Identifying Drivers of Outcomes: Linear Models

Luis Francisco Gomez Lopez

FAEDIS

2023-10-31

Contents

Please Read Me

- This presentation is based on (Chapman and Feit 2019, chap. 7)

Purpose

-

Amusement park survey

- **weekend**: whether the visit was on a weekend
- **num.child**: number of children in the visit
- **distance**: how far the customer traveled to the park in miles
- **rides**: satisfaction with rides using a scale [0, 100]
- **games**: satisfaction with games using a scale [0, 100]
- **wait**: satisfaction with waiting times using a scale [0, 100]
- **clean**: satisfaction with cleanliness using a scale [0, 100]
- **overall**: overall satisfaction rating using a scale [0, 100]

Amusement park survey

- Import data

```
amusement_park <- read_csv("http://goo.gl/HKnl74")
amusement_park |> head(n = 5)
```

```
# A tibble: 5 x 8
  weekend num.child distance rides games  wait clean overall
  <chr>       <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 yes             0    115.     87    73    60    89      47
2 yes             2     27.0    87    78    76    87      65
3 no              1     63.3    85    80    70    88      61
4 yes             0     25.9    88    72    66    89      37
5 no              4     54.7    84    87    74    87      68
```

Amusement park survey

- Transform data

```
amusement_park <- amusement_park |>
  mutate(weekend = factor(x = weekend,
                          labels = c('no', 'yes'),
                          ordered = FALSE),
         num.child = as.integer(num.child),
         # logarithmic transform
         logdist = log(distance, base = exp(x = 1)))
amusement_park |> head(n = 5)
```

```
# A tibble: 5 x 9
  weekend num.child distance rides games  wait clean overall logdist
  <fct>       <int>    <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
1 yes             0    115.     87    73    60    89      47    4.74
2 yes             2     27.0    87    78    76    87      65    3.30
3 no              1     63.3    85    80    70    88      61    4.15
4 yes             0     25.9    88    72    66    89      37    3.25
5 no              4     54.7    84    87    74    87      68    4.00
```

Amusement park survey

- Summarize data
  - Ups the table is really big!!! Try it in your console to see the complete table

```
amusement_park |> skim()
```

Table 1: Data summary

Name	amusement_park
Number of rows	500
Number of columns	9
_______________________	
Column type frequency:	
factor	1
numeric	8
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
weekend	0	1	FALSE	2	no: 259, yes: 241

Amusement park survey

- Correlation matrices
  - Pearson correlation coefficients for samples in a tibble

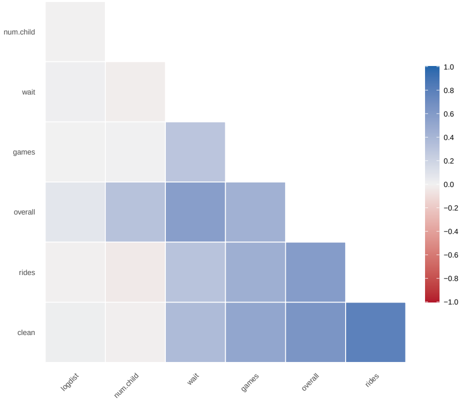
```
correlation_matrix <- amusement_park |>
  select(num.child, rides:logdist) |>
  corrr::correlate()
correlation_matrix
```

```
# A tibble: 7 x 8
  term      num.child    rides    games    wait    clean overall  logdist
  <chr>         <dbl>    <dbl>    <dbl>   <dbl>    <dbl>   <dbl>    <dbl>
1 num.child  NA       -0.0403  0.00466 -0.0210 -0.0135   0.319 -0.00459
2 rides     -0.0403   NA       0.455    0.314   0.790    0.586  -0.0110
3 games      0.00466   0.455  NA        0.299   0.517    0.437   0.00187
4 wait      -0.0210    0.314   0.299   NA       0.368    0.573   0.0175
5 clean     -0.0135    0.790   0.517    0.368  NA        0.639   0.0221
6 overall    0.319     0.586   0.437    0.573   0.639   NA       0.0763
7 logdist   -0.00459 -0.0110   0.00187  0.0175  0.0221   0.0763 NA
```

Amusement park survey

- Correlation matrices
  - Pearson correlation coefficients for samples in a tibble

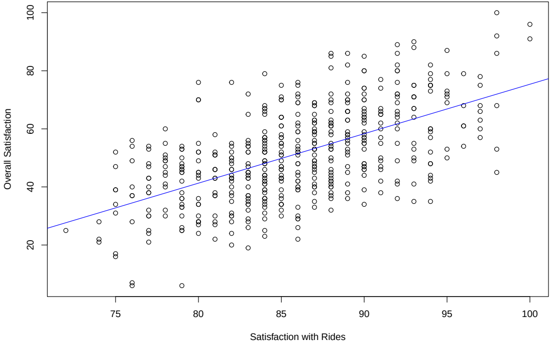
```
correlation_matrix |> autoplot(triangular = "lower")
```



Amusement park survey

- **Bivariate Association: the base R way**

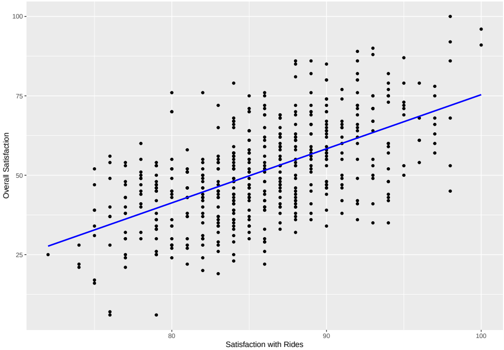
```
plot(overall~rides, data=amusement_park,
    xlab="Satisfaction with Rides", ylab="Overall Satisfaction")
abline(reg = lm(formula = overall~rides, data = amusement_park),
       col = 'blue')
```



Amusement park survey

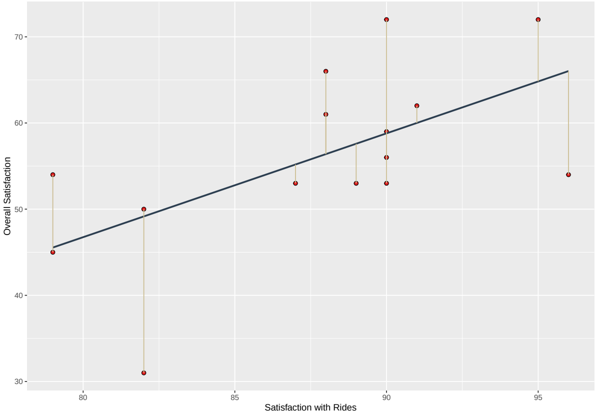
- **Bivariate Association: the tidyverse way**

```
amusement_park |> ggplot(aes(x = rides, y = overall)) +
  geom_point() +
  geom_smooth(method = 'lm',
              color = 'blue',
              se = FALSE) +
  labs(x = "Satisfaction with Rides",
       y = "Overall Satisfaction")
```



Amusement park survey

- Linear Model with a Single Predictor



Amusement park survey

- Linear Model with a Single Predictor

$$overall_i = \beta_0 + \beta_1 rides_i + \epsilon_i \text{ where } \epsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ and } i = 1, \dots, 500$$

$$\widehat{overall}_i = \hat{\beta}_0 + \hat{\beta}_1 rides_i \text{ and } \hat{\sigma}^2 \text{ where } i = 1, \dots, 500$$

$$overall_i - \widehat{overall}_i = \hat{\epsilon}_i \text{ where } i = 1, \dots, 500$$

```r
model1 <- lm(formula = overall ~ rides, data = amusement_park)
model1
```

```

Call:
lm(formula = overall ~ rides, data = amusement_park)

Coefficients:
(Intercept)        rides
    -94.962        1.703
```

Amusement park survey

- Linear Model with a Single Predictor

```r
ls.str(model1)
```

```
assign :  int [1:2] 0 1
call :  language lm(formula = overall ~ rides, data = amusement_park)
coefficients :  Named num [1:2] -95 1.7
df.residual :  int 498
effects :  Named num [1:500] -1146.2 -207.9 11.5 -17.9 20.3 ...
fitted.values :  Named num [1:500] 53.2 53.2 49.8 54.9 48.1 ...
model : 'data.frame':   500 obs. of  2 variables:
 $ overall: num  47 65 61 37 68 27 40 30 58 36 ...
 $ rides  : num  87 87 85 88 84 81 77 82 90 88 ...
qr : List of 5
 $ qr   : num [1:500, 1:2] -22.3607 0.0447 0.0447 0.0447 0.0447 ...
 $ qraux: num [1:2] 1.04 1.01
 $ pivot: int [1:2] 1 2
 $ tol  : num 1e-07
 $ rank : int 2
rank :  int 2
residuals :  Named num [1:500] -6.22 11.78 11.18 -17.93 19.89 ...
terms : Classes 'terms', 'formula'  language overall ~ rides
xlevels :  Named list()
```

Amusement park survey

- Linear Model with a Single Predictor

```r
summary(model1)
```

```
Call:
lm(formula = overall ~ rides, data = amusement_park)

Residuals:
    Min      1Q  Median      3Q     Max 
-33.597 -10.048   0.425   8.694  34.699 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -94.9622     9.0790  -10.46   <2e-16 ***
rides         1.7033     0.1055   16.14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.88 on 498 degrees of freedom
Multiple R-squared:  0.3434,	Adjusted R-squared:  0.3421 
F-statistic: 260.4 on 1 and 498 DF,  p-value: < 2.2e-16
```

Amusement park survey

- Linear Model with a Single Predictor

```
model1$coefficients
```

```
(Intercept)       rides
 -94.962246    1.703285
```

```
# Make some predictions
# We want to forecast the overall satisfaction rating
# if the satisfaction with rides is 95
-94.962246 + 1.703285*95
```

```
[1] 66.84983
```

Amusement park survey

- Linear Model with a Single Predictor
  - Std. Error column
    - Indicates uncertainty in the coefficient estimate
    - We can build a confidence interval

```r
summary(model1)$coefficients[, 2]
```

```
(Intercept)       rides 
  9.0790049   0.1055462 
```

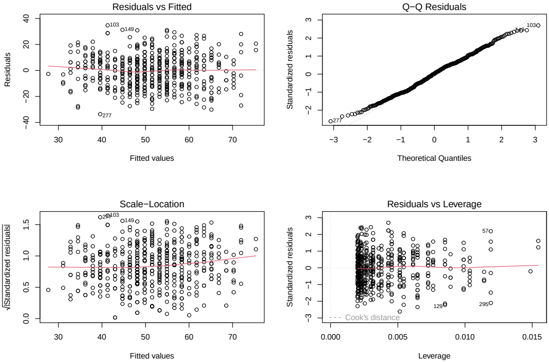
```r
confint(model1, level = 0.95)
```

```
                  2.5 %     97.5 %
(Intercept) -112.800120 -77.124371
rides          1.495915   1.910656
```

Amusement park survey

- Linear Model with a Single Predictor

```r
par(mfrow=c(2,2))
plot(model1)
```




```r
par(mfrow=c(1,1))
```

Amusement park survey

- Linear Model with a Single Predictor
  - **Linearity**: plot $(1, 1)$
    - Reference line should be flat and horizontal
  - **Normality of residuals**: plot $(1, 2)$
    - Dots should fall along the line
  - **Homogeneity of variance**: plot $(2, 1)$
    - Reference line should be flat and horizontal
  - **Influential observations**: plot $(2, 2)$
    - Points should be inside the contour lines

Amusement park survey

- Linear Model with Multiple Predictors

$$
\begin{aligned}
overall_i = \beta_0 + \beta_1 rides_i + \beta_2 games_i \\
+ \beta_3 wait_i + \beta_3 clean_i + \epsilon_i \\
\text{where } \epsilon_i \sim \mathcal{N}(0, \sigma^2) \text{ and } i = 1, \ldots, 500
\end{aligned}
$$

```r
model2 <- lm(formula = overall ~ rides + games + wait + clean,
             data = amusement_park)
model2
```

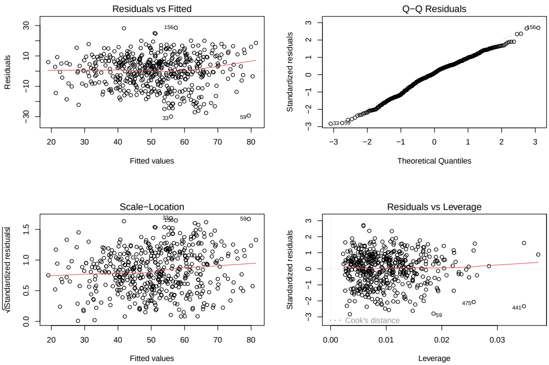
```
Call:
lm(formula = overall ~ rides + games + wait + clean, data = amusement_park)

Coefficients:
(Intercept)        rides        games         wait        clean
  -131.4092       0.5291       0.1533       0.5533       0.9842
```

Amusement park survey

- Linear Model with Multiple Predictors

```r
par(mfrow=c(2,2))
plot(model2)
```




```r
par(mfrow=c(1,1))
```

Amusement park survey

- Linear Model with Multiple Predictors

```
summary(model2)
```

```
Call:
lm(formula = overall ~ rides + games + wait + clean, data = amusement_park)

Residuals:
    Min      1Q  Median      3Q     Max 
-29.944  -6.841   1.072   7.167  28.618 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -131.40919    8.33377 -15.768  < 2e-16 ***
rides          0.52908    0.14207   3.724 0.000219 ***
games          0.15334    0.06908   2.220 0.026903 *  
wait           0.55333    0.04781  11.573  < 2e-16 ***
clean          0.98421    0.15987   6.156 1.54e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.59 on 495 degrees of freedom
Multiple R-squared:  0.5586,	Adjusted R-squared:  0.5551 
F-statistic: 156.6 on 4 and 495 DF,  p-value: < 2.2e-16
```

Amusement park survey

- Linear Model with Multiple Predictors

$$H_0 : \beta_{rides} = 0$$

$$H_1 : \beta_{rides} \neq 0$$

$$t_{rides} = \frac{\hat{\beta}_{rides} - \beta_{rides}}{\widehat{Var(\hat{\beta}_{rides})}} = \frac{0.529078 - 0}{0.14207176} = 3.724019$$

```
model2$coefficients
```

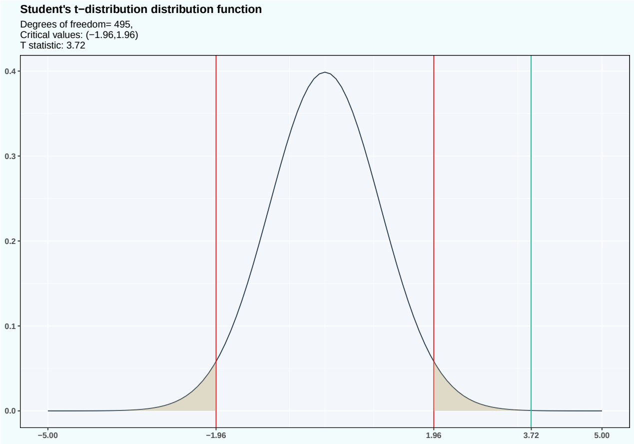
```
 (Intercept)        rides        games         wait        clean
-131.4091939    0.5290780    0.1533361    0.5533264    0.9842126
```

```
# Calculate the variance-covariance matrix, extract
# the diagonal and calculate the standard deviaton of
# the parameters
model2 |> vcov() |> diag() |> sqrt()
```

```
(Intercept)       rides       games        wait       clean
 8.33376643  0.14207176  0.06908486  0.04781282  0.15986712
```

Amusement park survey

- Linear Model with Multiple Predictors



Amusement park survey

- Linear Model with Multiple Predictors

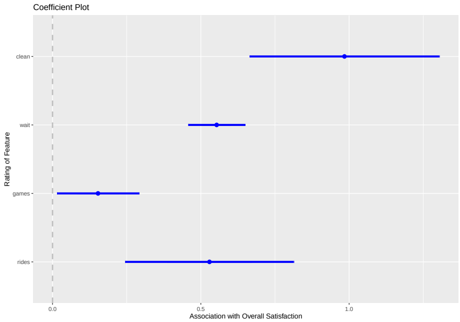
```r
confint(model2, level = 0.95)
```

```
                   2.5 %       97.5 %
(Intercept) -147.78311147 -115.0352764
rides          0.24993998    0.8082161
games          0.01760038    0.2890718
wait           0.45938535    0.6472675
clean          0.67011082    1.2983144
```

Amusement park survey

- Linear Model with Multiple Predictors

```
library(coefplot) # Remember to install the package if it is not installed
coefplot(model = model2,
        # The intercept is relatively large: -131.4092
        intercept = FALSE,
        ylab="Rating of Feature",
        xlab="Association with Overall Satisfaction",
        lwdOuter = 1.5)
```



Amusement park survey

- Comparing models

```
summary(model1)$r.squared
```

```
[1] 0.3433799
```

```
summary(model2)$r.squared
```

```
[1] 0.558621
```

```
summary(model1)$adj.r.squared
```

```
[1] 0.3420614
```

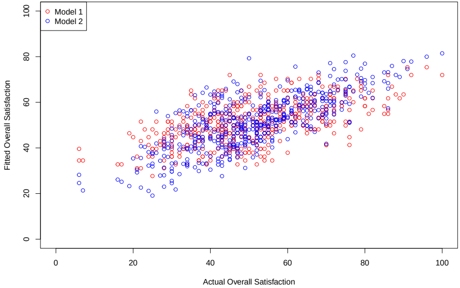
```
summary(model2)$adj.r.squared
```

```
[1] 0.5550543
```

Amusement park survey

- Comparing models
  - **Base R way**

```
plot(x = amusement_park$overall, y = fitted(model1),
     col = "red", xlim = c(0,100), ylim = c(0,100),
     xlab = "Actual Overall Satisfaction",
     ylab = "Fitted Overall Satisfaction")
points(x = amusement_park$overall, y = fitted(model2),
       col = "blue")
legend(x = "topleft", legend = c("Model 1", "Model 2"), col = c("red", "blue"), pch = 1)
```



Amusement park survey

- Comparing models
  - **Tidymodels and tidyverse way**: Prepare data

```
model1_augment <- augment(x = model1) |> mutate(model = "Model 1")
model2_augment <- augment(x = model2) |> mutate(model = "Model 2")
models_performance <- model1_augment |> bind_rows(model2_augment)

models_performance |> glimpse()
```

```
Rows: 1,000
Columns: 12
$ overall    <dbl> 47, 65, 61, 37, 68, 27, 40, 30, 58, 36, 71, 48, 75, 46, 59,~
$ rides      <dbl> 87, 87, 85, 88, 84, 81, 77, 82, 90, 88, 93, 79, 94, 81, 86,~
$ .fitted    <dbl> 53.22359, 53.22359, 49.81702, 54.92688, 48.11373, 43.00388,~
$ .resid     <dbl> -6.2235914, 11.7764086, 11.1829795, -17.9268769, 19.8862650~
$ .hat       <dbl> 0.002089430, 0.002089430, 0.002048063, 0.002311576, 0.00222~
$ .sigma     <dbl> 12.88964, 12.88182, 12.88289, 12.86751, 12.86171, 12.87260,~
$ .cooksd    <dbl> 2.449537e-04, 8.770564e-04, 7.751689e-04, 2.249493e-03, 2.6~
$ .std.resid <dbl> -0.48371422, 0.91529407, 0.86915315, -1.39348008, 1.5457218~
$ model      <chr> "Model 1", "Model 1", "Model 1", "Model 1", "Model 1", "Mod~
$ games      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ wait       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
$ clean      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

Amusement park survey

- Comparing models
  - **Tidymodels and tidyverse way**: Visualize

```
models_performance |>
  ggplot() +
  geom_point(aes(x = overall, y = .fitted,
                 color = model)) +
  labs(x = "Actual Overall Satisfaction",
       y = "Fitted Overall Satisfaction")
```

Amusement park survey

- Predictions

$$overall_j = \hat{\beta}_0 + \hat{\beta}_1 rides_j + \hat{\beta}_2 games_j$$
$$+ \hat{\beta}_3 wait_j + \hat{\beta}_3 clean_j$$

```
coef(model2) |> enframe(name = "coef")
```

```
# A tibble: 5 x 2
  coef          value
  <chr>         <dbl>
1 (Intercept) -131.
2 rides         0.529
3 games         0.153
4 wait          0.553
5 clean         0.984
```

Amusement park survey

- Predictions
    - Manual

```
(coef(model2)["(Intercept)"] +
    coef(model2)["rides"]*30 +
    coef(model2)["games"]*10 +
    coef(model2)["wait"]*57 +
    coef(model2)["clean"])*90 |>
  unname()
```

```
(Intercept)
  -7333.171
```

References

Chapman, Chris, and Elea McDonnell Feit. 2019. *R For Marketing Research and Analytics*. 2nd ed. 2019. Use R! Cham: Springer International Publishing : Imprint: Springer. https://doi.org/10.1007/978-3-030-14316-9.