Data Analysis
Week 8: Model Parameter Inference and Model Selection

1 Introduction

In Week 7 we began to consider the construction and use of confidence intervals (CIs) for the population parameters listed in Table 1 (reproduced below). In particular, we used bootstrap methods to estimate the sampling distributions of the estimates in Scenarios 1-4 and used these to construct CIs for the corresponding population parameters.

Table 1: Scenarios of sample statistics for inference.

		Population		Sample
Scenario	Population Parameter	Notation	Sample Statistic	Notation
1	Population proportion	p	Sample proportion	$\widehat{p}$
2	Population mean	μ	Sample mean	$\bar{x}$
3	Diff.in pop. props	$p_1 - p_2$	Diff. in sample props	$\widehat{p}_1 - \widehat{p}_2$
4	Diff. in pop. means	$\mu_1 - \mu_2$	Diff. in sample means	$\bar{x}_1 - \bar{x}_2$
5	Pop. intercept	β_0	Sample intercept	$\widehat{\beta}_0$ or b_0
6	Pop. slope	β_1	Sample slope	$\widehat{\beta}_1$ or b_1

This week we continue this process for Scenarios 5 and 6, namely construct CIs for the parameters in simple and multiple linear regression models. We will start with bootstrap methods and also consider CIs based on theoretical results when standard assumptions hold. We will also consider how to use CIs for variable selection and finish by considering a model selection strategy based on objective measures for model comparisons.

Now that you are familiar with RMarkdown, you are encouraged to collate your work in this tutorial in an RMarkdown file. Create a `.Rmd` file to load the following packages into R:

```r
library(dplyr)
library(ggplot2)
library(janitor)
library(moderndive)
library(infer)
library(broom)
```

Note: Additional information and examples can be found in Chapter 11 of An Introduction to Statistical and Data Science via R.

2 Confidence Intervals for Regression Parameters

2.1 Bootstrap Confidence Intervals for the slope β in Simple Linear Regression (SLR)

Just as we did for Scenarios 1-4 in Table 1 in Week 7, we can use the infer package to repeatedly sample from a data set to estimate the sampling distribution and standard error of the estimates of the intercept $(\widehat{\alpha})$ and the covariate's parameter $\left(\widehat{\beta}\right)$ in the simple linear regression model $y_i = \alpha + \beta x_i$. These sampling distributions enable us to directly find bootstrap confidence intervals for the model parameters. Usually, interest lies in β and so that will be our focus here.

To illustrate this, let's return to the teaching evaluations data evals in the moderndive package that we analyzed last week and start with the SLR model with age as the the single explanatory variable and the instructors' evaluation scores as the response variable. This data and the fitted model are shown here.

```
slr.model <- lm(score ~ age, data = evals)
coeff <- slr.model %>%
          coef()
```

```
 (Intercept)          age
 4.461932354 -0.005938225
```

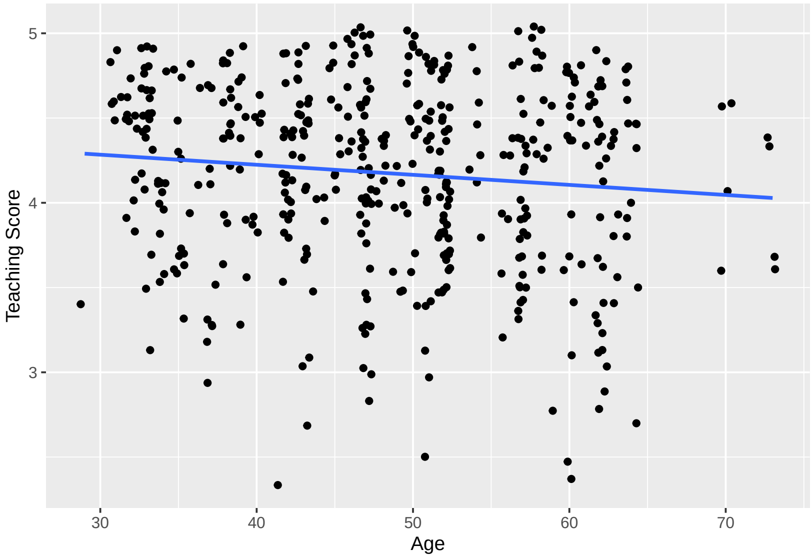
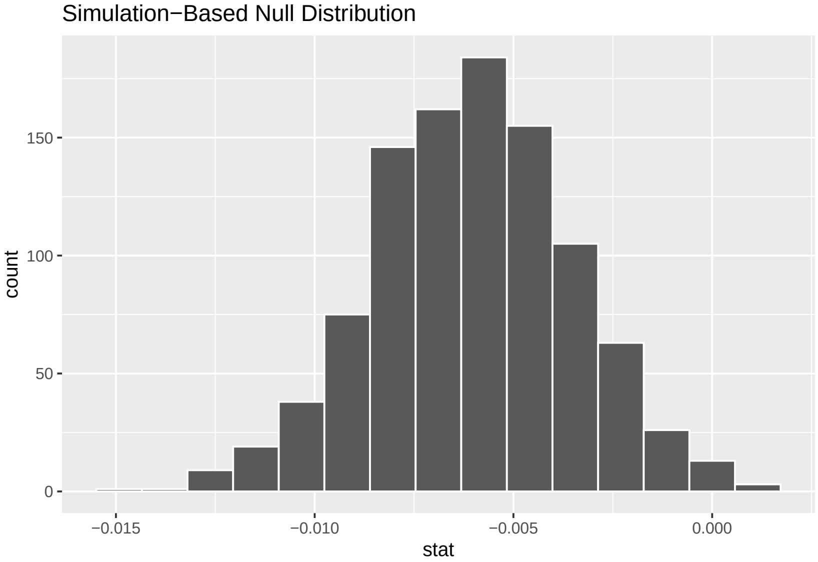


Figure 1: Figure 1: SLR model applied to the teaching evaluation Data.

The point estimate of the slope parameter here is $\widehat{\beta}$ = -0.006. The following code estimates the sampling distribution of $\widehat{\beta}$ via the bootstrap method.

```
bootstrap_beta_distn <- evals %>%
                        specify(score ~ age) %>%
                        generate(reps = 1000, type = "bootstrap") %>%
                        calculate(stat = "slope")

bootstrap_beta_distn %>%
  visualize()
```



Now we can use the `get_ci` function to calculate a 95% confidence interval. We can do this in two different ways. Remember that these denote a range of plausible values for an unknown true population slope parameter regressing teaching `score` on `age`.

```
percentile_beta_ci <- bootstrap_beta_distn %>%
                      get_ci(level = 0.95, type = "percentile")
```

```
# A tibble: 1 x 2
   `2.5%`  `97.5%`
    <dbl>    <dbl>
1 -0.0111 -0.00104
```

```
se_beta_ci <- bootstrap_beta_distn %>%
              get_ci(level = 0.95, type = "se", point_estimate = coeff[2])
```

```
# A tibble: 1 x 2
    lower     upper
    <dbl>     <dbl>
1 -0.0109 -0.000984
```

Using the 2.5% and 97.5% percentiles of the simulated bootstrap sampling distribution the 95% confidence interval is (-0.011, -0.001) and the 95% confidence interval using the standard deviation of the sampling distribution (i.e. estimated standard error of $\widehat{\beta}$) is (-0.011, -0.001). With the bootstrap distribution being close to symmetric, it makes sense that the two resulting confidence intervals are similar.

2.2 Confidence Intervals for the parameters in Multiple Regression

Let's continue with the teaching evaluations data by fitting the multiple regression model with one numerical and one categorical explanatory variable that we first saw in Week 7. In this model:

- y: response variable of instructor evaluation `score`
- explanatory variables
  - x_1: numerical explanatory variable of `age`
  - x_2: categorical explanatory variable of `gender`

```
evals_multiple <- evals %>%
                  select(score, gender, age)
```

First, recall that we had two competing potential models to explain professors' teaching evaluation scores:

1. Model 1: Parallel lines model (no interaction term) - both male and female professors have the same slope describing the associated effect of age on teaching score
2. Model 2: Interaction model - allowing for male and female professors to have different slopes describing the associated effect of age on teaching score

Refresher: Visualisations

Recall the plots we made for both these models:

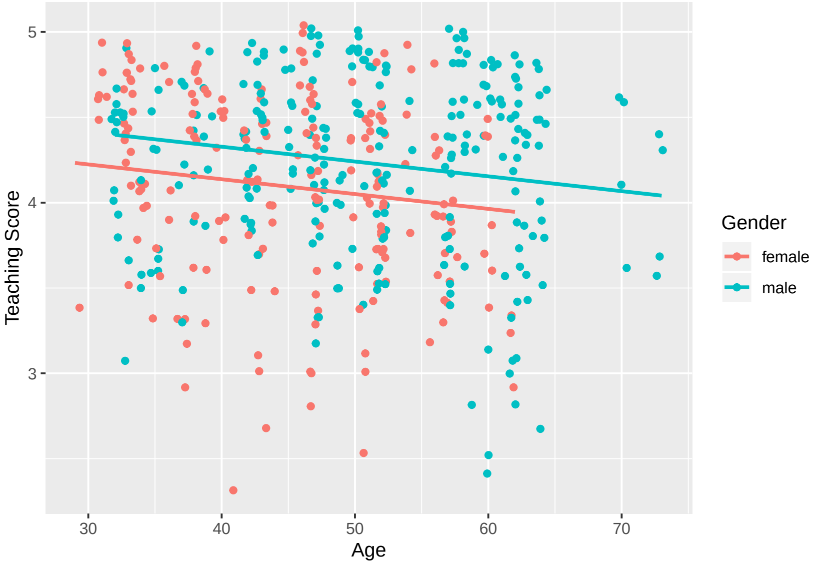


Figure 2: Model 1: Parallel regression lines.

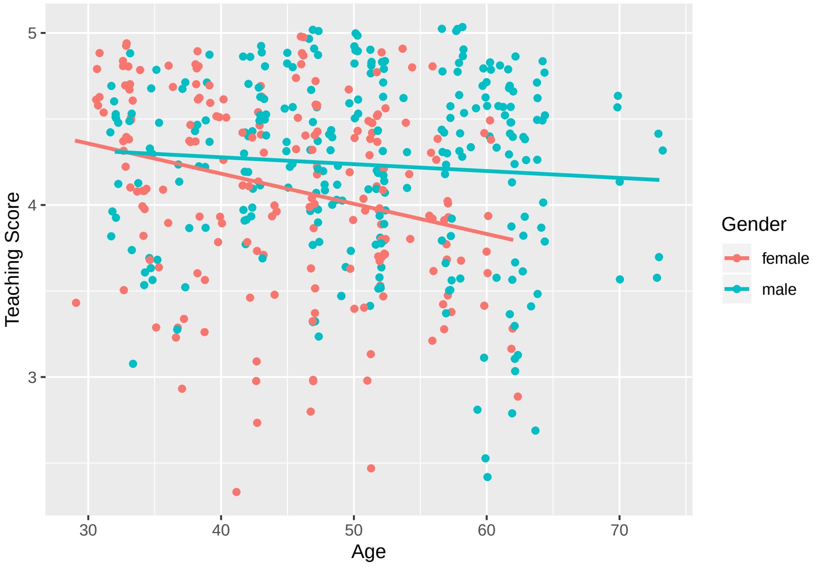


Figure 3: Model 2: Separate regression lines.

Refresher: Regression tables

Let's also recall the regression models. First, the regression model with no interaction effect: note the use of + in the formula.

```r
par.model <- lm(score ~ age + gender, data = evals_multiple)

get_regression_table(par.model) %>%
  knitr::kable(
               digits = 3,
               caption = "Model 1: Regression model with no interaction effect included.",
               booktabs = TRUE
        )
```

Table 2: Model 1: Regression model with no interaction effect included.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.484	0.125	35.792	0.000	4.238	4.730
age	-0.009	0.003	-3.280	0.001	-0.014	-0.003
gendermale	0.191	0.052	3.632	0.000	0.087	0.294

Second, the regression model with an interaction effect: note the use of `*` in the formula.

```r
int.model <- lm(score ~ age * gender, data = evals_multiple)

get_regression_table(int.model) %>%
  knitr::kable(
                digits = 3,
                caption = "Model 2: Regression model with interaction effect included.",
                booktabs = TRUE
      )
```

Table 3: Model 2: Regression model with interaction effect included.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.883	0.205	23.795	0.000	4.480	5.286
age	-0.018	0.004	-3.919	0.000	-0.026	-0.009
gendermale	-0.446	0.265	-1.681	0.094	-0.968	0.076
age:gendermale	0.014	0.006	2.446	0.015	0.003	0.024

Notice that, together with the estimated parameter values, the tables include other information about each estimated parameter in the model, namely:

- **std_error**: the standard error of each parameter estimate;
- **statistic**: the test statistic value used to test the null hypothesis that the population parameter is zero;
- **p_value**: the *p* value associated with the test statistic under the null hypothesis; and
- **lower_ci** and **upper_ci**: the lower and upper bounds of the 95% confidence interval for the population parameter

These values are calculated using the theoretical results based on the standard assumptions that you will have seen in *Regression Modelling* in first semester. Theses values are **not** based on bootstrapping techniques since these become much harder to implement when working with multiple variables and its beyond the scope of this course.

3 Inference using Confidence Intervals

Having described several ways of calculating confidence intervals for model parameters, we are now in a position to interpret them for the purposes of statistical inference.

Simple Linear Regression: $\widehat{y}_i = \alpha + \beta x_i$

Whether we have obtained a confidence interval for β in a simple linear regression model via bootstrapping or theoretical results based on assumptions, the interpretation of the interval is the same. As we saw in Week 7,

> A confidence interval gives a range of plausible values for a population parameter.

We can therefore use the confidence interval for β to state a range of plausible values and, just as usefully, what values are **not** plausible. The most common value to compare the confidence interval of β with is 0 (zero), since $\beta = 0$ says there is *no* (linear) relationship between the response variable (y) and the explanatory variable (x). Therefore, if 0 lies within the confidence interval for β then there is insufficient evidence of a linear relationship between y and x. However, if 0 **does not** lie within the confidence interval, then we conclude that β is significantly different from zero and therefore that there is evidence of a linear relationship between y and x.

Let's use the confidence interval based on theoretical results for the slope parameter in the SLR model applied to the teacher evaluation scores with `age` as the the single explanatory variable and the instructors' evaluation `score`s as the outcome variable.

```r
get_regression_table(slr.model) %>%
          knitr::kable(
          digits = 3,
          caption = "Estimates from the SLR model of `score` on `age`.",
          booktabs = TRUE
          )
```

Table 4: Estimates from the SLR model of `score` on `age`.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.462	0.127	35.195	0.000	4.213	4.711
age	-0.006	0.003	-2.311	0.021	-0.011	-0.001

Multiple Regression

Consider, again, the fitted interaction model for `score` with `age` and `gender` as the two explanatory variables.

```r
int.model <- lm(score ~ age * gender, data = evals_multiple)
get_regression_table(int.model)
```

Table 5: Model 2: Regression model with interaction effect included.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.883	0.205	23.795	0.000	4.480	5.286
age	-0.018	0.004	-3.919	0.000	-0.026	-0.009
gendermale	-0.446	0.265	-1.681	0.094	-0.968	0.076
age:gendermale	0.014	0.006	2.446	0.015	0.003	0.024

4 Variable selection using confidence intervals

When there is more than one explanatory variable in a model, the parameter associated with each explanatory variable is interpreted as the change in the mean response based on a 1-unit change in the corresponding explanatory variable **keeping all other variables held constant**. Therefore, care must be taken when interpreting the confidence intervals of each parameter by acknowledging that each are plausible values **conditional on all the other explanatory variables in the model**.

Because of the interdependence between the parameter estimates and the variables included in the model, choosing which variables to include in the model is a rather complex task. We will introduce some of the ideas in the simple case where we have 2 potential explanatory variables (x_1 and x_2) and use confidence intervals to decide which variables will be useful in predicting the response variable (y).

One approach is to consider a hierarchy of models:

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i}$$

$$y_i = \alpha + \beta_1 x_{1i} \qquad\qquad y_i = \alpha + \beta_2 x_{2i}$$

$$y_i = \alpha$$

Within this structure we might take a top-down approach:

1. Fit the most general model, i.e. $y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i}$ since we believe this is likely to provide a good description of the data
2. Construct confidence intervals for β_1 and β_2
   (a) If both intervals exclude 0 then retain the model with both x_1 and x_2.
   (b) If the interval for β_1 contains 0 but that for β_2 does not, fit the model with x_2 alone.
   (c) If the interval for β_2 contains 0 but that for β_1 does not, fit the model with x_1 alone.
   (d) If both intervals include 0 it may still be that a model with one variable is useful. In this case the two models with the single variables should be fitted and intervals for β_1 and β_2 constructed and compared with 0.

If we have only a few explanatory variables, then an extension of the strategy outlined above would be effective, i.e. start with the full model and simplify by removing terms until no further terms can be removed. When the number of explanatory variables is large the problem becomes more difficult. We will consider this more challenging situation in the next section.

Recall that as well as `age` and `gender`, there is also a potential explanatory variable `bty_avg` in the `evals` data, i.e. the numerical variable of the average beauty score from a panel of six students' scores between 1 and 10. We can fit the multiple regression model with the two continuous explanatory variables `age` and `bty_avg` as follows:

```
mlr.model <- lm(score ~ age + bty_avg, data = evals)
```

Table 6: Estimates from the MLR model with `age` and `bty_avg`.

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	4.055	0.170	23.870	0.000	3.721	4.389
age	-0.003	0.003	-1.148	0.251	-0.008	0.002
bty_avg	0.061	0.017	3.548	0.000	0.027	0.094

5 Model comparisons using objective criteria

As was noted in the last section, when the number of potential explanatory variables is large the problem of selecting which variables to include in the final model becomes more difficult. The selection of a final regression model always involves a compromise:

- Predictive accuracy (improved by including more predictor/explanatory variables)
- Interpretability (achieved by having less predictor/explanatory variables)

There are many objective criteria for comparing different models applied to the same data set. All of them trade off the two objectives above, i.e. fit to the data against complexity. Common examples include:

1. The R^2_{adj} values, i.e. the proportion of the total variation of the response variable explained by the models.

$$R^2_{adj} = 1 - \frac{RSS/(n-p-1)}{SST/(n-1)} = 100 \times \left[1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2/(n-p-1)}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2/(n-1)} \right]$$

- where
  - n is the sample size
  - p is the number of parameters in the model
  - RSS is the residual sum of squares from the fitted model
  - SST is the total sum of squares around the mean response.
- R^2_{adj} values are used for nested models, i.e. where one model is a particular case of the other

2. Akaike's Information Criteria (AIC)

$$AIC = -2 \cdot \text{log-likeihood} + 2p = n \ln \left(\frac{RSS}{n} \right) + 2p$$

- A value based on the maximum likelihood function of the parameters in the fitted model penalised by the number of parameters in the model
- It be used to compare any models fitted to the same response variable
- The smaller the AIC the 'better' the model, i.e. no distributional results are employed to assess differences
- See the `stepAIC` function from the `MASS` library that was mention in Week 6.

3. Bayesian Information Criteria

$$BIC = -2 \cdot \text{log-likeihood} + \ln(n)p$$

A popular data analysis strategy that can be adopted is to calculate R^2_{adj}, AIC and BIC and compare the models which **minimise** the AIC and BIC with the model that **maximises** the R^2_{adj}.

To illustrate this, let's return to the `evals` data and the MLR on the teaching evaluation score `score` with the two continuous explanatory variables `age` and `bty_avg` and compare this with the SLR model with just `bty_avg`. To access these measures for model comparisons we can use the `glance` function in the `broom` package (not to be confused with the `glimpse` function from the `dplyr` package).

```
model.comp.values.slr.age <- glance(lm(score ~ age, data = evals))
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
1    0.0115       0.00931 0.541      5.34  0.0213     2  -372.  750.  762.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
model.comp.values.slr.bty_avg <- glance(lm(score ~ bty_avg, data = evals))
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
1    0.0350        0.0329 0.535      16.7 5.08e-5     2  -366.  738.  751.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

```
model.comp.values.mlr <- glance(lm(score ~ age + bty_avg, data = evals))
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
1    0.0378        0.0336 0.535      9.03 1.42e-4     3  -366.  739.  756.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

Note that R^2_{adj}, AIC and BIC are contained in columns 2, 9 and 10 respectively. To access just these values and combine them in a single table we use:

```
Models <- c('SLR(age)','SLR(bty_avg)','MLR')
bind_rows(model.comp.values.slr.age, model.comp.values.slr.bty_avg,
          model.comp.values.mlr, .id = "Model") %>%
          select(Model, adj.r.squared, AIC, BIC) %>%
          mutate(Model = Models) %>%
          kable(
                digits = 2,
                caption = "Model comparison values for different models.",
          )
```

Table 7: Model comparison values for different models.

Model	adj.r.squared	AIC	BIC
SLR(age)	0.01	749.62	762.03
SLR(bty_avg)	0.03	738.44	750.86
MLR	0.03	739.12	755.67

6 A final word on model selection

A great deal of care should be taken in selecting predictor/explanatory variables for a model because the values of the regression coefficients depend upon the variables included in the model. Therefore, the predictors included and the order in which they are entered into the model can have great impact. In an ideal world, predictors should be selected based on past research and new predictors should be added to existing models based on the theoretical importance of the variables. One thing not to do is select hundreds of random predictors, bung them all into a regression analysis and hope for the best.

But in practice there are automatic strategies, such as **Stepwise** (see Week 6 on stepwise regression using **AIC**) and **Best Subsets** regression, based on systematically searching through the entire list of variables not in the current model to make decisions on whether each should be included. These strategies need to be handled with care, and a proper discussion of them is beyond this course. Our best strategy is a mixture of judgement on what variables should be included as potential explanatory variables, together with an interval estimation and hypothesis testing strategy for assessing these. The judgement should be made in the light of advice from the problem context.

Golden rule for modelling

> The key to modelling data is to only use the objective measures as a rough guide. In the end the choice of model will involve your own judgement. You have to be able to defend why you chose a particular model.

7 Further Tasks

You are encouraged to complete the following tasks by using RMarkdown to produce a single document which summarises all your work, i.e. the original questions, your R code, your comments and reflections, etc.

1. Data was collected on the characteristics of homes in the American city of Los Angeles (LA) in 2010 and can be found in the file `LAhomes.csv` on the Moodle page. The data contain the following variables:

- `city` - the district of LA where the house was located
- `type` - either `SFR` (Single Family Residences) or `Condo/Twh` (Condominium/Town House)
- `bed` - the number of bedrooms
- `bath` - the number of bathrooms
- `garage` - the number of car spaces in the garage
- `sqft` - the floor area of the house (in square feet)
- `pool` - `Y` if the house has a pool
- `spa` - `TRUE` if the house has a spa
- `price` - the most recent sales price ($US)

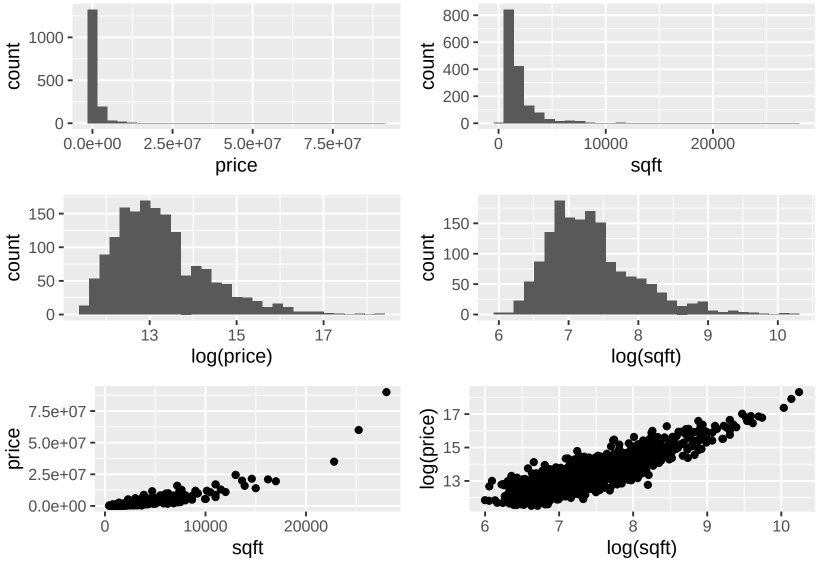
We are interested in exploring the relationships between `price` and the other variables.

Read the data into an object called `LAhomes` and answer the following questions.

a. By looking at the univariate and bivariate distributions on the `price` and `sqft` variables below, what would be a sensible way to proceed if we wanted to model this data? What care must be taken if you were to proceed this way?

```r
library(gridExtra) # Package to display plots side by side

hist1 <- ggplot(LAhomes, aes(x = price)) +
          geom_histogram()

hist2 <- ggplot(LAhomes, aes(x = sqft)) +
          geom_histogram()

hist1log <- ggplot(LAhomes, aes(x = log(price))) +
             geom_histogram()

hist2log <- ggplot(LAhomes, aes(x = log(sqft))) +
             geom_histogram()

plot1 <- ggplot(LAhomes, aes(x = sqft, y = price)) +
          geom_point()

plot2 <- ggplot(LAhomes, aes(x = log(sqft), y = log(price))) +
          geom_point()
```

```r
grid.arrange(hist1, hist2, hist1log, hist2log, plot1, plot2,
             ncol = 2, nrow = 3)
```



b. Fit the simple linear model with `log(price)` as the response and `log(sqft)` as the predictor. Display the fitted model on a scatterplot of the data and construct a bootstrap confidence interval (using the percentiles of the bootstrap distribution) for the slope parameter in the model and interpret its point and interval estimates.

   Hint: Although you can supply the `lm()` function with terms like `log(price)` when you use the **`infer`** package to generate bootstrap intervals the transformed variable needs to already exist. Use the `mutate()` funtion in the `dplyr` package to create new transformed variables.

c. Repeat the analysis in part b. but with the log of the number of bathrooms (`bath`) as the single explanatory variable.

d. Fit the multiple linear regression model using the **log transform of all the variables** `price` (as the response) and both `sqft` and `bath` (as the explanatory variables). Calculate the point and interval estimates of the coefficients of the two predictors separately. Compare their point and interval estimates to those you calculated in parts b. and c. Can you account for the differences?

   Hint: Remember that we didn't use bootstrapping to construct the confidence intervals for parameters in multiple linear regression models, but rather used the theoretical results based on assumptions. You can access these estimates using the `get_regression_table()` function in the `moderndive` package.

e. Using the objective measures for model comparisons, which of the models in parts b., c. and d. would you favour? Is this consistent with your conclusions in part d.?

---

2. You have been asked to determine the pricing of a New York City (NYC) Italian restaurant's dinner menu such that it is competitively positioned with other high-end Italian restaurants by analysing pricing data that have been collected in order to produce a regression model to predict the price of dinner.

   Data from surveys of customers of 168 Italian restaurants in the target area are available. The data can be found in the file `restNYC.csv` on the Moodle page. Each row represents one customer survey from Italian restaurants in NYC and includes the key variables:

- `Price` - price (in $US) of dinner (including a tip and one drink)
- `Food` - customer rating of the food (from 1 to 30)
- `Decor` - customer rating of the decor (from 1 to 30)

- Service - customer rating of the service (from 1 to 30)
- East - dummy variable with the value 1 if the restaurant is east of Fifth Avenue, 0 otherwise

a. Use the ggpairs function in the GGally package (see the following code) to generate an informative set of graphical and numerical summaries which illuminate the relationships between pairs of variables. Where do you see the strongest evidence of relationships between price and the potential explanatory variables? Is there evidence of multicollineatity in the data?

```r
library(GGally) # Package to produce matrix of 'pairs' plots and more!
restNYC$East <- as.factor(restNYC$East) # East needs to be a factor
# Including the `East` factor
ggpairs(restNYC[, 4:8], aes(colour = East, alpha = 0.4))
# Without the `East` factor
ggpairs(restNYC[, 4:7], aes(alpha = 0.4))
```

b. Fit the simple linear model with Price as the response and Service as the predictor and display the fitted model on a scatterplot of the data. Construct a bootstrap confidence interval (using the standard error from the bootstrap distribution) for the slope parameter in the model.

   Now fit a multiple regressing model of Price on Service, Food, and Decor. What happens to the significance of Service when additional variables were added to the model?

c. What is the correct interpretation of the coefficient on Service in the linear model which regresses Price on Service, Food, and Decor?