Chapter 2: simple regression

We start again by loading the data.

```
# Load the datasets from the RData file
load("../dataCreated/listings_clean.RData")

# Display the first few rows of the cleaned dataset
head(listings_clean)
```

```
## # A tibble: 6 x 75
##        id listing_url            scrape_id last_scraped source name  description
##     <dbl> <chr>                      <dbl> <date>       <chr>  <chr> <chr>
## 1   73155 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Apar~ This brigh~
## 2   77592 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Char~ <NA>
## 3  101526 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Apar~ <NA>
## 4  539905 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Sunn~ <NA>
## 5  763422 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ City~ A lovely, ~
## 6 1322782 https://www.airbnb.co~   2.02e13 2024-06-25   city ~ Spac~ <NA>
## # i 68 more variables: neighborhood_overview <chr>, picture_url <chr>,
## #   host_id <dbl>, host_url <chr>, host_name <chr>, host_since <date>,
## #   host_location <chr>, host_about <chr>, host_response_time <chr>,
## #   host_response_rate <chr>, host_acceptance_rate <chr>,
## #   host_is_superhost <lgl>, host_thumbnail_url <chr>, host_picture_url <chr>,
## #   host_neighbourhood <chr>, host_listings_count <dbl>,
## #   host_total_listings_count <dbl>, host_verifications <chr>, ...
```

Filter the data so that we use only listings for at most 6 people. From now on use this

```
listings_clean_filtered <- listings_clean %>%
  filter(accommodates <= 6)
```

Regress price on `review_scores_rating`.

```
# Run a linear regression of price on review_scores_rating
model <- lm(price ~ review_scores_rating, data = listings_clean_filtered)

# Display the summary of the regression model
summary(model)
```

```
##
## Call:
## lm(formula = price ~ review_scores_rating, data = listings_clean_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -104.965  -43.824   -7.964   32.321  163.448
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)             63.92      63.57   1.005    0.315
## review_scores_rating    14.27      13.37   1.067    0.287
##
## Residual standard error: 58.09 on 419 degrees of freedom
## Multiple R-squared:  0.00271,    Adjusted R-squared:  0.0003294
## F-statistic: 1.138 on 1 and 419 DF,  p-value: 0.2866
```

Run a regression of price on how many people can be accommodated.

```
# Run a linear regression of price on accommodates
model_accommodates <- lm(price ~ accommodates, data = listings_clean_filtered)

# Display the summary of the regression model
summary(model_accommodates)
```

```
##
## Call:
## lm(formula = price ~ accommodates, data = listings_clean_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144.65  -33.46  -10.91   22.35  159.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    60.731      5.735   10.59   <2e-16 ***
## accommodates   25.183      1.855   13.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 48.48 on 419 degrees of freedom
## Multiple R-squared:  0.3055, Adjusted R-squared:  0.3038
## F-statistic: 184.3 on 1 and 419 DF,  p-value: < 2.2e-16
```

On slide 21, we have said that the sample moment conditions immediately imply that: 1. the average residual is zero 2. the average covariance between the residual and the regressor is zero 3. the regression line goes through the average of the dependent variable and the regressor.

We now verify this.

```
# 1. Verify that the average residual is zero
residuals_accommodates <- residuals(model_accommodates)
mean_residuals <- mean(residuals_accommodates)
cat("1. Average of residuals:", mean_residuals, "\n")
```

```
## 1. Average of residuals: 2.961103e-15
```

```
# 2. Verify that the average covariance between the residual and the regressor (accommodates) is zero
cov_residuals_regressor <- cov(residuals_accommodates, listings_clean_filtered$accommodates)
cat("2. Covariance between residuals and accommodates:", cov_residuals_regressor, "\n")
```

```
## 2. Covariance between residuals and accommodates: -1.017081e-14
```

```
# 3. Verify that the regression line goes through the averages of the dependent variable and the regres
mean_price <- mean(listings_clean_filtered$price)
mean_accommodates <- mean(listings_clean_filtered$accommodates)

# Calculate the predicted price at the average of accommodates
predicted_price_at_mean <- coef(model_accommodates)[1] + coef(model_accommodates)[2] * mean_accommodate
```

```
cat("3. Average price:", mean_price, "\n")
```

```
## 3. Average price: 131.6746
```

```
cat("   Predicted price at average accommodates:", predicted_price_at_mean, "\n")
```

```
##    Predicted price at average accommodates: 131.6746
```

We have also said that the total sum of squares for the dependent variable is equal to the explained sum of squares plus the residual sum of squares. We can also verify this.

```
# 1. Calculate the Total Sum of Squares (TSS)
mean_price <- mean(listings_clean_filtered$price)
TSS <- sum((listings_clean_filtered$price - mean_price)^2)
cat("Total Sum of Squares (TSS):", TSS, "\n")
```

```
## Total Sum of Squares (TSS): 1417766
```

```
# 2. Calculate the Residual Sum of Squares (RSS)
RSS <- sum(residuals_accommodates^2)
cat("Residual Sum of Squares (RSS):", RSS, "\n")
```

```
## Residual Sum of Squares (RSS): 984669.4
```

```
# 3. Calculate the Explained Sum of Squares (ESS)
# ESS is the difference between TSS and RSS
ESS <- TSS - RSS
cat("Explained Sum of Squares (ESS):", ESS, "\n")
```

```
## Explained Sum of Squares (ESS): 433097
```

```
# 4. Verify that TSS = ESS + RSS
cat("TSS == ESS + RSS:", TSS == (ESS + RSS), "\n")
```

```
## TSS == ESS + RSS: TRUE
```

From this we can also compute the R^2 measure by hand.

```
# 1. Calculate Total Sum of Squares (TSS)
mean_price <- mean(listings_clean_filtered$price)
TSS <- sum((listings_clean_filtered$price - mean_price)^2)

# 2. Calculate Residual Sum of Squares (RSS)
RSS <- sum(residuals_accommodates^2)

# 3. Calculate Explained Sum of Squares (ESS)
ESS <- TSS - RSS

# 4. Calculate R^2 using the formula R^2 = ESS / TSS or 1 - RSS / TSS
R_squared <- 1 - (RSS / TSS)

# Display the result
cat("R^2 (by hand):", R_squared, "\n")
```

```
## R^2 (by hand): 0.3054784
```

It is the same as above in the regressian output (which of course has to be the case)!

Next we use the Stargazer package that we loaded in the beginning to produce a nice table with regression results for three different specifications. See slide 24 for details.

```r
# 1. Run a level-level regression of price on accommodates
level_level <- lm(price ~ accommodates, data = listings_clean_filtered)

# 2. Run a log-level regression of log(price) on accommodates
log_level <- lm(log(price) ~ accommodates, data = listings_clean_filtered)

# 3. Run a log-log regression of log(price) on log(accommodates)
log_log <- lm(log(price) ~ log(accommodates), data = listings_clean_filtered)

# 4. Create a table with stargazer displaying all three results
stargazer(level_level, log_level, log_log,
          type = "text",
          title = "Regression Results",
          column.labels = c("level-level", "log-level", "log-log"),
          dep.var.labels = "price",
          covariate.labels = c("accommodates", "log(accommodates)", "constant"),
          omit.stat = c("f", "ser"),
          digits = 3)
```

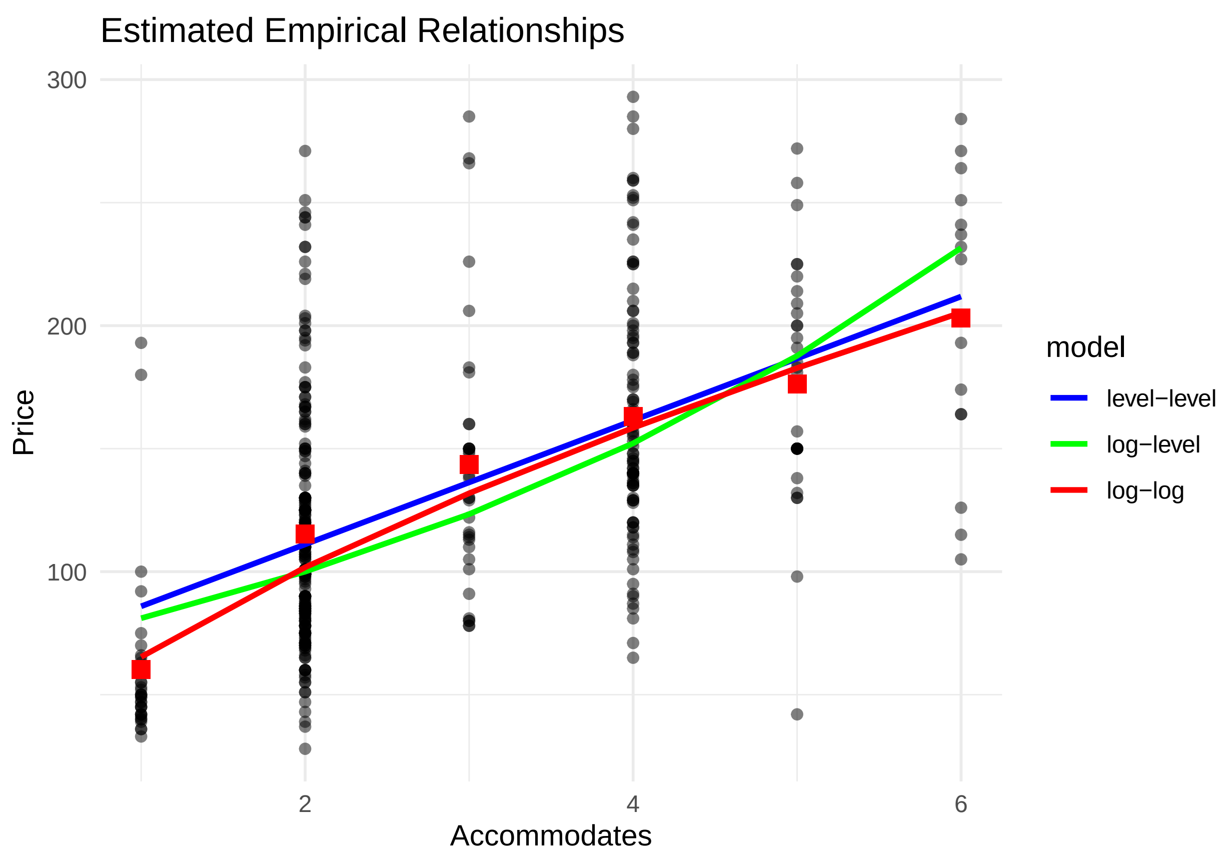
```
##
## Regression Results
## ================================================
##                       Dependent variable:
##                     ------------------------------
##                        price       log(price)
##                     level-level log-level log-log
##                         (1)        (2)      (3)
## ------------------------------------------------
## accommodates         25.183***  0.210***
##                       (1.855)    (0.015)
##
## log(accommodates)                          0.639***
##                                            (0.040)
##
## constant             60.731***  4.186***  4.180***
##                       (5.735)    (0.046)  (0.041)
##
## ------------------------------------------------
## Observations            421        421      421
## R2                     0.305      0.322    0.381
## Adjusted R2            0.304      0.320    0.379
## ================================================
## Note:                 *p<0.1; **p<0.05; ***p<0.01
```

Finally, we produce a figure with the data points and the three fitted regression lines.[1]

```r
# 1. Create a new data frame with predictions
listings_clean_filtered$pred_level_level <- predict(level_level, newdata = listings_clean_filtered)
listings_clean_filtered$pred_log_level <- exp(predict(log_level, newdata = listings_clean_filtered))
listings_clean_filtered$pred_log_log <- exp(predict(log_log, newdata = listings_clean_filtered))

# 2. Calculate the average price for each value of accommodates
```

[1]Here, we ignore an issue that the 7th edition of Wooldrige's book discusses on p. 206ff. In brief, the issue is that the predicted value of the dependent variable is not the expenential function evaluated at the fitted value when the dependent variable in the regression was in log form. We ignore this issue here for simplicity.

```r
avg_prices <- listings_clean_filtered %>%
  group_by(accommodates) %>%
  summarise(avg_price = mean(price))

# 3. Scatter plot of the actual data (price vs accommodates)
p <- ggplot(listings_clean_filtered, aes(x = accommodates, y = price)) +
  geom_point(alpha = 0.5) +
  labs(title = "Estimated Empirical Relationships",
       x = "Accommodates",
       y = "Price") +
  theme_minimal()

# 4. Add the fitted values from the level-level model
p <- p + geom_line(aes(y = pred_level_level, color = "level-level"), linewidth = 1)

# 5. Add the fitted values from the log-level model
p <- p + geom_line(aes(y = pred_log_level, color = "log-level"), linewidth = 1)

# 6. Add the fitted values from the log-log model
p <- p + geom_line(aes(y = pred_log_log, color = "log-log"), linewidth = 1)

# 7. Add the average price for each accommodates value as red squares
p <- p + geom_point(data = avg_prices, aes(x = accommodates, y = avg_price),
                    color = "red", shape = 15, size = 3)

# 8. Add legend and show the plot
p <- p + scale_color_manual(name = "model",
                            values = c("level-level" = "blue", "log-level" = "green", "log-log" = "red"

# Display the plot
print(p)
```


Estimated Empirical Relationships
300
200
100
Price
2
4
6
Accommodates
model
level−level
log−level
log−log