Chapter 3: Linear Regression

Linear regression is a simple approach for supervised learning when the response is quantitative. Linear regression has a long history and we could actually spend most of this semester talking about it.

Although linear regression is not the newest, shiniest thing out there, it is still a highly used technique out in the real world. It is also useful for talking about more modern techniques that are **generalizations** of it. ridge regression, lasso, logistic regression, GAMs, etc.

We will review some key ideas underlying linear regression and discuss the least squares approach that is most commonly used to fit this model.

Linear regression can help us to answer the following questions about our `Advertising` data:

1. Is there a relationship btw/ advertising and sales?
2. How strong is the relationship?
3. Which media contribute to sales?
4. How accurately can we predict future sales?
5. How accurately can we estimate the effect of each medium on sales?
6. Is the relationship linear?
7. Is there synergy among the advertising media?

   i.e. is \$50K for TV and \$50K for radio better than \$100K on radio or TV alone?

1 Simple Linear Regression

Simple Linear Regression is an approach for predictiong a quantitative response Y on the basis of a single predictor variable X.

It assumes:

- approximately linear relationship between X and Y
- random error term has constant variance
- random error term is Normally distributed.

Which leads to the following model:

$$Y = \overbrace{\beta_0 + \beta_1 X}^{\text{linear relationship}} + \varepsilon$$

$$\varepsilon \sim \underbrace{N(0, \sigma^2)}_{\text{assumptions about error}}$$

For example, we may be interested in regressing `sales` onto `TV` by fitting the model

$$\text{sales} = \beta_0 + \beta_1 \times \text{TV}$$

intercept & slope unknown constants

"parameters" "model coefficients"

^ hat = estimate or prediction.

Once we have used training data to produce estimates $\hat{\beta}_0$ and $\hat{\beta}_1$, we can predict future sales on the basis of a particular TV advertising budget.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

prediction of sales

particular TV budget value

1.1 Estimating the Coefficients

In practice, β_0 and β_1 are **unknown**, so before we can predict $\hat{y}$, we must use our training data to estimate them.

"fit the model"

Training data

Let $(x_1, y_1), \ldots, (x_n, y_n)$ represent n observation pairs, each of which consists of a measurement of X and Y.

In the advertising data,

X = TV ad budget

Y = sales

n = 200 markets.

Goal: Obtain coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ such that the linear model fits the available data well.

i.e. $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$ for $i = 1, \ldots, n$

We want to find an intercept $\hat{\beta}_0$ and slope $\hat{\beta}_1$ s.t. the resulting line is "close" to all n observations.

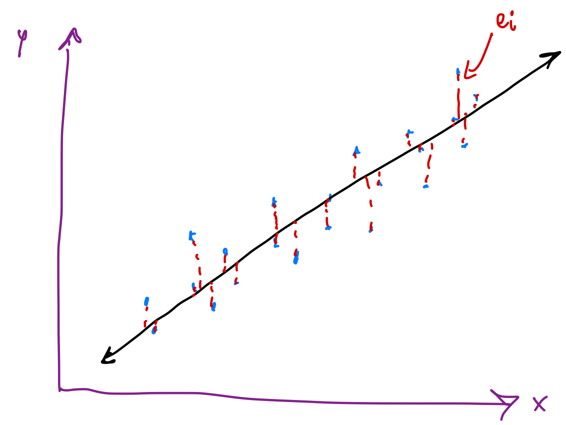
The most common approach involves minimizing the *least squares* criterion.

Let $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ prediction for Y based on i^{th} value of X.

$e_i = y_i - \hat{y}_i$ i^{th} residual

$RSS = e_1^2 + \ldots + e_n^2$ residual sum of squares.

choose $\hat{\beta}_0$ and $\hat{\beta}_1$ to minimize RSS.



using calculus
take derivatives, set = 0, solve for $\hat{\beta}_0$ and $\hat{\beta}_1$

The least squares approach results in the following estimates:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

"least squares coefficients"

previously used lm()

We can get these estimates using the following commands in `R` and `tidymodels`:

```
library(tidymodels) ## load library

## load the data in
ads <- read_csv("../data/Advertising.csv", col_select = -1)

## fit the model
lm_spec <- linear_reg() |>
  set_mode("regression") |>
  set_engine("lm")
```

Model specification

↰ least squares.

```
slr_fit <- lm_spec |>
  fit(sales ~ TV, data = ads)
```

Y~X formula "regress Y on X"

↰ specify data frame "training data"

```
slr_fit |>
  pluck("fit") |>
  summary()
```

exactly the same as result of calling lm(sales ~ TV, data = ads)

```
##
## Call:
## stats::lm(formula = sales ~ TV, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0$ $\hat{\beta}_1$

1.2 Assessing Accuracy

Recall we assume the *true* relationship between X and Y takes the form

$$Y = f(X) + \varepsilon,$$ f is unknown, ε mean-zero random error.

If f is to be approximated by a linear function, we can write this relationship as

population regression line.

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

average increase in Y associated w/ 1 unit increase in X.

catch-all for what we miss w/ this simple model – true relationship may not be linear; may be other important variables not included; measurement error.

expected value of Y when $X = 0$

and when we fit the model to the training data, we get the following estimate of the *population model*

least squares line.

$$Y = \hat{\beta}_0 + \hat{\beta}_1 X$$

But how close this this to the truth? measure w/ standard error

$$\text{Var}(\hat{\beta}_0) = SE(\hat{\beta}_0)^2 = \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right]$$

$$\text{Var}(\hat{\beta}_1) = SE(\hat{\beta}_1)^2 = \sigma^2\left[\frac{1}{\sum_{i=1}^n (x_i - \bar{x})^2}\right]$$

In general, σ^2 $= \text{Var}(\varepsilon)$. is not known, so we estimate it with the *residual standard error*, $RSE = \sqrt{RSS/(n-2)}$.

residual sum of squares

We can use these standard errors to compute confidence intervals and perform hypothesis tests.

approx normality of $\hat{\beta}_1$

$$95\% \text{ CI for } \beta_1: \quad \hat{\beta}_1 \pm 2\,SE(\hat{\beta}_1)$$

$$95\% \text{ CI for } \beta_0: \quad \hat{\beta}_0 \pm 2\,SE(\hat{\beta}_0).$$

Hypothesis tests:

H_0: There is no relationship btw/ X and Y $\iff$ $H_0: \beta_1 = 0$

H_a: There is a relationship btw/ X and Y $\quad H_a: \beta_1 \neq 0$

?: is $\hat{\beta}_1$ far enough away from 0 to be confident it is nonzero?

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)} \sim t_{n-2}$$ (under H_0) $\Rightarrow$ compute P(observing any number equal to or more extreme than $|t|$) = p-value

small p-value means highly unlikely to see this t given H_0

$\Rightarrow$ reject H_0

Once we have decided that there is a significant linear relationship between X and Y that is captured by our model, it is natural to ask

> To what extent does the model fit the data?

The quality of the fit is usually measured by the *residual standard error* and the R^2 statistic.

RSE: Roughly speaking, the RSE is the average amount that the response will deviate from the true regression line. This is considered a measure of the *lack of fit* of the model to the data.

R^2: The RSE provides an absolute measure of lack of fit, but is measured in the units of Y. So, we don't know what a "good" RSE value is! R^2 gives the proportion of variation in Y explained by the model. i.e. will always be between 0 and 1.

Advertising data example.

```r
slr_fit |>
  pluck("fit") |>
  summary()
```

```
##
## Call:
## stats::lm(formula = sales ~ TV, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## TV          0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0, \hat{\beta}_1$ SE($\hat{\beta}_0$), SE($\hat{\beta}_1$) $H_0: \beta_i = 0$ vs. $H_a: \beta_i \neq 0$ $i = 0, 1$.

RSE

revisit later.

R^2 = proportion of variability in Y explained by a linear relationship w/ X

2 Multiple Linear Regression

Simple linear regression is useful for predicting a response based on one predictor variable, but we often have **more than one** predictor.

How can we extend our approach to accommodate additional predictors?

- We could run separate SLR for each predictor

But! How to make a single prediction for y based on levels of all predictors?
Also each model would ignore the other predictors ... what if they are related?
→ misleading results.

Solution: We can give each predictor a separate slope coefficient in a single model.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$$

We interpret β_j ($j \neq 0$) as the "average effect on Y of a one unit increase in X_j, *holding all other predictors fixed*".

In our `Advertising` example,

$$\text{Sales} = \beta_0 + \beta_1 \text{TV} + \beta_2 \text{radio} + \beta_3 \text{newspaper} + \varepsilon$$

2.1 Estimating the Coefficients

As with the case of simple linear regression, the coefficients $\beta_0, \beta_1, \dots, \beta_p$ are unknown and must be estimated. Given estimates $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$, we can make predictions using the formula

using training data

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p.$$

now instead of a line, we are fitting a hyperplane.

The parameters are again estimated using the same least squares approach that we saw in the context of simple linear regression.

alternatively

```
# mlr_fit <- lm_spec |> fit(sales ~ TV + radio + newspaper, data = ads)
mlr_fit <- lm_spec |>
  fit(sales ~ ., data = ads)
```

linear model specification from before.

formula

y ~ every other column in data frame.

pull out fit object

```
mlr_fit |>
  pluck("fit") |>
  summary()
```

```
##
## Call:
## stats::lm(formula = sales ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_0$ $\hat{\beta}_1$ $\hat{\beta}_2$ $\hat{\beta}_3$

2.2 Some Important Questions

When we perform multiple linear regression we are usually interested in answering a few important questions:

1. Is at least one of the predictors $X_1, \dots, X_p$ useful in predicting the response?

2. Do all predictors help explain Y, or only a subset is useful?

3. How well does model fit data?

4. Given a set of predictors values what response should we predict and how accurate is our prediction?

2.2.1 Is there a relationship between response and predictors?

We need to ask whether all of the regression slope coefficients are zero, which leads to the following hypothesis test.

$$H_0: \beta_1 = \beta_2 = \dots = \beta_p = 0$$

$$H_a: \text{at least one } \beta_j \text{ is non-zero.}$$

This hypothesis test is performed by computing the F-statistic

variance in response explained by model

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)} \sim F_{p, n-p-1} \text{ under } H_0$$

variance unexplained by model.

If this ratio F is large (much larger than 1), evidence against the null H_0, i.e. evidence there is some relationship.

2.2.2 Deciding on Important Variables

After we have computed the F-statistic and concluded that there is a relationship between predictor and response, it is natural to wonder

> Which predictors are related to the response?

We could look at the p-values on the individual coefficients, but if we have many variables this can lead to false discoveries.

Instead we could consider *variable selection*. We will revisit this in Ch. 6.

forward selection, backward selection, # Lasso #

2.2.3 Model Fit

Two of the most common measures of model fit are the RSE and R^2. These quantities are computed and interpreted in the same way as for simple linear regression.

Be careful with using these alone, because R^2 will **always increase** as more variables are added to the model, even if it's just a small increase.

How to avoid overfitting? Use test data! Ch. 5

```r
# model with TV, radio, and newspaper
mlr_fit |> pluck("fit") |> summary()
```

```
##
## Call:
## stats::lm(formula = sales ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## radio        0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

individual slopes hypothesis test
$H_0: \beta_j = 0$
$H_a: \beta_j \neq 0$ $j = 0, \dots, p$

R^2

89.72% of variability in Y is captured by this linear model.

F-test
$H_0: \beta_1 = \cdots = \beta_p = 0$
$H_a: \beta_j \neq 0$ $j \in \{1, \dots, p\}$

$F \sim F_{3,196}$ under H_0

prob of seeing F stat value as or more extreme than 570.3 if H_0 true.

```r
# model without newspaper
lm_spec |> fit(sales ~ TV + radio, data = ads) |>
  pluck("fit") |> summary()
```

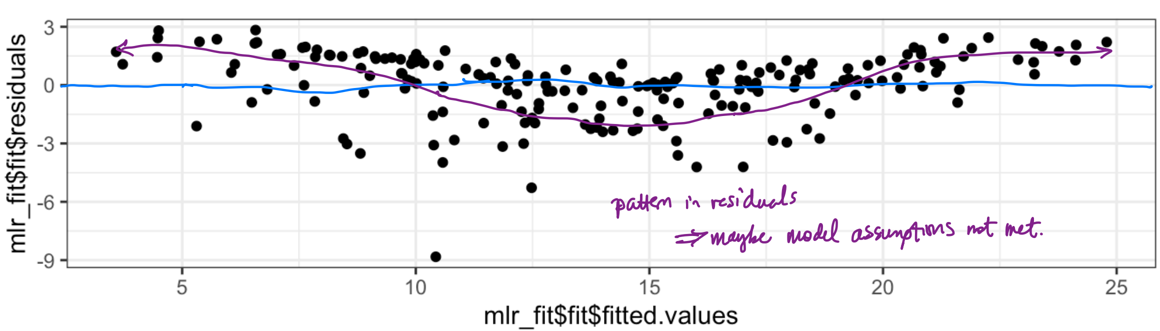
```
##
## Call:
## stats::lm(formula = sales ~ TV + radio, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7977 -0.8752  0.2422  1.1708  2.8328
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.92110    0.29449   9.919   <2e-16 ***
## TV           0.04575    0.00139  32.909   <2e-16 ***
## radio        0.18799    0.00804  23.382   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.681 on 197 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
## F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

R^2 barely decreased when we took out newspaper ⇒ not contributing much.

It may also be useful to plot residuals to get a sense of the model fit.

$e_i = y_i - \hat{y}_i$

```r
ggplot() +
  geom_point(aes(mlr_fit$fit$fitted.values, mlr_fit$fit$residuals))
```



Want: random noise around zero, no pattern.

QQ plot
residuals vs. each predictor

3 Other Considerations

3.1 Categorical Predictors

predictor + response

So far we have assumed all variables in our linear model are quantitiative. what to do when X_i categorical?

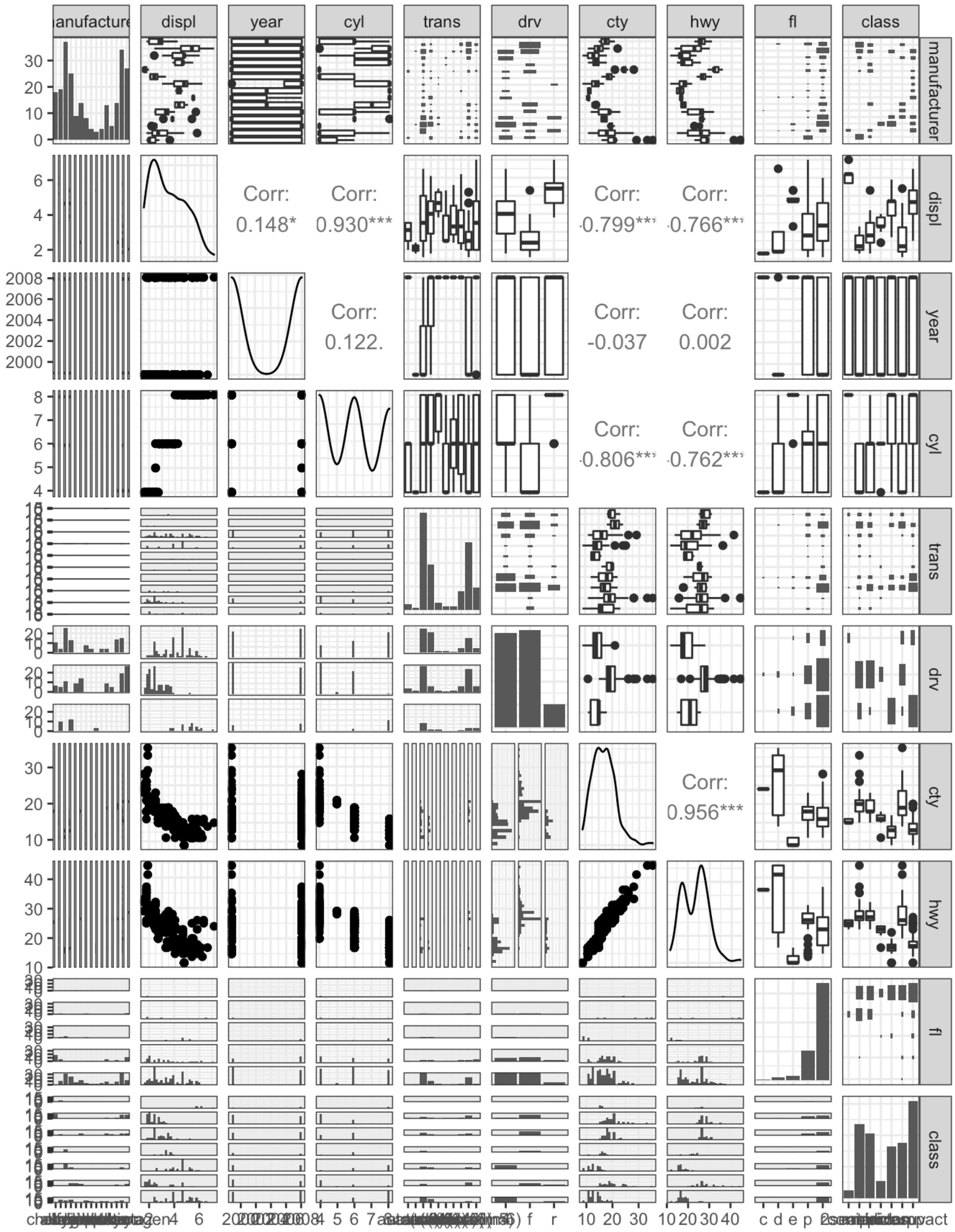
For example, consider building a model to predict highway gas mileage from the `mpg` data set.

```
head(mpg)
```

```
## # A tibble: 6 × 11
##   manufacturer model displ  year   cyl trans      drv     cty   hwy
fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int>
<chr> <chr>
## 1 audi         a4      1.8  1999     4 auto(l5)   f        18    29
p     compa…
## 2 audi         a4      1.8  1999     4 manual(m5) f        21    29
p     compa…
## 3 audi         a4      2    2008     4 manual(m6) f        20    31
p     compa…
## 4 audi         a4      2    2008     4 auto(av)   f        21    30
p     compa…
## 5 audi         a4      2.8  1999     6 auto(l5)   f        16    26
p     compa…
## 6 audi         a4      2.8  1999     6 manual(m5) f        18    26
p     compa…
```

```
library(GGally)

mpg %>%
  select(-model) %>% # too many models
  ggpairs() # plot matrix
```

makes $\frac{p(p-1)}{2}$ plots to look at each pair of variables in data frame.


anufacturer
displ
year
cyl
trans
drv
cty
hwy
fl
class
manufacturer
Corr: 0.148*
Corr: 0.930***
Corr: -0.799***
Corr: -0.766***
Corr: 0.122.
Corr: -0.037
Corr: 0.002
Corr: -0.806***
Corr: -0.762***
Corr: 0.956***
c d e p
f r
10 20 30
10 20 30 40

To incorporate these categorical variables into the model, we will need to introduce $k-1$ dummy variables, where $k =$ the number of levels in the variable, for each qualitative variable.

For example, for `drv`, we have 3 levels: `4`, `f`, and `r`. $k=3$

$$x_{i1} = \begin{cases} 1 & \text{if } i^{th} \text{ car is FWD} \\ 0 & \text{if } i^{th} \text{ car is not FWD} \end{cases}$$

$$x_{i2} = \begin{cases} 1 & \text{if } i^{th} \text{ car is RWD} \\ 0 & \text{if } i^{th} \text{ car is not RWD} \end{cases}$$

$$y_i = \beta_0 + \boxed{\beta_1} x_{i1} + \boxed{\beta_2} x_{i2} + \varepsilon_i = \begin{cases} \beta_0 + \beta_1 + \varepsilon_i & \text{if } i^{th} \text{ car is FWD} \\ \beta_0 + \beta_2 + \varepsilon_i & \text{if } i^{th} \text{ car is RWD} \\ \beta_0 + \varepsilon_i & \text{if } i^{th} \text{ car is 4WD} \end{cases}$$

$\beta_0 =$ avg hwy mpg for 4WD cars

$\beta_1 =$ difference in average hwy btw/ FWD and 4WD cars.

$\beta_2 =$ difference in average hwy btw/ RWD and 4WD cars.

quantitative variables.

```
lm_spec |>
  fit(hwy ~ displ + cty + drv, data = mpg) |>
  pluck("fit") |>
  summary()
```

categorical variable

```
##
## Call:
## stats::lm(formula = hwy ~ displ + cty + drv, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6499 -0.8764 -0.3001  0.9288  4.8632
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.42413    1.09313   3.132  0.00196 **
## displ       -0.20803    0.14439  -1.441  0.15100
## cty          1.15717    0.04213  27.466  < 2e-16 ***
## drvf         2.15785    0.27348   7.890 1.23e-13 ***
## drvr         2.35970    0.37013   6.375 9.95e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.49 on 229 degrees of freedom
## Multiple R-squared:  0.9384, Adjusted R-squared:  0.9374
## F-statistic: 872.7 on 4 and 229 DF,  p-value: < 2.2e-16
```

3.2 Extensions of the Model

The standard regression model provides interpretable results and works well in many problems. However it makes some very strong assumptions that may not always be reasonable.

* linear relationship
constant error variance
uncorrelated errors w/ X

Additive Assumption

The additive assumption assumes that the effect of each predictor on the response is not affected by the value of the other predictors. What if we think the effect should depend on the value of another predictor?

interaction term.

```
lm_spec |>
  fit(sales ~ TV + radio + TV*radio, data = ads) |>
  pluck("fit") |>
  summary()
```

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$
$$= \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \varepsilon$$
$$= \beta_0 + \beta_1 X_1 + (\beta_2 + \beta_3 X_1) X_2 + \varepsilon$$

```
##
## Call:
## stats::lm(formula = sales ~ TV + radio + TV * radio, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3366 -0.4028  0.1831  0.5948  1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.750e+00  2.479e-01  27.233   <2e-16 ***
## TV          1.910e-02  1.504e-03  12.699   <2e-16 ***
## radio       2.886e-02  8.905e-03   3.241   0.0014 **
## TV:radio    1.086e-03  5.242e-05  20.727   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic:  1963 on 3 and 196 DF,  p-value: < 2.2e-16
```

β_3 significantly different from 0.

significant relationship

$R^2 = .89$ without interaction
big increase ⇒ better-fitting model.

"an increase of \$1000 in radio advertising will be associated w/ an expected increase in sales of $(\hat{\beta}_2 + \hat{\beta}_3 * TV) \times 1000 = 29 + 1.1\,TV$"

* If we include interaction term be sure to keep the original variables, otherwise interpretation is confusing.

Alternatively:

"recipe" ↳ plan to fit model.

interaction step transformation of matrix $\underline{X}$

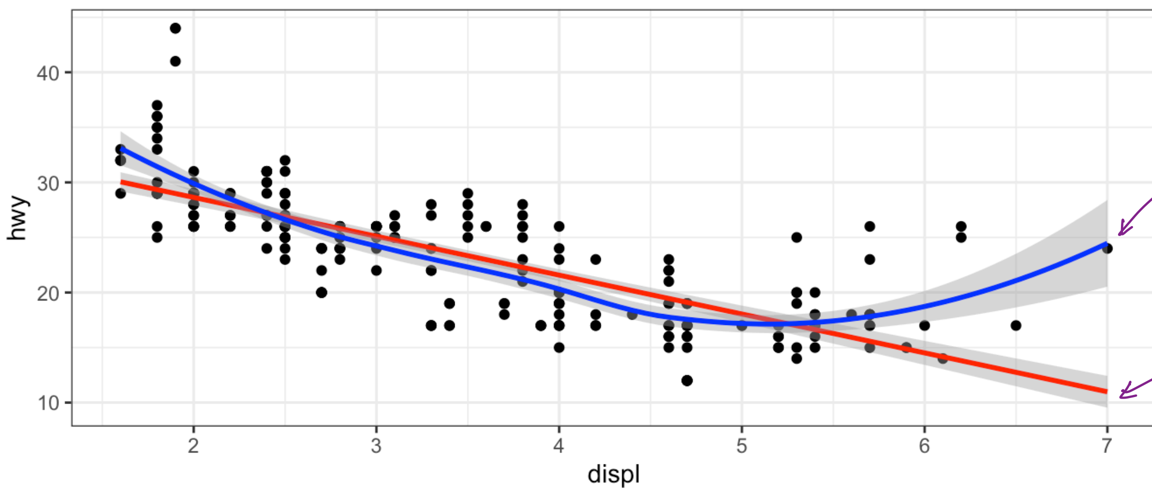
```
rec_spec_interact <- recipe(sales ~ TV + radio, data = ads) |>
  step_interact(~ TV:radio)

lm_wf_interact <- workflow() |>
  add_model(lm_spec) |>
  add_recipe(rec_spec_interact)

lm_wf_interact |> fit(ads)
```

```
## ══ Workflow [trained] ══════════════════════════════════════════════════════════
## Preprocessor: Recipe
## Model: linear_reg()
##
## ── Preprocessor ────────────────────────────────────────────────────────────────
## 1 Recipe Step
##
## • step_interact()
##
## ── Model ───────────────────────────────────────────────────────────────────────
##
## Call:
## stats::lm(formula = ..y ~ ., data = data)
##
## Coefficients:
## (Intercept)           TV        radio    TV_x_radio
##    6.750220     0.019101     0.028860      0.001086
```

Linearity Assumption

The linear regression model assumes a linear relationship between response and predictors. In some cases, the true relationship may be non-linear.

```
ggplot(data = mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth(method = "lm", colour = "red") +
  geom_smooth(method = "loess", colour = "blue")
```



nonparam smoothed fit

linear fit

How to include non-linear terms in the model?

"Identity"

```
lm_spec |>
  fit(hwy ~ displ + I(displ^2), data = mpg) |>
  pluck("fit") |> summary()
```

```
##
## Call:
## stats::lm(formula = hwy ~ displ + I(displ^2), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.6258 -2.1700 -0.7099  2.1768 13.1449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  49.2450     1.8576  26.510  < 2e-16 ***
## displ       -11.7602     1.0729 -10.961  < 2e-16 ***
## I(displ^2)    1.0954     0.1409   7.773 2.51e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.423 on 231 degrees of freedom
## Multiple R-squared:  0.6725, Adjusted R-squared:  0.6696
## F-statistic: 237.1 on 2 and 231 DF,  p-value: < 2.2e-16
```

Significant.

Be careful throwing higher order terms → can lead to overfitting & very bad predictions on edges of the space.

Will come back to a better way to do this later.

3.3 Potential Problems

1. Non-linearity of response-predictor relationships

   diagnosis:
   plot residuals vs. fitted values → or each predictor
   see pattern.

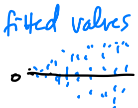
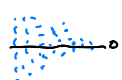
   solution
   - add polynomial term
   - transform predictor

2. Correlation of error terms

   diagnosis:
   understanding how data is collected.
   time series? spatial data?

   solution
   * use models formulated for correlated errors (not this class)
   - include a predictor to capture dependence.

3. Non-constant variance of error terms

   diagnosis
   plot residuals vs. fitted values
   see funnel pattern  or 

   solution
   transform Y. Try log Y or $\sqrt{y}$

4. Outliers

   diagnosis
   plot data

   solution
   Is your data wrong? i.e. error in data collection?
   fix it.
   otherwise - maybe you are missing a predictor?

4 K-Nearest Neighbors

In Ch. 2 we discuss the differences between *parametric* and *nonparametric* methods. Linear regression is a parametric method because it assumes a linear functional form for $f(X)$.

easy to fit
easy to interpret
can do hypothesis testing.

make strong assumptions, what if they are wrong?
- parametric model will perform poorly.

A simple and well-known non-parametric method for regression is called K-nearest neighbors regression (KNN regression).

\# of neighbors

Given a value for K and a prediction point x_0, KNN regression first identifies the K training observations that are closest to x_0 ($\mathcal{N}_0$). It then estimates $f(x_0)$ using the average of all the training responses in $\mathcal{N}_0$,

neighborhood

$$\hat{f}(x_0) = \frac{1}{K}\sum_{x_i \in \mathcal{N}_0} y_i$$

training data

value to be predicted.

```r
set.seed(445) #reproducibility

## generate data
x <- rnorm(100, 4, 1) # pick some x values
y <- 0.5 + x + 2*x^2 + rnorm(100, 0, 2) # true relationship
df <- data.frame(x = x, y = y) # data frame of training data

for (k in seq(2, 10, by = 2)) {
  nearest_neighbor(mode = "regression", neighbors = k) |>
    fit(y ~ x, data = df) |>
    augment(new_data = df) |>
    ggplot() +
    geom_point(aes(x, y)) +
    geom_line(aes(x, .pred), colour = "red") +
    ggtitle(paste("KNN, k = ", k)) +
    theme(text = element_text(size = 30)) -> p

  print(p)
}

lm_spec |>
  fit(y ~ x, df) |>
  augment(new_data = df) |>
  ggplot() +
    geom_point(aes(x, y)) +
    geom_line(aes(x, .pred), colour = "red") +
```

generate fake data

K = 2, 4, 6, 8, 10

model specification

fit model

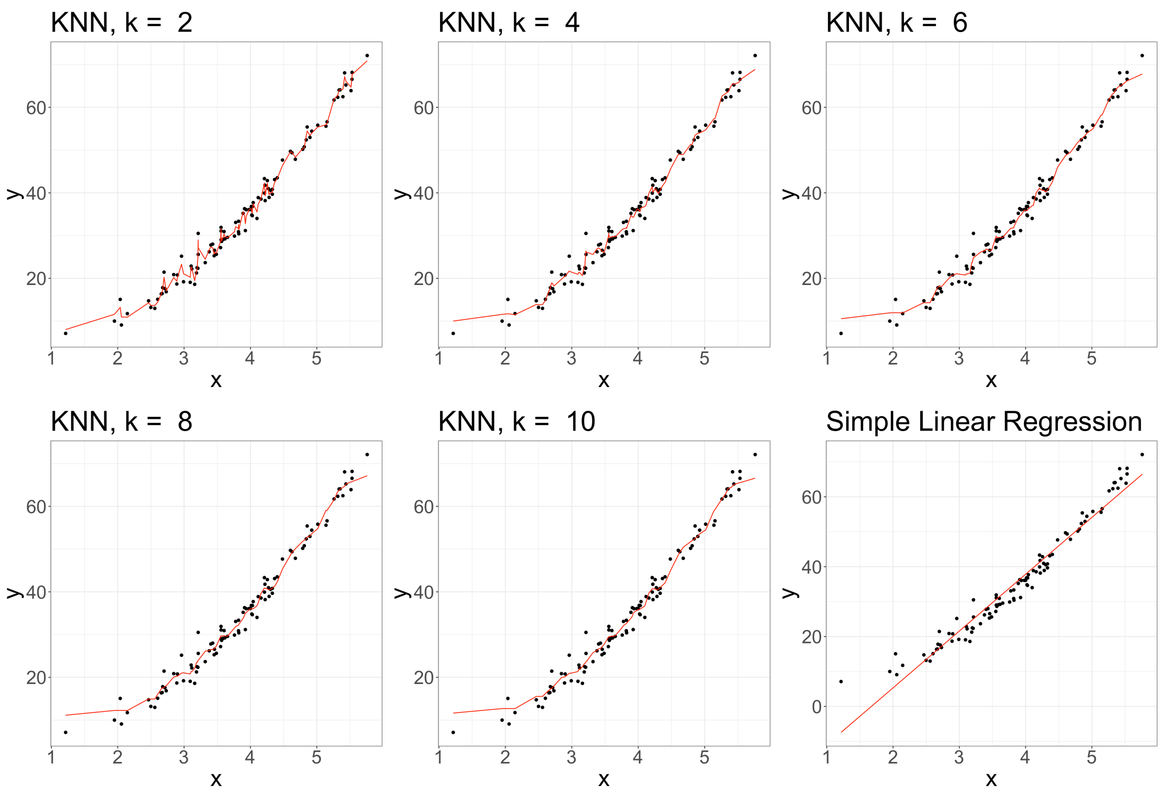
formula just like before.

data

add predicted values + residuals. to the data frame "new_data"

compare to linear model

```
ggtitle("Simple Linear Regression") +
theme(text = element_text(size = 30)) # slr plot
```



as k↑, KNN fitted line gets smoother

↑ missing quadratic relationship.