STAT 100: Week 10

Ricky’s Section

Introductions and Attendance

Introduction: Name

Question of the Week: If money/time/prestige didn't matter, what would be your dream job and why?

Important Reminders

Anonymous Feedback

https://docs.google.com/forms/d/e/1FAIpQLSfKv_FGvsooqm-IvtxKx3Vf6bBzSJE2jamK1gklAzL6NkXE8w/viewform

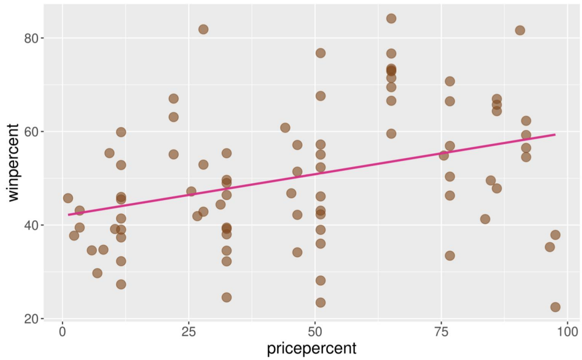
Content Review: Week 10

Linear Regression (In a Nutshell)

- **Linear regression**: Models the **linear** relationship between **numerical response variable (y)** and **explanatory variables (x)**, which can be either **numerical** or **categorical**
  - For now, we'll focus on **simple linear regression**, which only has one **explanatory variable**
- The form of this model is $\hat{\mathbf{y}} = \hat{\mathbf{B}}_0 + \hat{\mathbf{B}}_1\mathbf{x}$
  - Note: $\hat{B}$ is supposed to represent beta hat (β + ^)
- The **coefficients** ($\hat{\mathbf{B}}_0$ **and** $\hat{\mathbf{B}}_1$) have different interpretations depending on whether x is **numerical** or **categorical**

Explanatory Variable: Numerical

- When x is **numerical...**
  - The model represents a "line of best fit"
  - $\hat{B}_0$ is the **y-intercept**
    - *When price percentage equals 0%, the average win percentage is 42%*
  - $\hat{B}_1$ is the **slope**
    - *As price percentage increases by 1%, the win percentage increases by 0.178%, on average*
  - **Least-squares regression** finds the optimal values of $\hat{B}_0$ **and** $\hat{B}_1$ by minimizing **residuals** (errors)




```
# A tibble: 2 × 7
  term         estimate std_error statistic p_value lower_ci upper_ci
  <chr>           <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept      42.0       2.91      14.4    0        36.2     47.8
2 pricepercent    0.178     0.053      3.35   0.001     0.072    0.283
```

- Interpretation of the coefficients?

$\hat{\beta}_0 = 42$

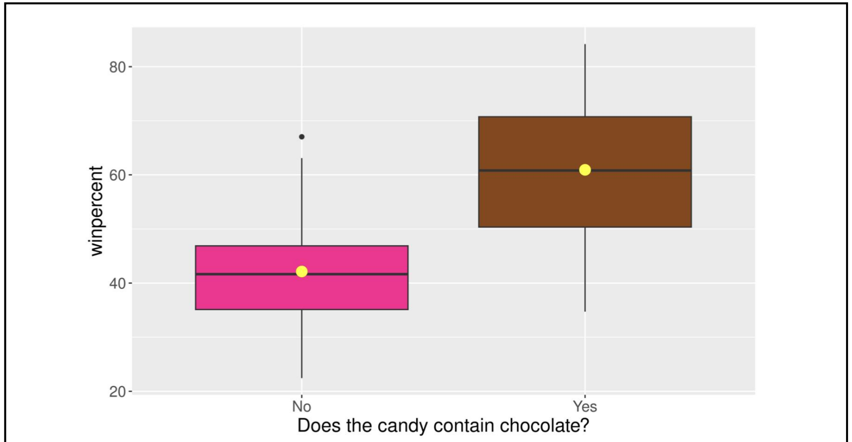
$\hat{\beta}_1 = 0.178$

intercept: when candy has $% of 0, we expect its win % to be 42

slope: as we increase $ by 1%, we expect win % to increase by 0.178% points.

Explanatory Variable: Binary Categorical

- When x is **binary categorical**…
  - The model represents means (one for each of the two group)
  - $\hat{B}_0$ is the mean of y in the **baseline group** (when x = 0)
    - *For candy without chocolate, the average win percentage is 42.1%*
  - $\hat{B}_1$ is the **difference in means** of **other group** from **baseline group** ($\bar{y}_{other}$ - $\bar{y}_{baseline}$)
    - *Candy with chocolate has a higher average win percentage than candy without chocolate by 18.8%*




```
# A tibble: 2 × 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept     42.1      1.65     25.6        0     38.9     45.4
2 chocolate     18.8      2.50      7.52       0     13.8     23.7
```

$\hat{B}_0$ → row 1; $\hat{B}_1$ → row 2

win % of non choc

pos. slope bc. pos. relationship between choco. and win %

Linear Regression: Code

- **Fitting the model**: Use this to build your model
  - MODEL <- lm(Y-VAR ~ X-VAR, data = DATASET)
  - *model <- lm(winpercent ~ pricepercent, data = candy)*
- **Getting the numbers**: Use this to summarize your model
  - get_regression_table(MODEL)
  - *get_regression_table(model)*
- **Predicting**: Use this for your model to predict y-value of new instances
  - predict(MODEL, newdata = data.frame(Y-VAR = VALUE))
  - *predict(model, newdata = data.frame(pricepercent = 85))*

More on Linear Regression

- **Interpolation**: Predicting values that fall **within** a dataset (generally good)
- **Extrapolation**: Predicting values that fall **outside** an observed range (generally not good)
- **Residual**: Error in **observed y** versus **predicted y** (**positive residual** means model **underestimated**; **negative residual** means model **overestimated**)
  - $e_i = y_i - \hat{y}_i$ (observed - predicted)
- **Sample correlation coefficient (r)**: Measures **strength** of **linear relationship** between 2 **numeric variables** in a **sample**, ranging from -1 to 1
  - -1 is perfectly negative relationship
  - 1 is perfectly positive relationship

If r ranges from –1 to 1, what are the possible values for r^2?

Question:

If r ranges from -1 to 1, what are the possible values for r^2?

0-1!

As a result of squaring the numbers, r^2 can only take on non-negative values.

r^2: Coefficient of Determination

- **r^2**: Percent of **total variation** in y (**response variable**) explained by the **model**
    - **$r^2 = (r)^2 = Var(\hat{y}_i)/Var(y_i)$**
    - If the **linear model** perfectly captured the **variability** in the observed data, then $Var(\hat{y}_i) = Var(y_i)$; thus, **$r^2$** would be 1
    - If **r^2** is too low, try different model; however, **r^2** only increases as new **predictors** are added to a model
- **adj(r^2)**: Value of **r^2** adjusted for size of model (penalizes too-large models)
    - **$adj(r^2) = r^2 \times ((n - 1)/(n - p - 1))$**
    - n is sample size, p is number of predictors in model
- Basically, graph your data and pick the model with **highest adj(r^2)**
    - glance(MODEL)
    - *Ex: glance(model)*

The model predicts a y-value of 26 while the (actual) observed y-value is 30. What is the residual, and what does it mean?

Question:

The model predicts a y-value of 26 while the (actual) observed y-value is 30. What is the residual, and what does it mean?

$\mathbf{e_i} = \mathbf{y_i} - \mathbf{\hat{y}_i}$ (observed - predicted)

The **residual** is 4 (30 - 26). Thus, the model **underestimated** by 4.

Visually, the "line of best fit" is below the actual data point.

Population Model vs. Estimated Model

- **Population model**: $y = B_0 + B_1x + \varepsilon$
  - ε is **error/"random noise"** around the line (**population parameter** for the **residuals**)
  - $\varepsilon \sim N(0, \sigma)$
  - B_0 **and** B_1 are **population parameters**
- **Estimated model**: $\hat{y} = \hat{B}_0 + \hat{B}_1x$
  - This is what our "line of best fit" is
  - $\hat{B}_0$ **and** $\hat{B}_1$ are estimates of the **population parameters**
  - ε "disappears" because the **estimated model** is a straight line

Where else have we seen “hats” (^) used to indicate estimates?

Question:

Where else have we seen “hats” (^) used to indicate estimates?

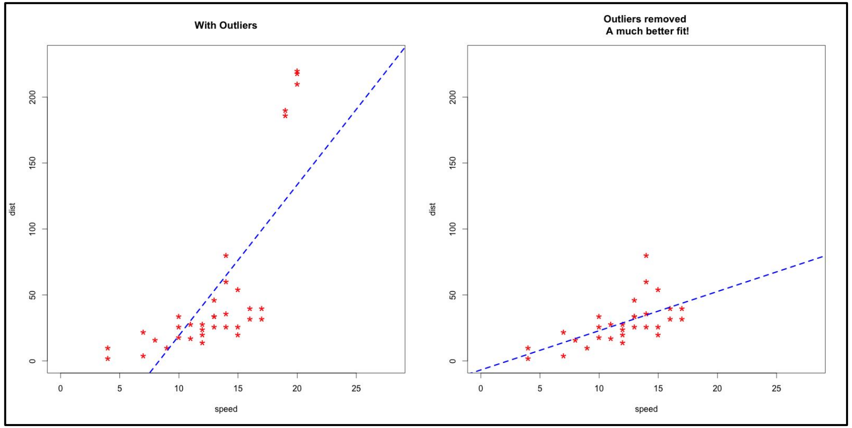
Inference!

Recall $\hat{\mathbf{p}}$ (sample proportion) is used to estimate **p** (population proportion).

This is a common theme in statistics.

Influential Points

- **High leverage**: Points with unusual **x-values** relative to rest of data points
  - These points have a large effect on $\hat{B}_0$ **and** $\hat{B}_1$
- **Outliers**: Points with unusual **y-values** relative to their **x-values**
  - These points do not follow the general linear trend in the data
- **Influential points**: Points with a strong effect on $\hat{B}_0$ **and** $\hat{B}_1$ (when removed, these coefficients substantially change)
  - **Outliers** with **high leverage** are potentially **influential**

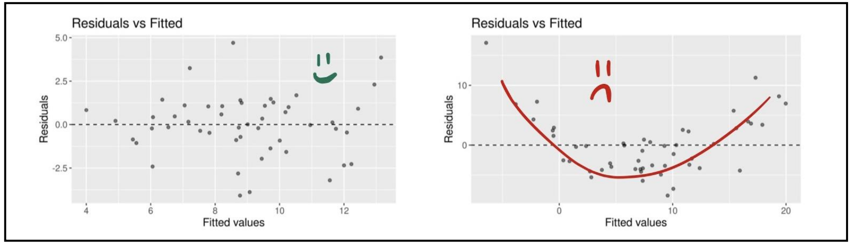


Assumptions for Linear Regression

- **Linearity**: The data shows a **linear** trend (thus, a linear model is appropriate)
- **Constant Variability**: The **variability** of the **response variable** about the line remains roughly constant as the **explanatory variable changes**
- **Independence**: Each observation is **independent** (i.e., value of one observation provide no information about value of others)
- **Normality**: The **residuals** (errors) are approximately **normally distributed**

Assumption #1: Linearity

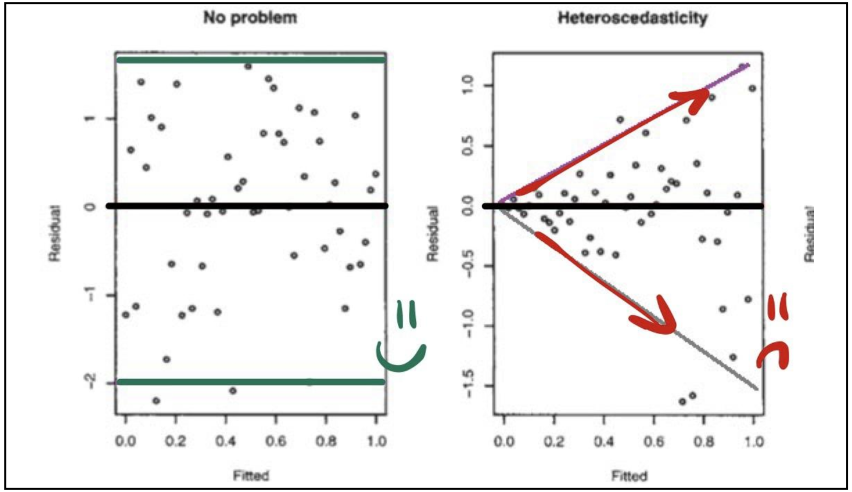
- Check via **residual plot**, which plots residuals of model across domain
- If data is linear, points should scatter from y = 0 randomly, with no pattern



- ggplot(MODEL) + stat_fitted_resid()
- *ggplot(model) + stat_fitted_resid(alpha = 0.25)*

Assumption #2: Constant Variance

- Check via **residual plot**, which plots residuals of model across domain
- Vertical spread of points should be roughly constant across domain, with no "fanning"
    - This interpretation is different from linearity; here, cite the upper and lower bounds (in green) to show there is no "fanning"



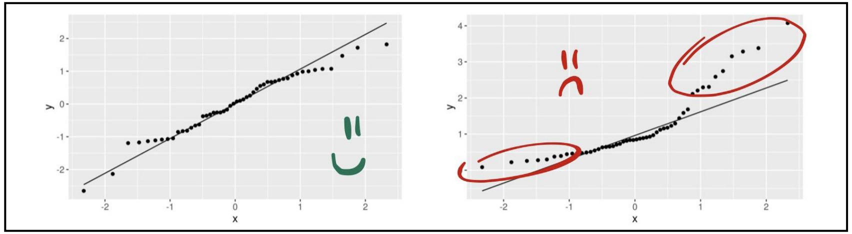
- ggplot(MODEL) + stat_fitted_resid()
- *ggplot(model) + stat_fitted_resid(alpha = 0.25)*

Assumption #3: Independence

- Check by considering **how data was collected**
- If there's **independence**, knowing observation #1 gives no information about observation #2
  - *Ex: If data was randomly sampled, then independence can be reasonably assumed*
  - *Ex: If data was collected within a family (and we're measuring blood sugar, e.g.), then independence might not apply. Why?*

Assumption #4: Normality

- Check via **Q-Q plot**, which plots residuals against theoretical quantiles of **normal distribution**
  - If residuals were perfectly **normally distributed**, they'd exactly follow the diagonal
  - We're not looking for perfect—just make sure it's reasonable
- Points should have a linear relationship, with no breaks at tails



- ggplot(MODEL) + stat_normal_qq()
- *ggplot(model) + stat_normal_qq(alpha = 0.25)*

Inference in Regression: Hypothesis Tests

- The **observed data** (x_i, y_i) is assumed to have been **randomly sampled** from a population where the **explanatory variable** (X) and the **response variable** (Y) follow a **population model**
  - **Population model**: $Y = B_0 + B_1X + \varepsilon$
    - Like before, but we're now using capital letters to indicate **random variables**
  - **Estimated model**: $\hat{y} = \hat{B}_0 + \hat{B}_1x$
- Usually, we're concerned with **slope parameter** (B_1)
  - $H_0: B_1 = 0$ (i.e., the slope is zero, so there is no association between X and Y)
  - $H_A: B_1 \neq 0$ (i.e., the slope is non-zero, so there is some association between X and Y)

Inference in Regression: Hypothesis Tests

- When **assumptions** are met (including 4 assumptions for linear regression), then the ***t*-statistic** follows a ***t*-distribution** with **degrees of freedom n − 2**, where n is the number of ordered pairs in the dataset
  - **$t = (\hat{B}_1 - B_1^0)/SE(\hat{B}_1)$**
    - Recall our null hypothesis is (often) $B_1 = 0$, so the B_1^0 term can go away
  - **$t = (\hat{B}_1)/SE(\hat{B}_1)$**
- Our computers can calculate this for us!
  - get_regression_table(MODEL)
  - *get_regression_table(model)*

Inference in Regression: Confidence Intervals

- **Confidence interval**: Recall the form of a confidence interval is **CI = sample statistic ± ME**
- **CI = $\hat{B}_1 \pm (t^* \times SE(\hat{B}_1))$**
    - **t*** is the point on a ***t*-distribution** with **n−2 degrees of freedom** and **α/2 area to the right**
    - **"We are {α}% confident B_1 is in the CI; that is, with {α}% confidence, an increase in {explanatory variable} by 1 unit is associated with a change in average {response variable} between {lower bound} and {upper bound} units."**
    - *Ex: With 95% confidence, an increase in age of one year is associated with a change in average RFFT score between (-1.44, -1.08) points; i.e., a decrease in average RFFT score between 1.08 to 1.44 points.*
- Again, our computers can calculate this for us (use `get_regression_table()`)!

Confidence Interval vs. Prediction Interval

- **Confidence interval for mean response**: Tries to find plausible range for **parameter**
  - Centered at $\hat{y}$, with **smaller SE**
  - *Ex: We are 95% confident that the average RFFT score for individuals who are 50 years old is between 72.27 and 76.69 points.*

- **Prediction interval for individual response**: Tries to find plausible range for a **single, new observation**
  - Centered at $\hat{y}$, with **larger SE**
  - *Ex: For a 50-year-old individual, we predict, with 95% confidence, their RFFT score is between 28.87 and 120.10 points.*

Confidence Interval vs. Prediction Interval: Code

- OBSERVATION-OF-INTEREST <- data.frame(EXPL-VAR(S) = VALUE(S))
- predict(MODEL, newdata = OBSERVATION-OF-INTEREST, interval = "confidence", level = CONF-LEVEL)
    - *house_of_interest <- data.frame(livingArea = 1500, age = 20, bathrooms = 2, centralAir = "yes")*
    - *predict(model, house_of_interest, interval = "confidence", level = 0.95)*

- OBSERVATION-OF-INTEREST <- data.frame(EXPL-VAR(S) = VALUE(S))
- predict(MODEL, newdata = OBSERVATION-OF-INTEREST, interval = "prediction", level = CONF-LEVEL)
    - *house_of_interest <- data.frame(livingArea = 1500, age = 20, bathrooms = 2, centralAir = "yes")*
    - *predict(model, house_of_interest, interval = "prediction", level = 0.95)*

Intuitively, why would there be more uncertainty (and thus a higher SE) in a prediction interval than in a confidence interval?

Question:

Intuitively, why would there be more uncertainty (and thus a higher SE) in a prediction interval than in a confidence interval?

There are many factors (other than age) that go a person's RFFT score. Thus, **prediction** is highly variable.

Conversely, a **CI** tries to find a plausible range for a **parameter** (specifically, population mean). We're now thinking about a **population** rather than a **single observation**, and means "average out" with large numbers.

Questions?

Oral Exam Practice

Person A (Grade Q1 and Q3, Answer Q2 and Q4)

https://docs.google.com/document/d/1jw1NSm-EGHBConSE2OLU8WY5CfpnC5Q1TI_JgEHGDNg/edit?usp=sharing

Person B (Grade Q2 and Q4, Answer Q1 and Q3)

https://docs.google.com/document/d/1smvpX3niZ5QWhvKGahQ-es8mqJAeXlAmpVNX5ZZlg9M/edit?usp=sharing

Solutions

https://docs.google.com/document/d/1EBAkWs375OYEPssi8EPg8YbCbDoy9H6VFs7Y43cYJ4U/edit?usp=sharing

P-Set 7

Have a great rest of your week!