>%
  select(value) %>%
  as.numeric()

1 - RSS_linear / TSS
```

```
## [1] 0.9533582
```

```
# R^2 quadratic regression
RSS_quadratic <- model_residual_summaries %>%
  filter(model == "quadratic", summary == "SSR", train_test == "train") %>%
  select(value) %>%
  as.numeric()

1 - RSS_quadratic / TSS
```

```
## [1] 0.9778709
```

```
# R^2 degree 9 polynomial regression
RSS_degree9 <- model_residual_summaries %>%
  filter(model == "degree9", summary == "SSR", train_test == "train") %>%
  select(value) %>%
  as.numeric()

1 - RSS_degree9 / TSS
```

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