Section 001 CDS 101 *Introduction to Computational and Data Sciences*

Class 27: Modeling III

May 1, 2018

General

Annoucements

- Homework 5 posted and due on Friday, May 4th by 11:59pm
    - First 3 questions required, last 3 questions can be completed (in order) for extra credit
- Final Portfolio, due Friday, May 11th by 11:59pm
- Office hours available by appointment during the week of May 7th – May 11th for questions related to the final portfolio
- The Final Interviews will take place here (1004 Exploratory Hall) during the time scheduled for the Final Exam: Tuesday, May 15th, 1:30pm – 4:15pm
    - Scheduled time slots for each student to be posted on Slack

Case study: Mario Kart eBay prices dataset

Machine Learning and prediction

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions
- The model is "trained" on a dataset and "learns" how to reproduce the general structure and features in that dataset

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions
- The model is "trained" on a dataset and "learns" how to reproduce the general structure and features in that dataset
- In the best case scenario, you get a model with strong predictive powers that can take a series of inputs and generate a highly accurate output

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions
- The model is "trained" on a dataset and "learns" how to reproduce the general structure and features in that dataset
- In the best case scenario, you get a model with strong predictive powers that can take a series of inputs and generate a highly accurate output
- Generally only interested in accuracy, not understanding, making **prediction** distinct from **inference**

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions
- The model is "trained" on a dataset and "learns" how to reproduce the general structure and features in that dataset
- In the best case scenario, you get a model with strong predictive powers that can take a series of inputs and generate a highly accurate output
- Generally only interested in accuracy, not understanding, making **prediction** distinct from **inference**
- This accuracy comes at a price, as the most accurate prediction models are frequently the most complicated

Machine Learning and prediction

- Machine Learning models are built with the purpose of making predictions
- The model is "trained" on a dataset and "learns" how to reproduce the general structure and features in that dataset
- In the best case scenario, you get a model with strong predictive powers that can take a series of inputs and generate a highly accurate output
- Generally only interested in accuracy, not understanding, making **prediction** distinct from **inference**
- This accuracy comes at a price, as the most accurate prediction models are frequently the most complicated
- This is what people mean when they say that Machine Learning algorithms are like a "black box"

Can we predict accurately eBay prices?

- Data scraped from eBay listings for the video game *Mario Kart Wii*

Image: *Mario Kart Wii* cover art, ©Nintendo, downloaded from Wikipedia, https://en.wikipedia.org/wiki/File:Mario_Kart_Wii.png

Can we predict accurately eBay prices?

- Data scraped from eBay listings for the video game *Mario Kart Wii*
- Can we predict each game's final selling price using other information on a eBay listing page?

Image: *Mario Kart Wii* cover art, ©Nintendo, downloaded from Wikipedia, https://en.wikipedia.org/wiki/File:Mario_Kart_Wii.png

Can we predict accurately eBay prices?

- Data scraped from eBay listings for the video game *Mario Kart Wii*
- Can we predict each game's final selling price using other information on a eBay listing page?

Goal

Build a model that predicts the dataset variable `totalPr` using the other columns

Image: *Mario Kart Wii* cover art, ©Nintendo, downloaded from Wikipedia, https://en.wikipedia.org/wiki/File:Mario_Kart_Wii.png

Data exploration

What's in this dataset?

What's in this dataset?

- What are the first several entries of the *Mario Kart* dataset?

What's in this dataset?

- What are the first several entries of the *Mario Kart* dataset?

```
mariokart %>%
  glimpse()
```

What's in this dataset?

- What are the first several entries of the *Mario Kart* dataset?

```
mariokart %>%
  glimpse()
```

```
## Observations: 143
## Variables: 12
## $ ID         <dbl> 150377422259, 260483376854, 320432342985, 280405224...
## $ duration   <int> 3, 7, 3, 3, 1, 3, 1, 1, 3, 7, 1, 1, 1, 1, 7, 7, 3, ...
## $ nBids      <int> 20, 13, 16, 18, 20, 19, 13, 15, 29, 8, 15, 15, 13, ...
## $ cond       <fct> new, used, new, new, new, new, used, new, used, use...
## $ startPr    <dbl> 0.99, 0.99, 0.99, 0.99, 0.01, 0.99, 0.01, 1.00, 0.9...
## $ shipPr     <dbl> 4.00, 3.99, 3.50, 0.00, 0.00, 4.00, 0.00, 2.99, 4.0...
## $ totalPr    <dbl> 51.55, 37.04, 45.50, 44.00, 71.00, 45.00, 37.02, 53...
## $ shipSp     <fct> standard, firstClass, firstClass, standard, media, ...
## $ sellerRate <int> 1580, 365, 998, 7, 820, 270144, 7284, 4858, 27, 201...
## $ stockPhoto <fct> yes, yes, no, yes, yes, yes, yes, yes, yes, no, yes...
## $ wheels     <int> 1, 1, 1, 1, 2, 0, 0, 2, 1, 1, 2, 2, 2, 2, 1, 0, 1, ...
## $ title      <fct> ~~ Wii MARIO KART &amp; WHEEL ~ NINTENDO Wii ~ BRAN...
```

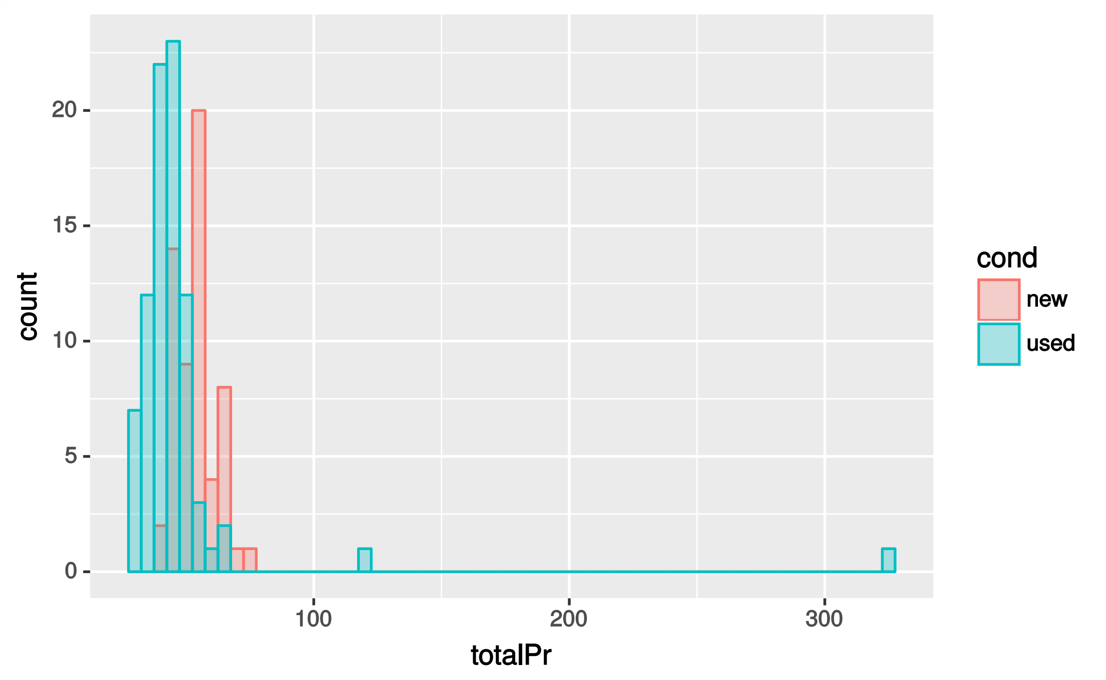
Exploring the response variable

- What is the shape and center of the response variable `totalPr`?

Exploring the response variable

- What is the shape and center of the response variable `totalPr`?

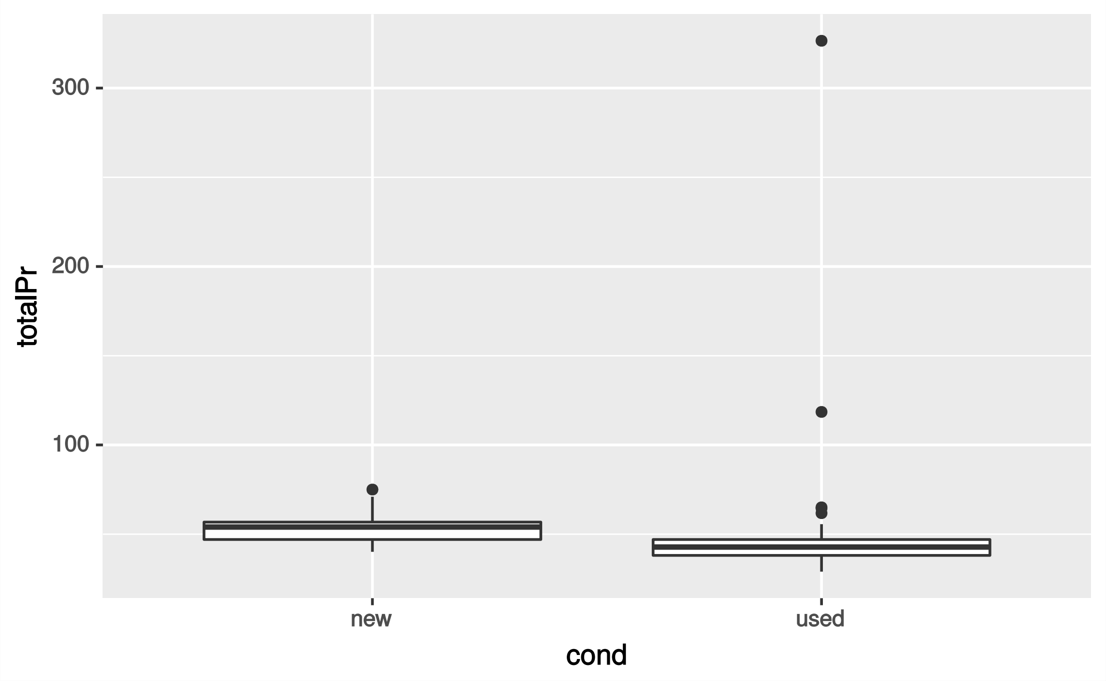
```
ggplot(mariokart) +
  geom_histogram(
    mapping = aes(x = totalPr, fill = cond, color = cond),
    position = "identity", alpha = 0.3, binwidth = 5, center = 0)
```



Exploring the response variable

- A box plot is nice to use for exploration as well

```
ggplot(mariokart) +
  geom_boxplot(mapping = aes(x = cond, y = totalPr))
```



Find the outliers

Find the outliers

- What are the outliers?

Find the outliers

- What are the outliers?
- Filter the dataset to isolate them

Find the outliers

- What are the outliers?
- Filter the dataset to isolate them

```
mariokart %>%
  filter(totalPr > 100) %>%
  glimpse()
```

```
## Observations: 2
## Variables: 12
## $ ID         <dbl> 110439174663, 130335427560
## $ duration   <int> 7, 3
## $ nBids      <int> 22, 27
## $ cond       <fct> used, used
## $ startPr    <dbl> 1.00, 6.95
## $ shipPr     <dbl> 25.51, 4.00
## $ totalPr    <dbl> 326.51, 118.50
## $ shipSp     <fct> parcel, parcel
## $ sellerRate <int> 115, 41
## $ stockPhoto <fct> no, no
## $ wheels     <int> 2, 0
## $ title      <fct> Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart...
```

Inspect outlier characteristics

Inspect outlier characteristics

- Look at the listing titles

Inspect outlier characteristics

- Look at the listing titles

```
mariokart %>%
  filter(totalPr > 100) %>%
  select(title) %>%
  head()
```

title
Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc

Inspect outlier characteristics

- Look at the listing titles

```
mariokart %>%
  filter(totalPr > 100) %>%
  select(title) %>%
  head()
```

title
Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc

- These are bundled items, not like the rest of the items in the dataset.

Inspect outlier characteristics

- Look at the listing titles

```
mariokart %>%
  filter(totalPr > 100) %>%
  select(title) %>%
  head()
```

title
Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc

- These are bundled items, not like the rest of the items in the dataset.

- Let's remove the outliers

Inspect outlier characteristics

- Look at the listing titles

```
mariokart %>%
  filter(totalPr > 100) %>%
  select(title) %>%
  head()
```

title
Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc

- These are bundled items, not like the rest of the items in the dataset.
- Let's remove the outliers
- For simplicity, we will also restrict ourselves to a subset of variables: `cond`, `stockPhoto`, `duration`, and `wheels`

Removing outliers

```
mariokart2 <- mariokart %>%
  filter(totalPr <= 100) %>%
  select(totalPr, cond, stockPhoto, duration, wheels)
```

Removing outliers

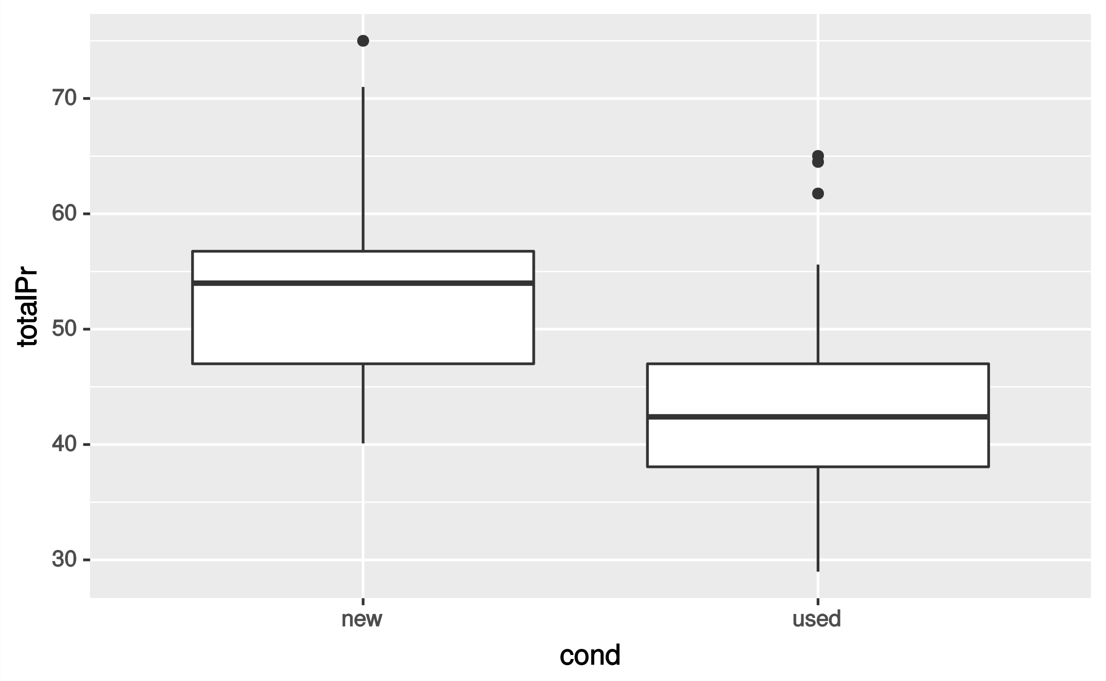
```
mariokart2 <- mariokart %>%
  filter(totalPr <= 100) %>%
  select(totalPr, cond, stockPhoto, duration, wheels)
```

- Let's check the box plot again, this time with no outliers

Removing outliers

```
mariokart2 <- mariokart %>%
  filter(totalPr <= 100) %>%
  select(totalPr, cond, stockPhoto, duration, wheels)
```

- Let's check the box plot again, this time with no outliers



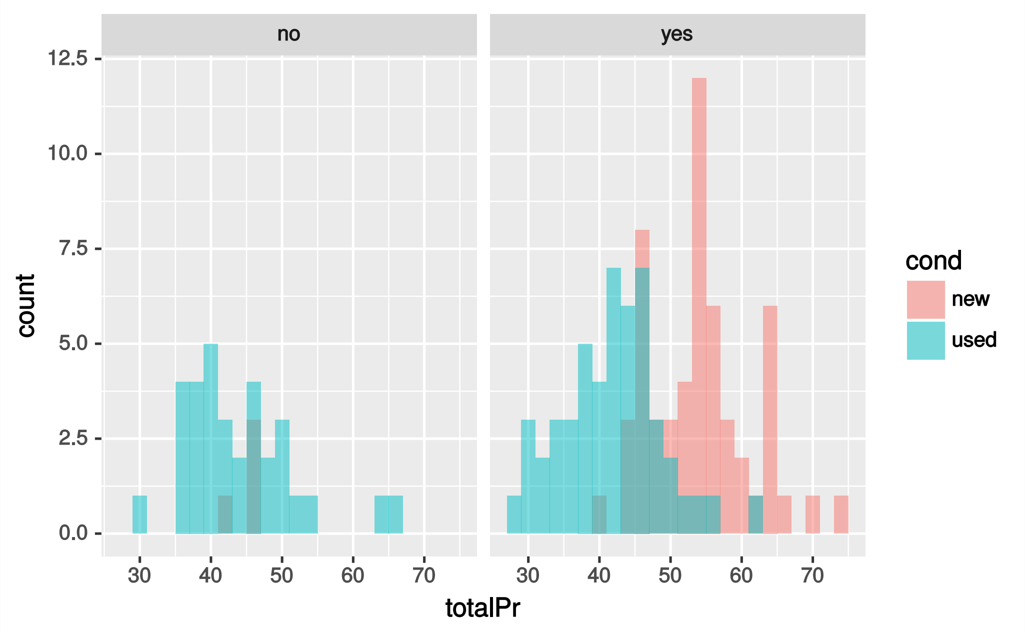
Looking for trends

- Continue exploring the dataset to find trends: does game condition and using a stock photo affect the total price?

Looking for trends

- Continue exploring the dataset to find trends: does game condition and using a stock photo affect the total price?

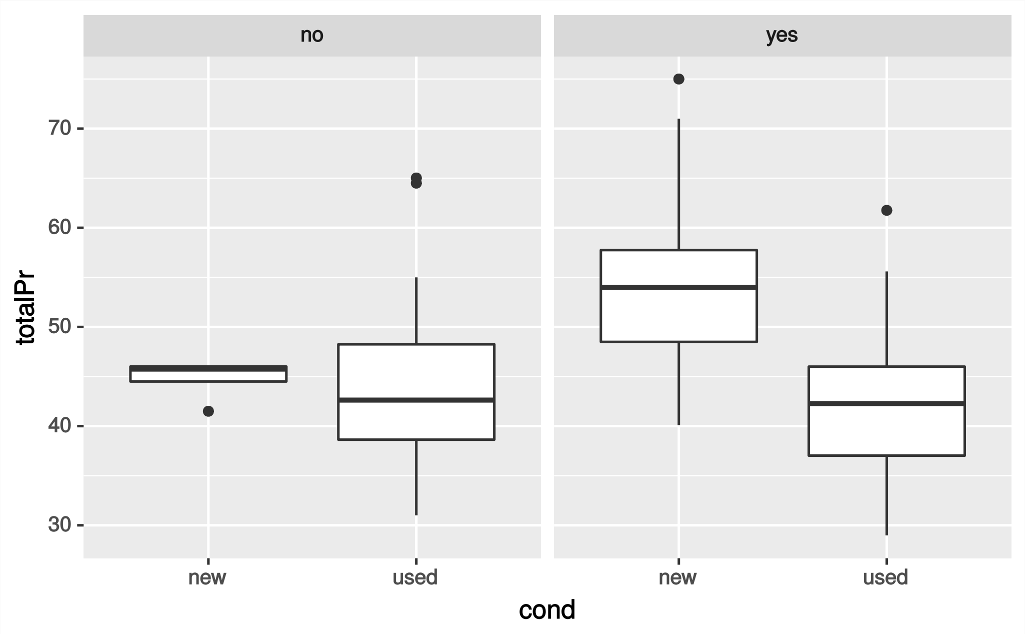
```
ggplot(mariokart2) +
  geom_histogram(
    mapping = aes(totalPr, fill = cond), position = "identity",
    alpha = 0.5, center = 0, binwidth = 2) +
  facet_wrap(~stockPhoto)
```



Looking for trends

- A box plot would also be an appropriate way to show this data:

```
ggplot(mariokart2) +
  geom_boxplot(mapping = aes(x = cond, y = totalPr)) +
  facet_wrap(~stockPhoto)
```



Data distribution of `totalPr`

Data distribution of `totalPr`

- Is `totalPr` nearly normal?

Data distribution of `totalPr`

- Is `totalPr` nearly normal?
- How does the distribution shape change within categories?

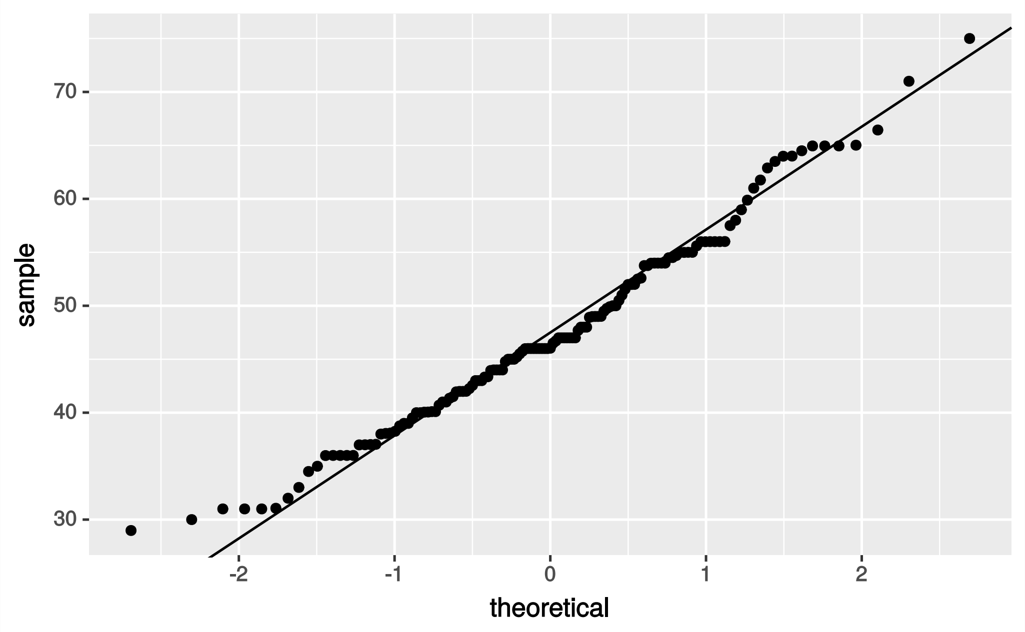
Data distribution of `totalPr`

- Is `totalPr` nearly normal?
- How does the distribution shape change within categories?
- Use Q-Q plot to check `totalPr` by itself:

Data distribution of `totalPr`

- Is `totalPr` nearly normal?
- How does the distribution shape change within categories?
- Use Q-Q plot to check `totalPr` by itself:

```
ggplot(mariokart2) +
  geom_qq(mapping = aes(sample = totalPr))
```



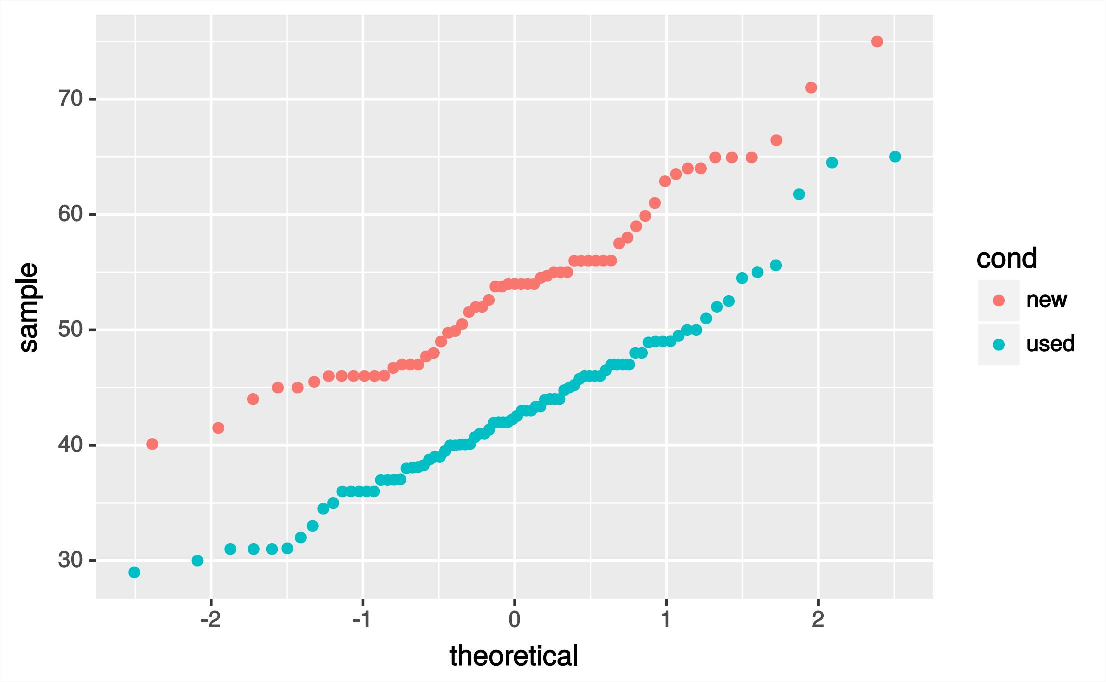
`totalPr` distribution within groups

- Q-Q plot with `totalPr` split by game condition:

`totalPr` distribution within groups

- Q-Q plot with `totalPr` split by game condition:

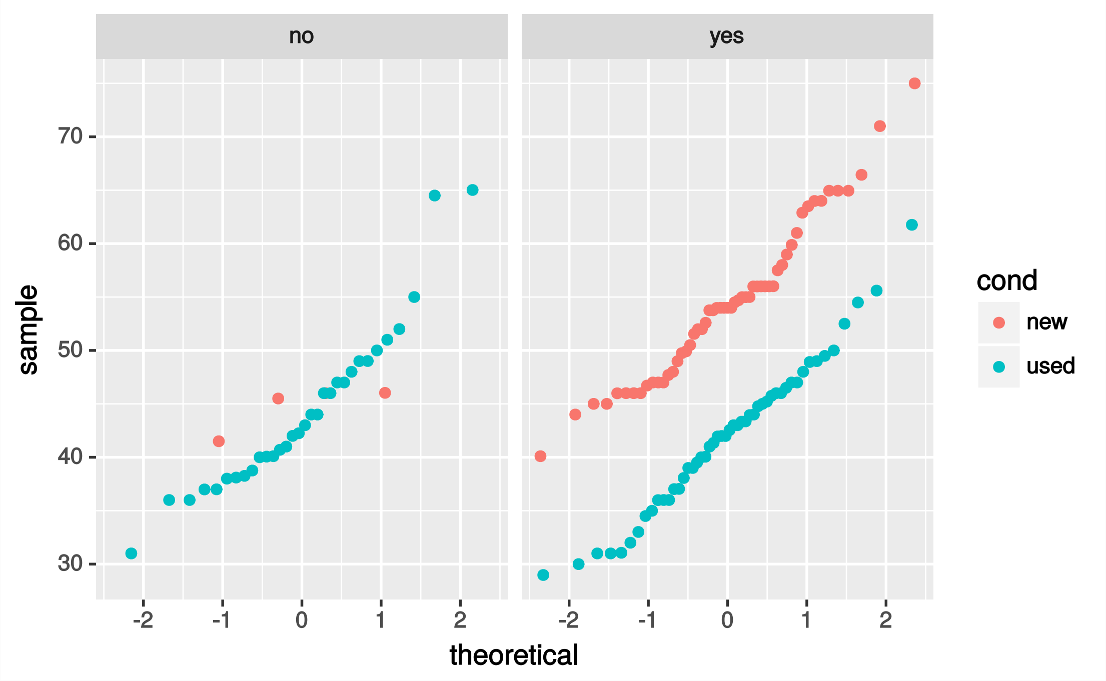
```
ggplot(mariokart2) +
  geom_qq(mapping = aes(sample = totalPr, color = cond))
```



`totalPr` distribution within groups

- Q-Q plot with `totalPr` split by game condition and faceted by `stockPhoto`:

```
ggplot(mariokart2) +
  geom_qq(mapping = aes(sample = totalPr, color = cond)) +
  facet_wrap( ~ stockPhoto)
```



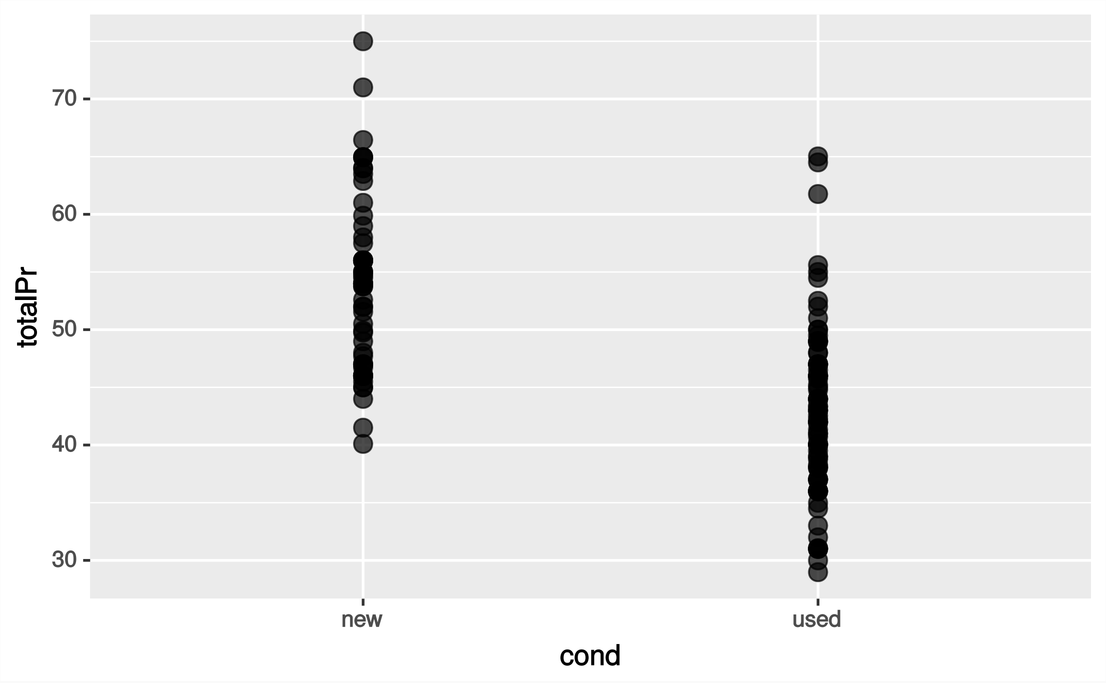
Categorical variables in scatterplots

- What happens if we plot `totalPr` as a function of `cond`, a categorical variable?

Categorical variables in scatterplots

- What happens if we plot `totalPr` as a function of `cond`, a categorical variable?

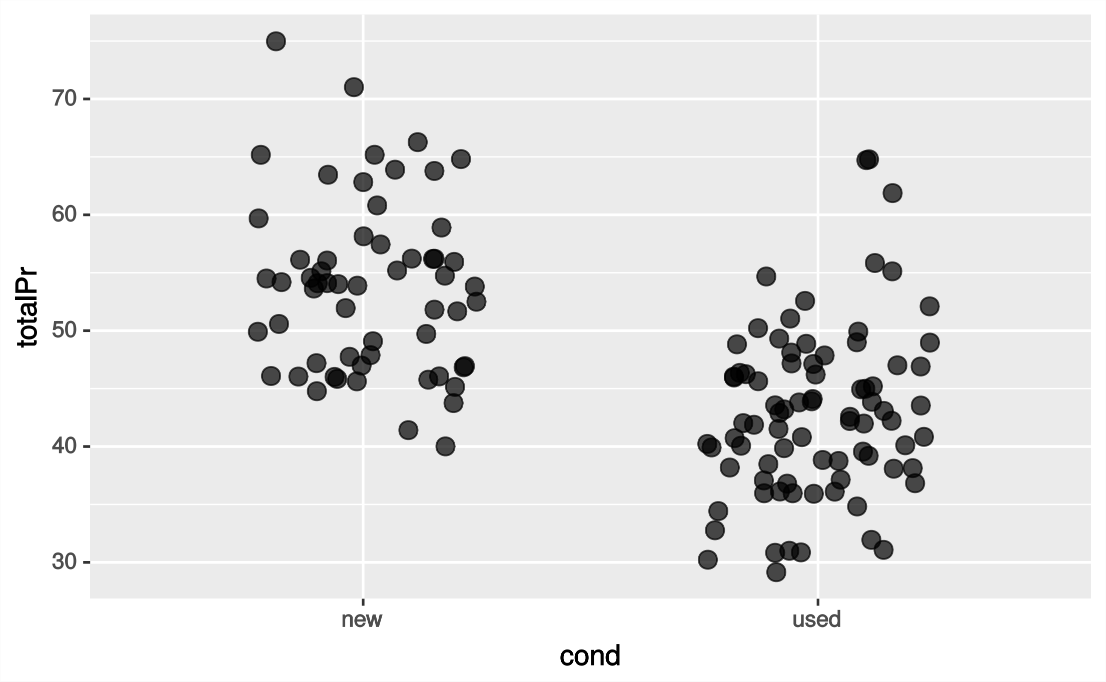
```
ggplot(mariokart2) +
  geom_point(mapping = aes(cond, totalPr), size = 3, alpha = 0.7)
```



Categorical variables in scatterplots

- It's easier to see the points if we jitter them

```
ggplot(mariokart2) +
  geom_jitter(
    mapping = aes(cond, totalPr), size = 3, alpha = 0.7, width = 0.25,
    height = 0.25)
```



Training and testing datasets

Split dataset 80/20

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.
- The following code splits the data into two partitions

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.
- The following code splits the data into two partitions

```
mariokart_with_ids <- mariokart2 %>%
 bind_cols(id = 1:nrow(mariokart2))

train <- mariokart_with_ids %>%
  sample_frac(size = 0.80, replace = FALSE)

test  <- mariokart_with_ids %>%
  anti_join(train, by = 'id')
```

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.
- The following code splits the data into two partitions

```
mariokart_with_ids <- mariokart2 %>%
 bind_cols(id = 1:nrow(mariokart2))

train <- mariokart_with_ids %>%
  sample_frac(size = 0.80, replace = FALSE)

test  <- mariokart_with_ids %>%
  anti_join(train, by = 'id')
```

- 80% is randomly selected and placed in the training dataset

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.
- The following code splits the data into two partitions

```
mariokart_with_ids <- mariokart2 %>%
 bind_cols(id = 1:nrow(mariokart2))

train <- mariokart_with_ids %>%
  sample_frac(size = 0.80, replace = FALSE)

test  <- mariokart_with_ids %>%
  anti_join(train, by = 'id')
```

- 80% is randomly selected and placed in the training dataset
- Remaining 20% is used for the testing dataset

Split dataset 80/20

- Frequently, it's good practice to split a dataset prior to testing a model.
- The following code splits the data into two partitions

```
mariokart_with_ids <- mariokart2 %>%
 bind_cols(id = 1:nrow(mariokart2))

train <- mariokart_with_ids %>%
  sample_frac(size = 0.80, replace = FALSE)

test  <- mariokart_with_ids %>%
  anti_join(train, by = 'id')
```

- 80% is randomly selected and placed in the training dataset
- Remaining 20% is used for the testing dataset
- All subsequent model building will be done using the `train` dataset

Univariate linear regression models

Predict using game condition

- Let's start with a refresher on creating a univariate linear model using `lm()`

Predict using game condition

- Let's start with a refresher on creating a univariate linear model using `lm()`
- Build a model that uses the `cond` categorical variable to predict the total price `totalPr`

Predict using game condition

- Let's start with a refresher on creating a univariate linear model using `lm()`
- Build a model that uses the `cond` categorical variable to predict the total price `totalPr`

```
mariokart_cond_model_lm <- lm(totalPr ~ cond, data = train)
```

Predict using game condition

- Let's start with a refresher on creating a univariate linear model using `lm()`
- Build a model that uses the `cond` categorical variable to predict the total price `totalPr`

```
mariokart_cond_model_lm <- lm(totalPr ~ cond, data = train)
```

- Predict training dataset and compute the residuals

Predict using game condition

- Let's start with a refresher on creating a univariate linear model using `lm()`
- Build a model that uses the `cond` categorical variable to predict the total price `totalPr`

```
mariokart_cond_model_lm <- lm(totalPr ~ cond, data = train)
```

- Predict training dataset and compute the residuals

```
mariokart_cond_model_df <- train %>%
  add_predictions(mariokart_cond_model_lm) %>%
  add_residuals(mariokart_cond_model_lm)
```

Summary of our fit

Print out some basic details about the linear fit:

Summary of our fit

Print out some basic details about the linear fit:

```
summary(mariokart_cond_model_lm)
```

```
##
## Call:
## lm(formula = totalPr ~ cond, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5157  -5.4957   0.5043   3.5043  21.8156
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   53.184      1.117   47.62  < 2e-16 ***
## condused      -9.689      1.440   -6.73 7.75e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.491 on 111 degrees of freedom
## Multiple R-squared:  0.2898,    Adjusted R-squared:  0.2834
## F-statistic:  45.3 on 1 and 111 DF,  p-value: 7.753e-10
```

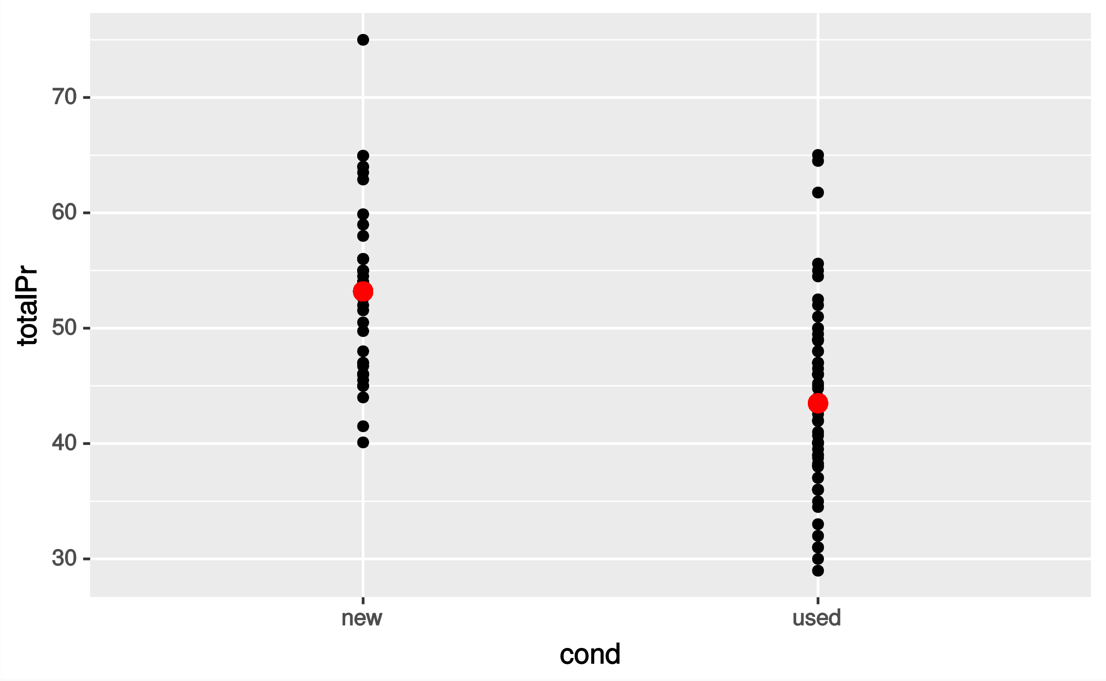
Visualize the model

- Since `cond` is categorical, what will it look like when we overlay our models' predictions on the data?

Visualize the model

- Since `cond` is categorical, what will it look like when we overlay our models' predictions on the data?

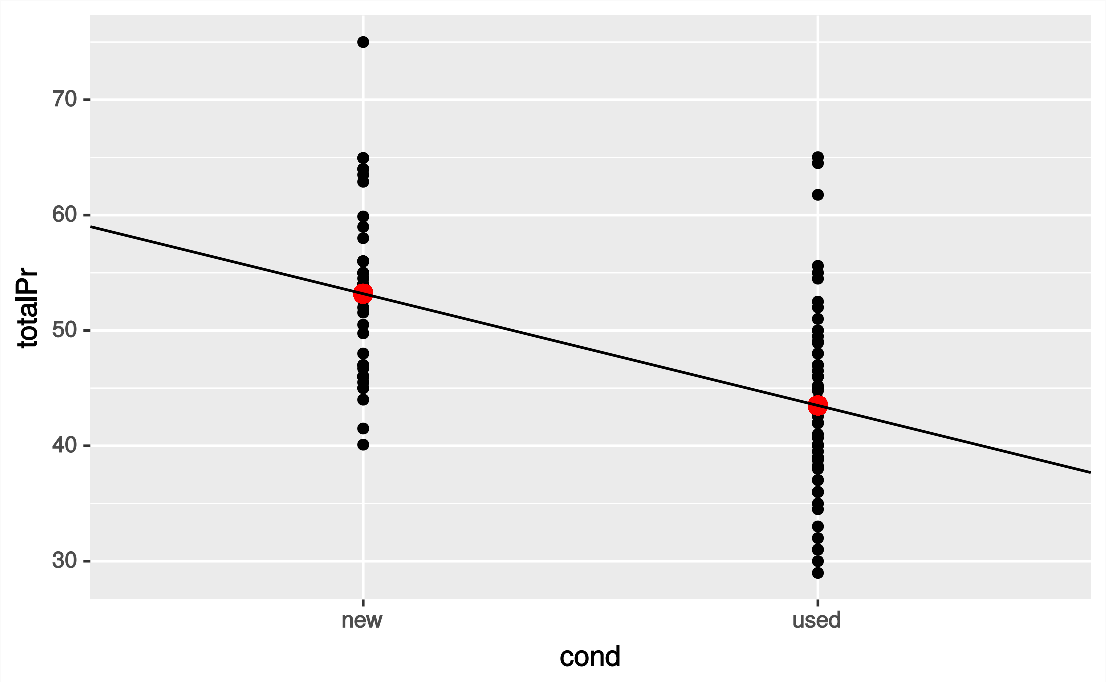
```
ggplot(mariokart_cond_model_df) +
  geom_point(mapping = aes(x = cond, y = totalPr)) +
  geom_point(mapping = aes(x = cond, y = pred), color = "red", size = 3)
```



Visualize the model

- Since `cond` is categorical, what will it look like when we overlay our models' predictions on the data?

```
ggplot(mariokart_cond_model_df) +
  geom_point(mapping = aes(x = cond, y = totalPr)) +
  geom_point(mapping = aes(x = cond, y = pred), color = "red", size = 3)
```



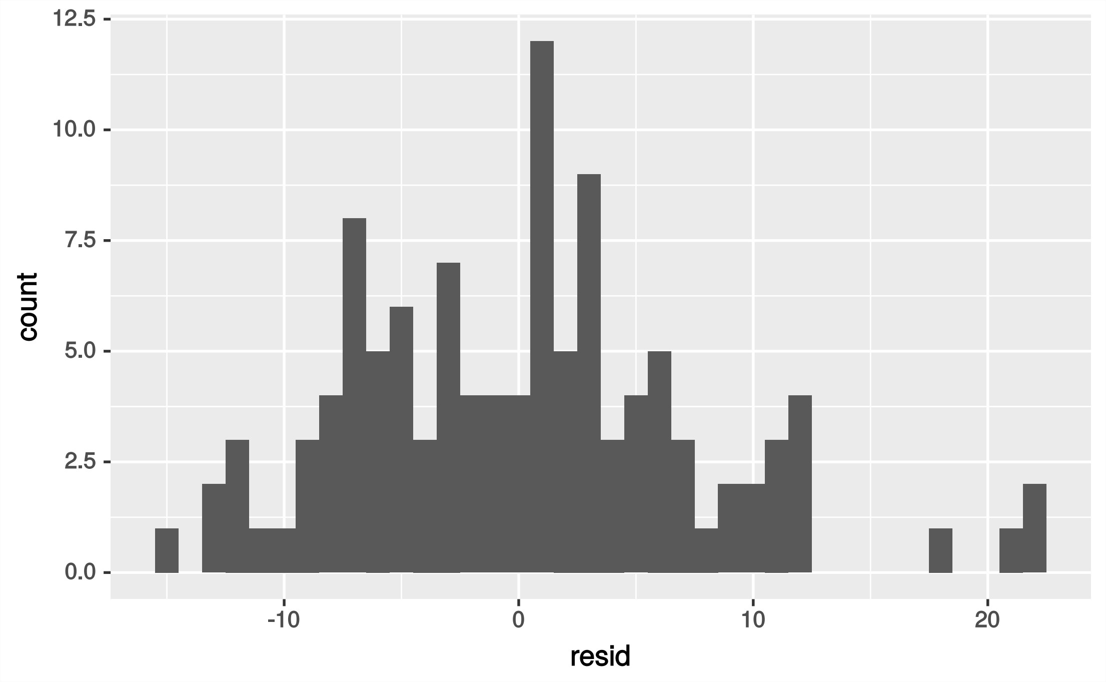
Inspect residuals

- Let's inspect the residuals:

Inspect residuals

- Let's inspect the residuals:

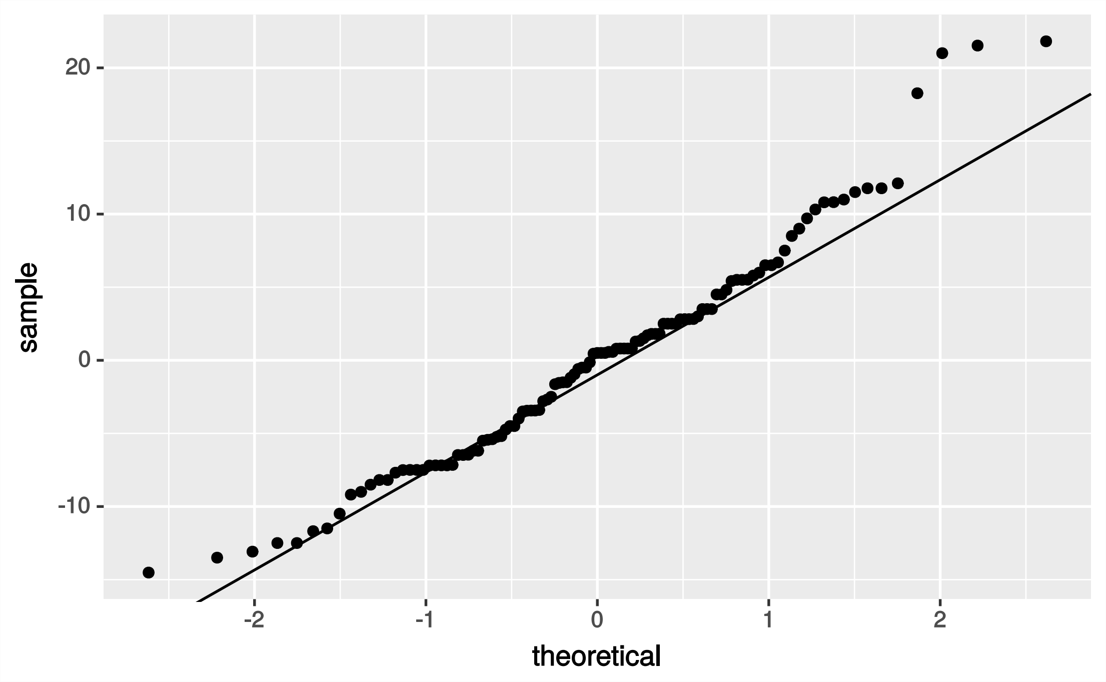
```
ggplot(mariokart_cond_model_df) +
  geom_histogram(mapping = aes(x = resid), binwidth = 1, center = 0)
```



Inspect residuals

- Let's inspect the residuals:

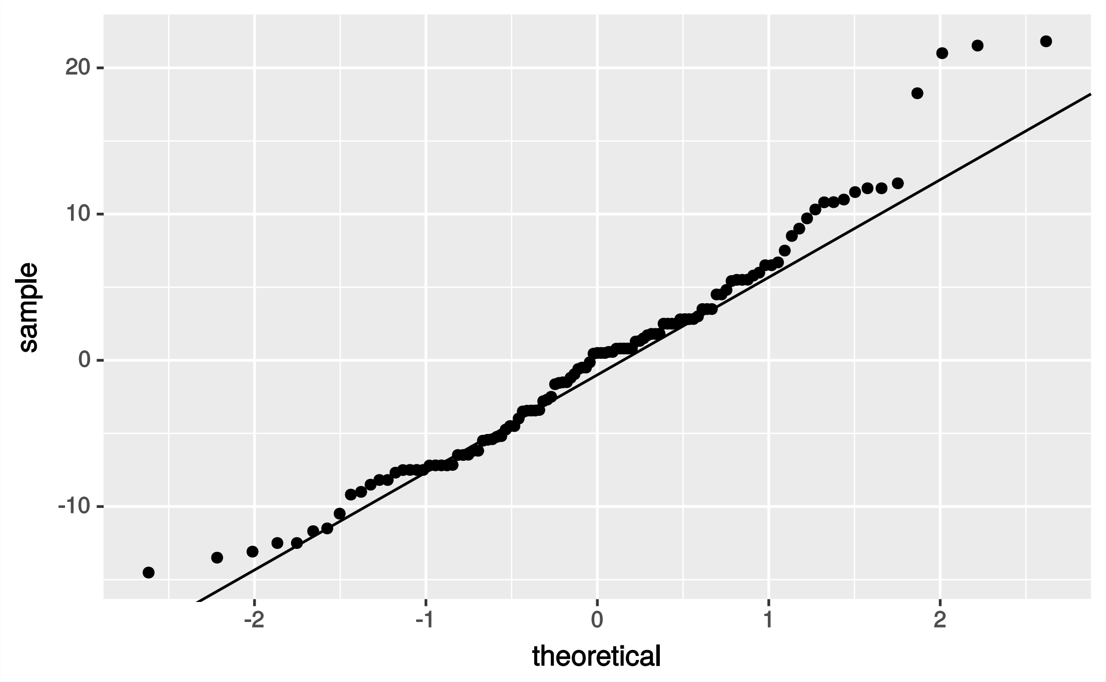
```
ggplot(mariokart_cond_model_df) +
  geom_qq(mapping = aes(sample = resid))
```



Inspect residuals

- Let's inspect the residuals:

```
ggplot(mariokart_cond_model_df) +
  geom_qq(mapping = aes(sample = resid))
```



- Deviations from normal distribution with long tail on the right

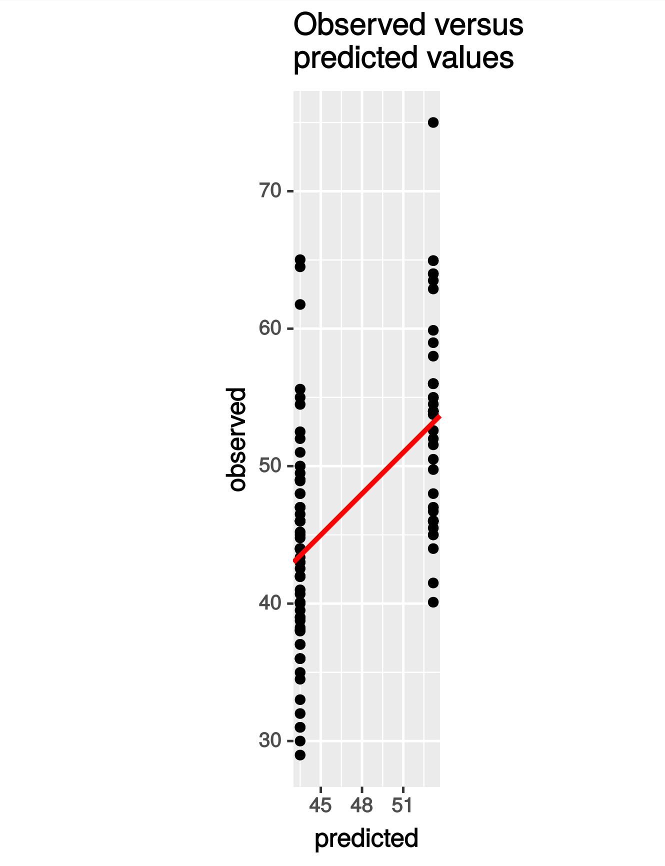
Observed values vs. predicted values

- Accurate prediction is our goal, so we should visualize how well the predictions match with the actual values

Observed values vs. predicted values

- Accurate prediction is our goal, so we should visualize how well the predictions match with the actual values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(totalPr, pred)) +
  geom_abline(
    slope = 1, intercept = 0,
    color = "red", size = 1)
```

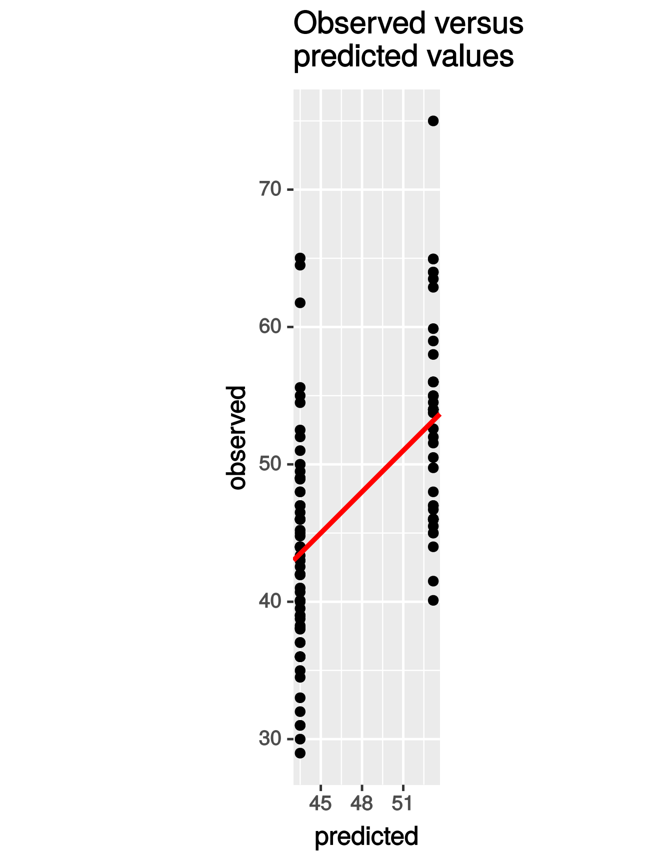


Observed values vs. predicted values

- Accurate prediction is our goal, so we should visualize how well the predictions match with the actual values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(totalPr, pred)) +
  geom_abline(
    slope = 1, intercept = 0,
    color = "red", size = 1)
```

- This is called an "observed versus predicted" plot†



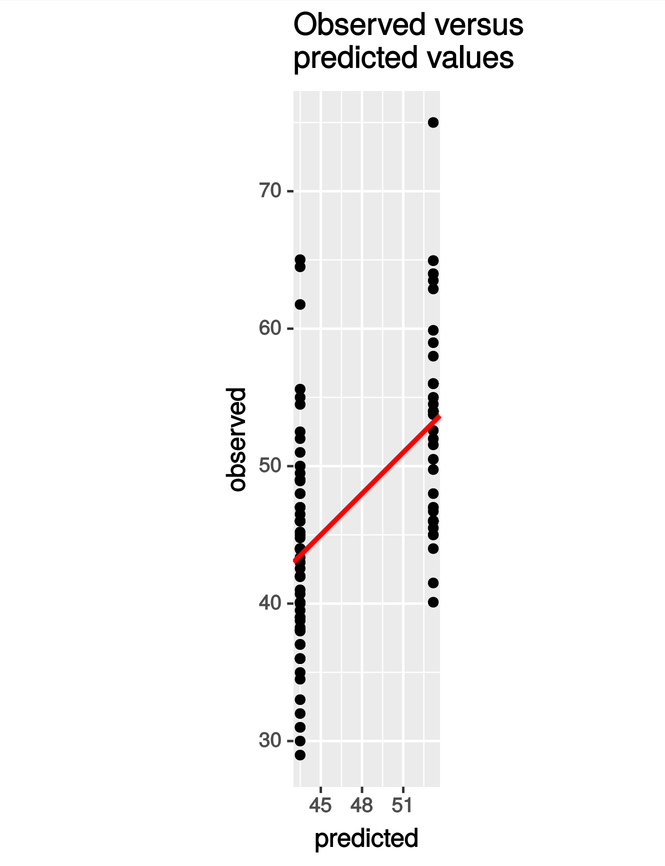
† There isn't a precise name for this type of plot, so you may see this called an "actual versus predicted" plot or an "actual versus fitted" plot, or something else.

Observed values vs. predicted values

- Accurate prediction is our goal, so we should visualize how well the predictions match with the actual values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(totalPr, pred)) +
  geom_abline(
    slope = 1, intercept = 0,
    color = "red", size = 1)
```

- This is called an "observed versus predicted" plot†
- There's a residuals version of this, the "residual versus predicted" plot



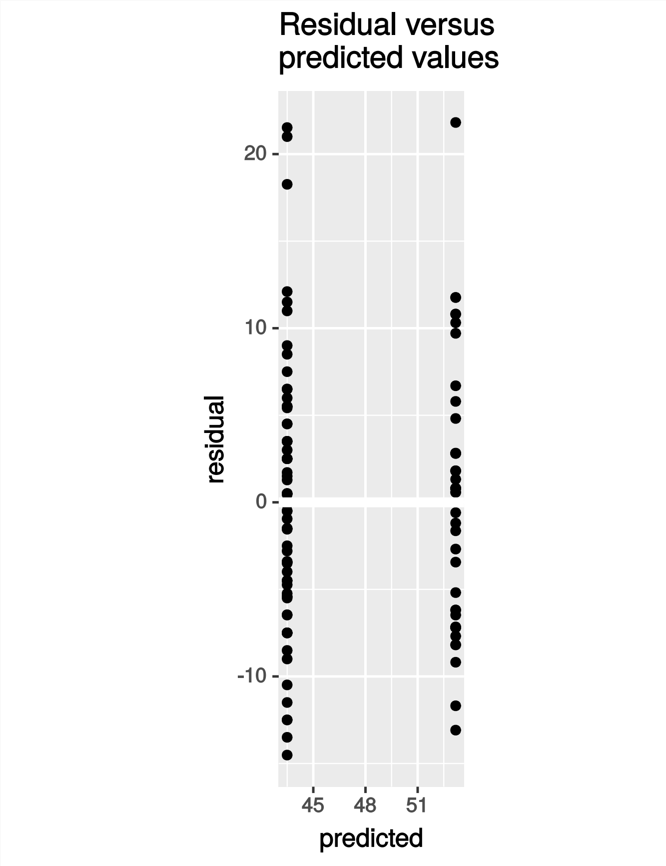
† There isn't a precise name for this type of plot, so you may see this called an "actual versus predicted" plot or an "actual versus fitted" plot, or something else.

Residual vs. predicted values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

Residual vs. predicted values

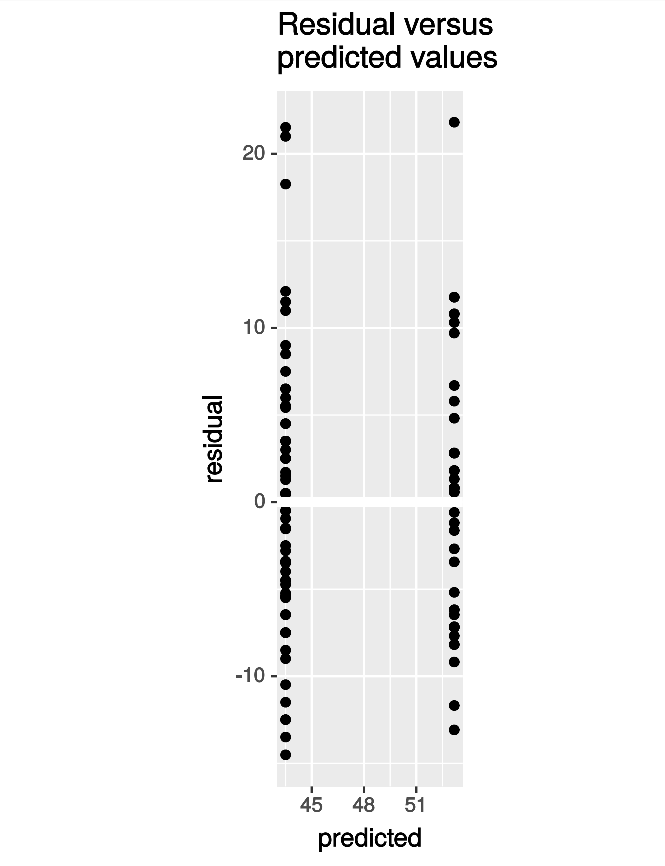
```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```



Residual vs. predicted values

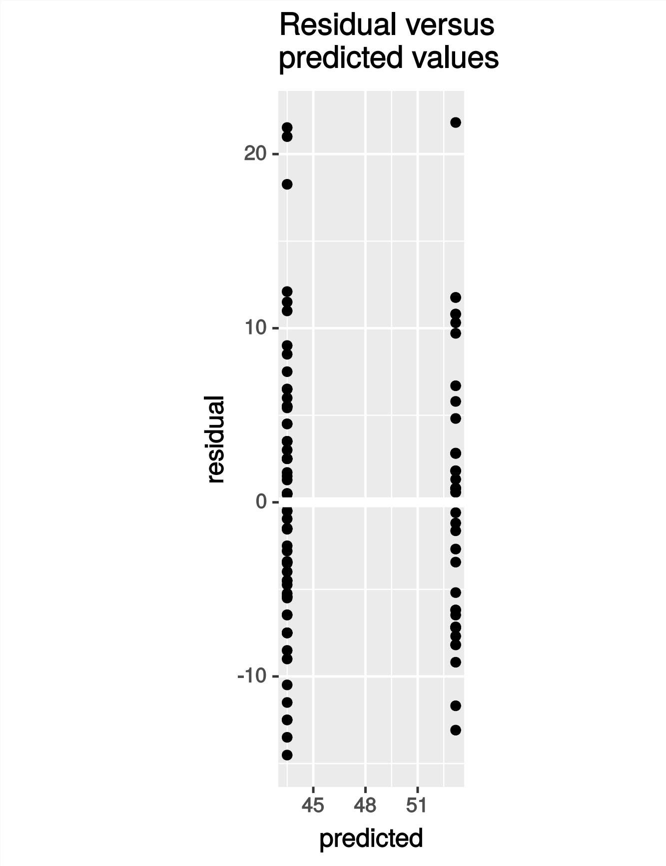
```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

- The residual spread stays consistent, so that's good



Residual vs. predicted values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

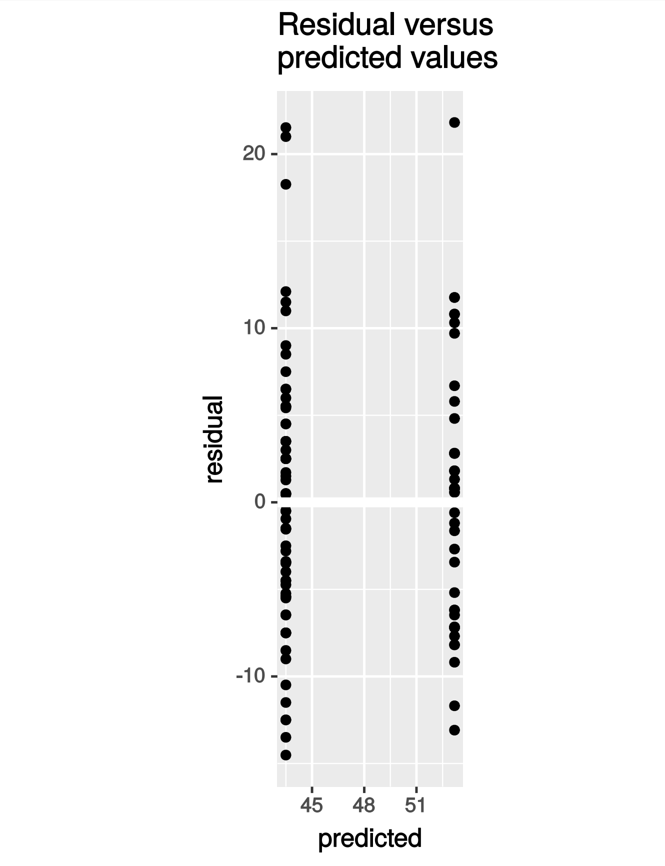


- The residual spread stays consistent, so that's good
- However, the long tails and this model's poor prediction ability are good enough reason to try and build a better model

Residual vs. predicted values

```r
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

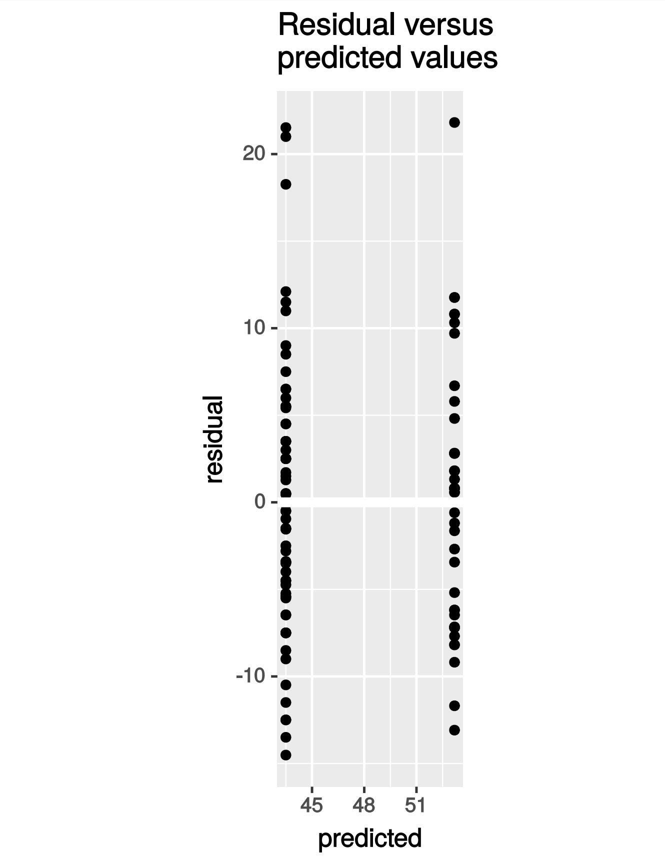
- The residual spread stays consistent, so that's good
- However, the long tails and this model's poor prediction ability are good enough reason to try and build a better model
- We can try building other univariate models with the other columns



Residual vs. predicted values

```
ggplot(mariokart_cond_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

- The residual spread stays consistent, so that's good
- However, the long tails and this model's poor prediction ability are good enough reason to try and build a better model
- We can try building other univariate models with the other columns
- However, as we'll find out, it's better to train **multivariate** models on this dataset



Multivariate linear regression models

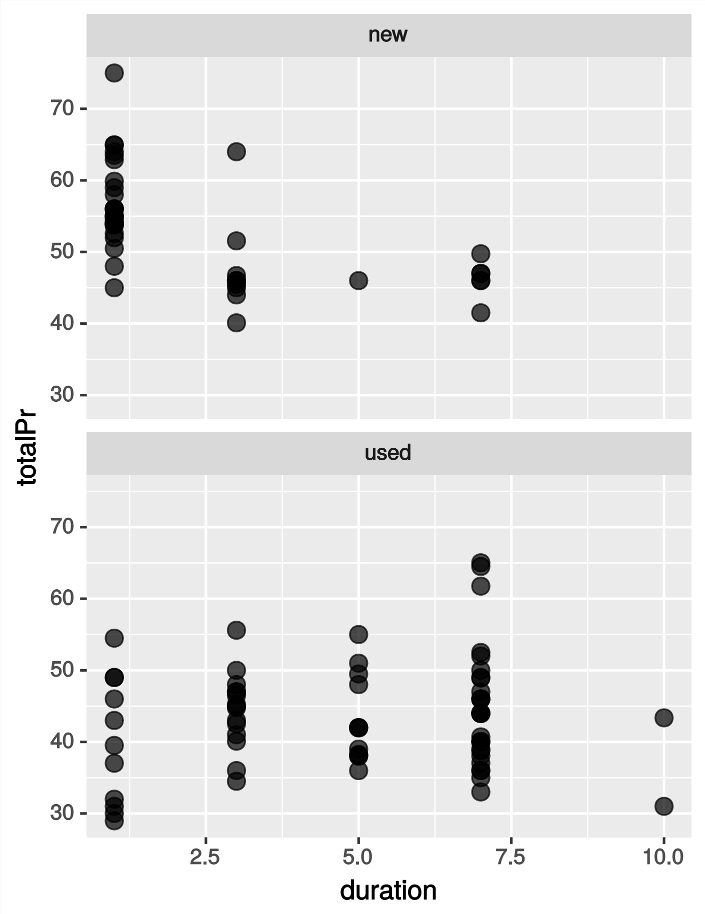
When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

```
ggplot(train) +
  geom_point(aes(duration, totalPr)) +
  facet_wrap(~cond, ncol = 1)
```

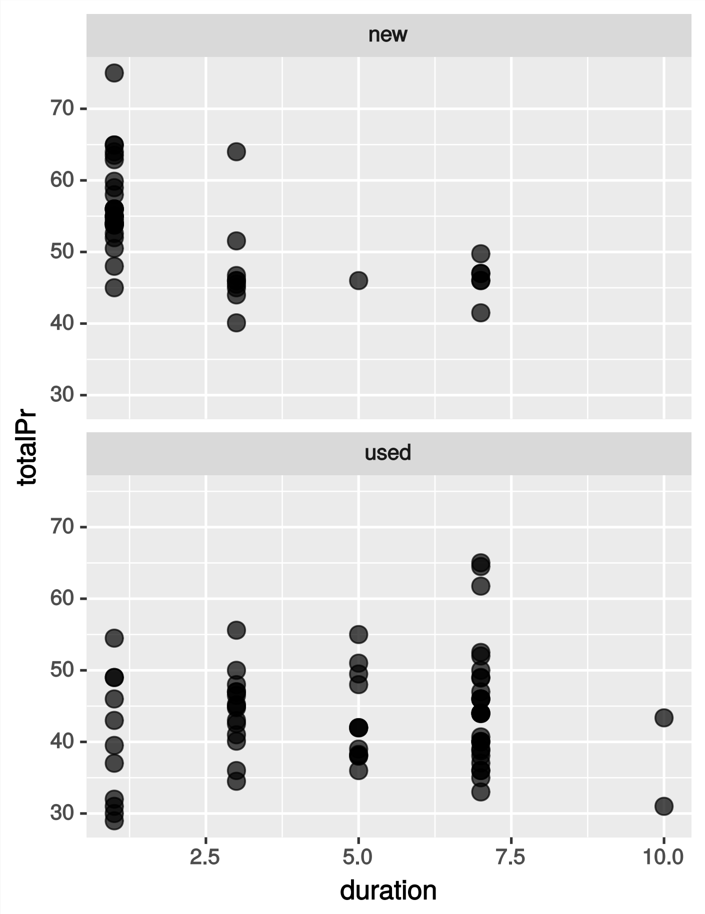


When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

```
ggplot(train) +
  geom_point(aes(duration, totalPr)) +
  facet_wrap(~cond, ncol = 1)
```

- There's a modest dependence of `duration` on `cond`, especially with new games of short duration



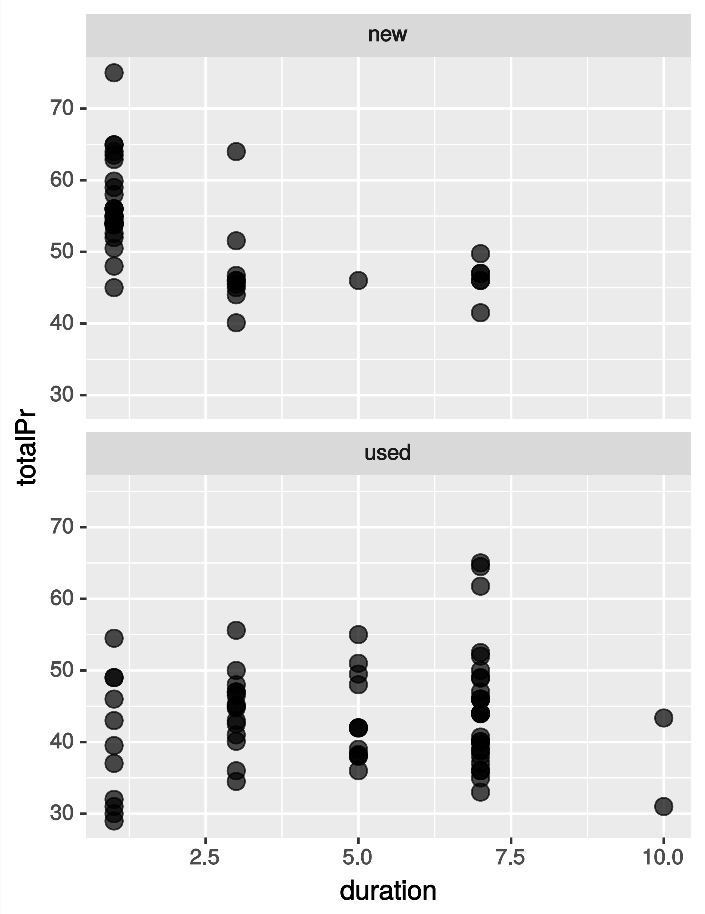
When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

```
ggplot(train) +
  geom_point(aes(duration, totalPr)) +
  facet_wrap(~cond, ncol = 1)
```

- There's a modest dependence of `duration` on `cond`, especially with new games of short duration

- **If independent:** you'd see same trend in both boxes, just shifted by a constant amount

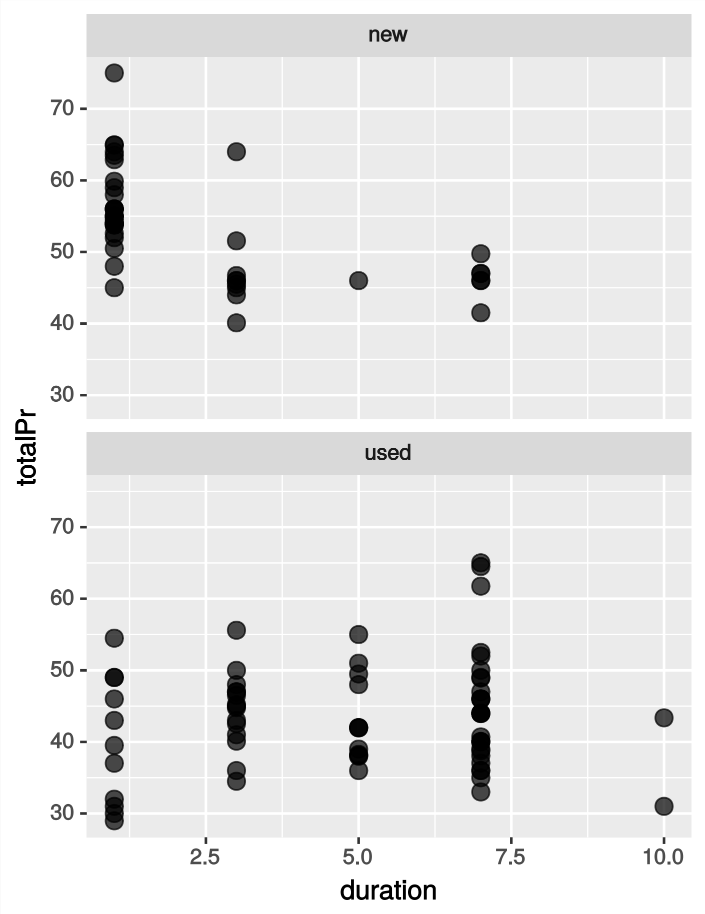


When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

```
ggplot(train) +
  geom_point(aes(duration, totalPr)) +
  facet_wrap(~cond, ncol = 1)
```

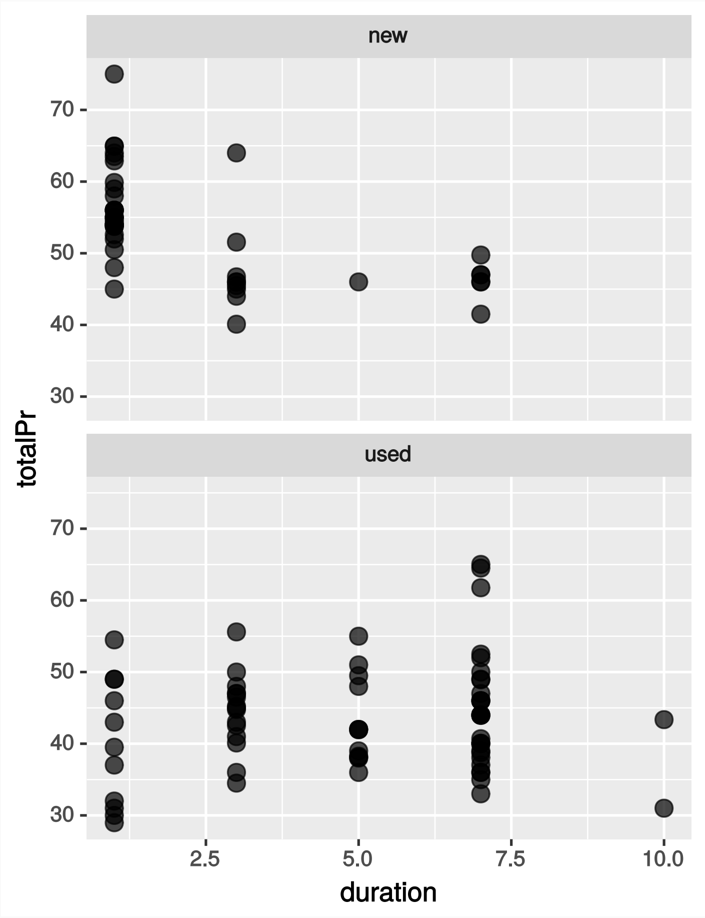
- There's a modest dependence of `duration` on `cond`, especially with new games of short duration
- **If independent:** you'd see same trend in both boxes, just shifted by a constant amount
- **If interacting:** different trends in both boxes, not just a constant shift



When many variables matter

Let's see how `cond` and `duration` affect `totalPr`:

```
ggplot(train) +
  geom_point(aes(duration, totalPr)) +
  facet_wrap(~cond, ncol = 1)
```



- There's a modest dependence of `duration` on `cond`, especially with new games of short duration
- **If independent:** you'd see same trend in both boxes, just shifted by a constant amount
- **If interacting:** different trends in both boxes, not just a constant shift
- Modest interaction between `cond` and `duration`, keep that in mind

Predicting price using four variables

Predicting price using four variables

- Build a linear model using the variables `cond`, `stockPhoto`, `duration`, and `wheels`

Predicting price using four variables

- Build a linear model using the variables `cond`, `stockPhoto`, `duration`, and `wheels`
- Variables are independent in this model and we do not consider interaction terms like `cond * duration`.

Predicting price using four variables

- Build a linear model using the variables `cond`, `stockPhoto`, `duration`, and `wheels`
- Variables are independent in this model and we do not consider interaction terms like `cond * duration`.

```
mariokart_multivar_model_lm <- lm(
  formula = totalPr ~ cond + stockPhoto + duration + wheels,
  data = train
)
```

Predicting price using four variables

- Build a linear model using the variables `cond`, `stockPhoto`, `duration`, and `wheels`
- Variables are independent in this model and we do not consider interaction terms like `cond * duration`.

```
mariokart_multivar_model_lm <- lm(
  formula = totalPr ~ cond + stockPhoto + duration + wheels,
  data = train
)
```

- Predict training dataset and compute the residuals

Predicting price using four variables

- Build a linear model using the variables `cond`, `stockPhoto`, `duration`, and `wheels`
- Variables are independent in this model and we do not consider interaction terms like `cond * duration`.

```
mariokart_multivar_model_lm <- lm(
  formula = totalPr ~ cond + stockPhoto + duration + wheels,
  data = train
)
```

- Predict training dataset and compute the residuals

```
mariokart_multivar_model_df <- train %>%
  add_predictions(mariokart_multivar_model_lm) %>%
  add_residuals(mariokart_multivar_model_lm)
```

Visualize the model...

- We do this just like last time, right? We can plot the model on top of a plot of the predictor variables.

Visualize the model...

- We do this just like last time, right? We can plot the model on top of a plot of the predictor variables.
- This would be possible...

Visualize the model...

- We do this just like last time, right? We can plot the model on top of a plot of the predictor variables.
- This would be possible... if we could create 5-dimensional images

Visualize the model...

- We do this just like last time, right? We can plot the model on top of a plot of the predictor variables.
- This would be possible... if we could create 5-dimensional images
- Use observed versus predicted and residual versus predicted plots like we created for the `totalPr ~ cond` model

Visualize the model...

- We do this just like last time, right? We can plot the model on top of a plot of the predictor variables.

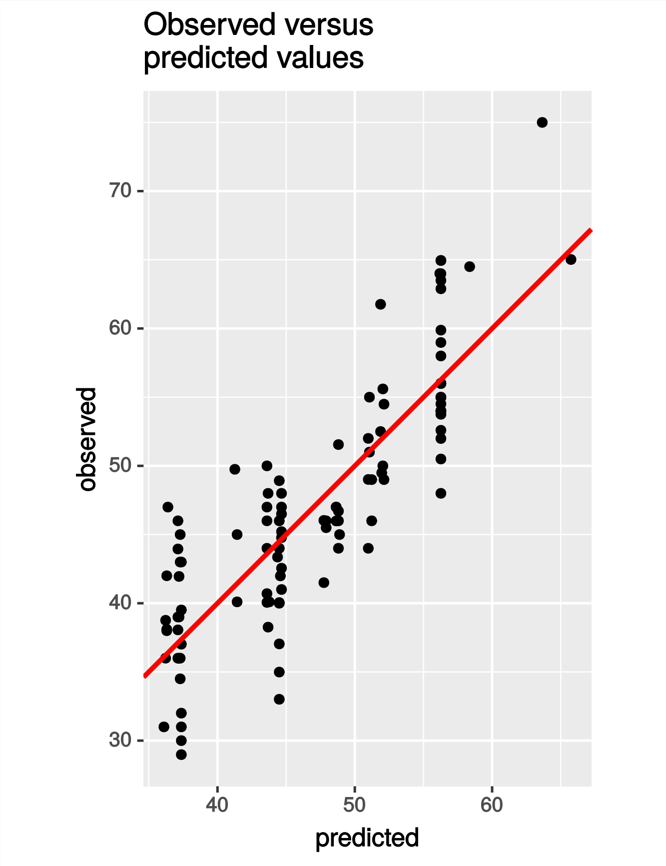
- This would be possible... if we could create 5-dimensional images

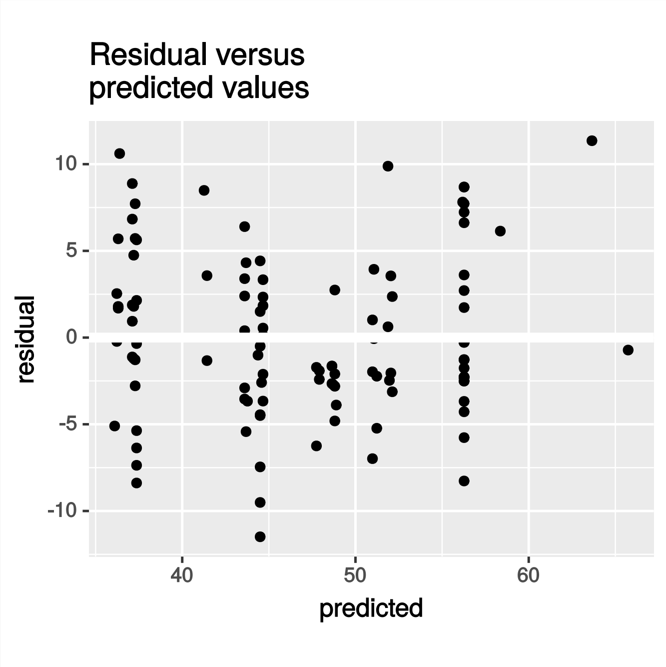
- Use observed versus predicted and residual versus predicted plots like we created for the `totalPr ~ cond` model

```
ggplot(mariokart_multivar_model_df) +
  geom_point(aes(pred, totalPr)) +
  geom_abline(slope = 1, intercept = 0, color = "red", size = 1)

ggplot(mariokart_multivar_model_df) +
  geom_point(aes(pred, resid)) +
  geom_ref_line(h = 0)
```

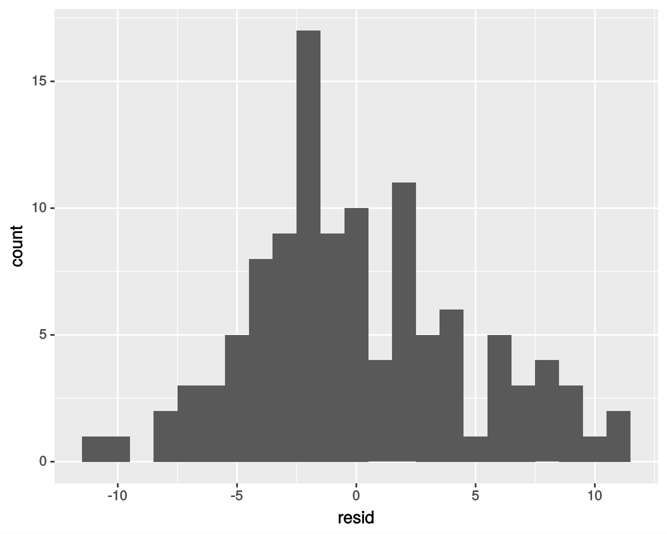
Multivariate model performance



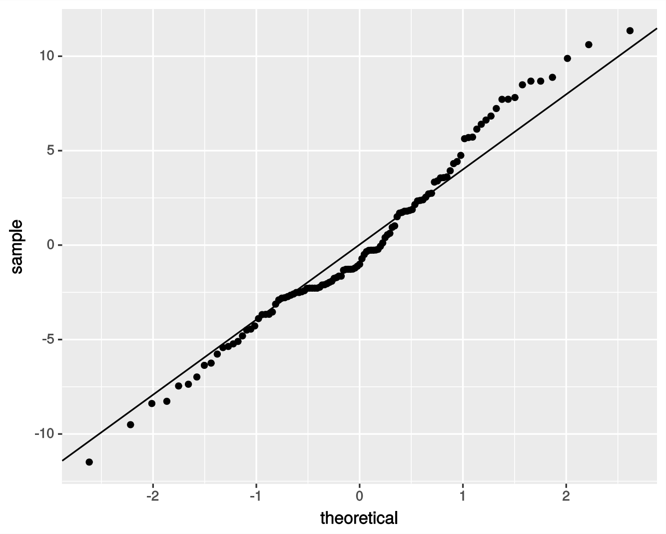


Inspect multivariate model residuals

```
ggplot(mariokart_multivar_model_df) +
  geom_histogram(
    mapping = aes(x = resid), binwidth = 1,
    center = 0)
```

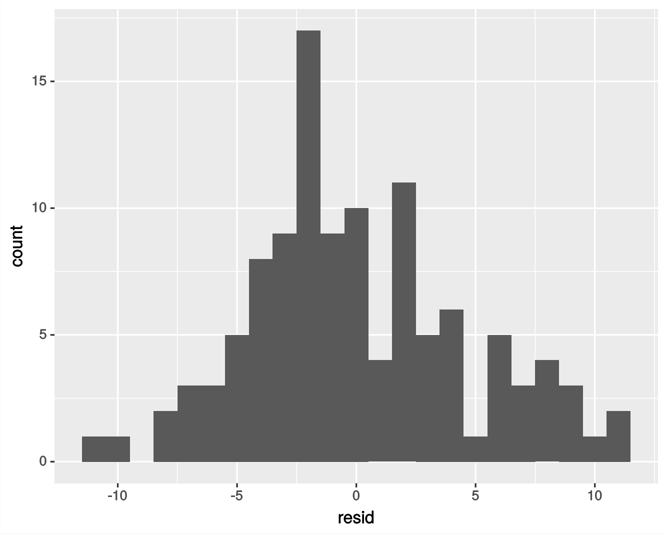



```
ggplot(mariokart_multivar_model_df) +
  geom_qq(mapping = aes(sample = resid))
```

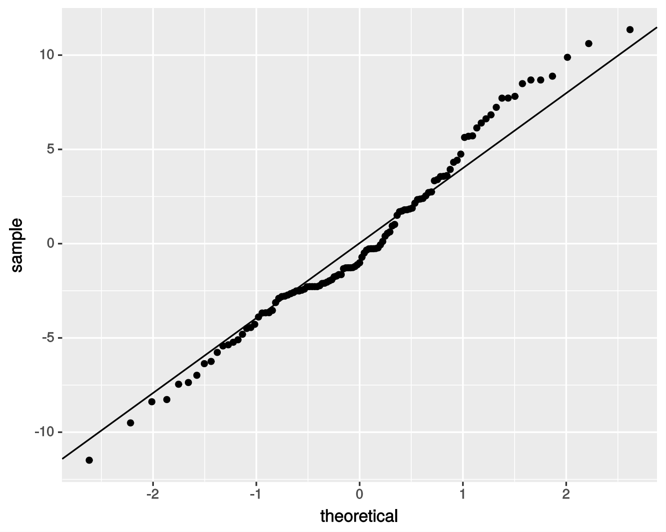


Inspect multivariate model residuals

```
ggplot(mariokart_multivar_model_df) +
  geom_histogram(
    mapping = aes(x = resid), binwidth = 1,
    center = 0)
```




```
ggplot(mariokart_multivar_model_df) +
  geom_qq(mapping = aes(sample = resid))
```



- Residuals still show deviations from the normal distribution on the right-side tail, but they're smaller overall

Comparing the two models

- Compare the residual histograms of the two models

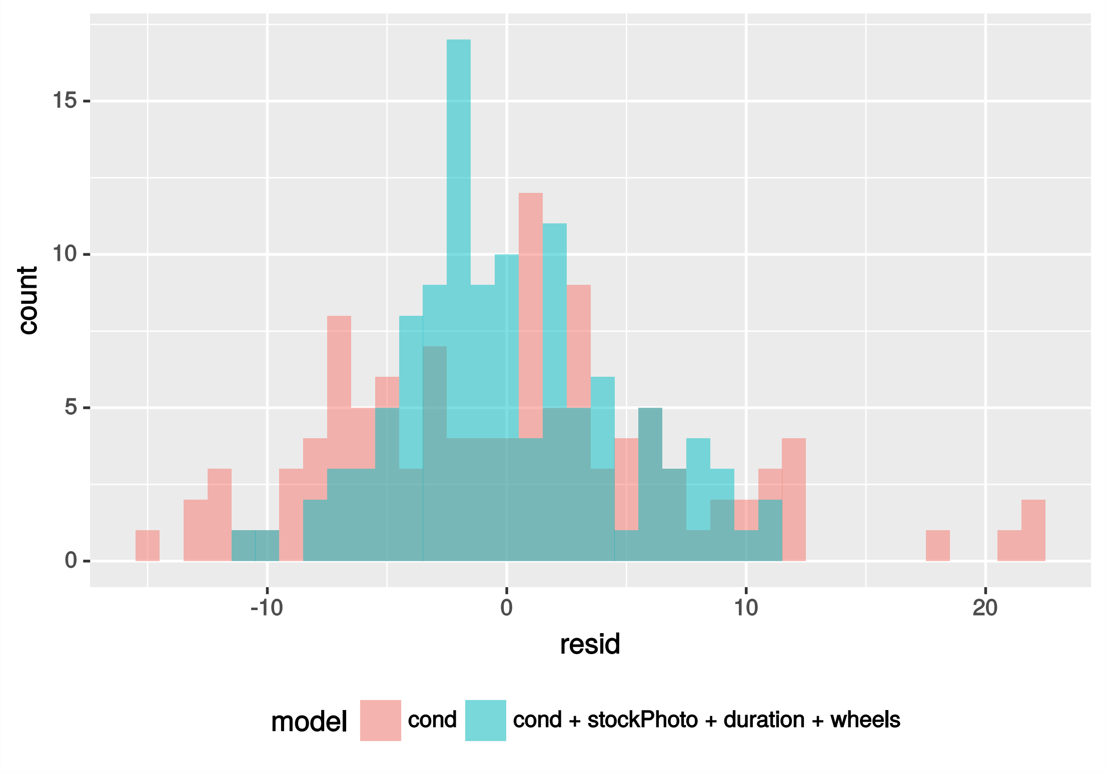
Comparing the two models

- Compare the residual histograms of the two models

```
data_frame(
  model = c(
    rep("cond", nrow(mariokart_cond_model_df)),
    rep(
      "cond + stockPhoto + duration + wheels",
      nrow(mariokart_multivar_model_df)
    )
  ),
  resid = c(
    pull(mariokart_cond_model_df, "resid"),
    pull(mariokart_multivar_model_df, "resid")
  )
) %>%
  ggplot() +
  geom_histogram(
    mapping = aes(x = resid, fill = model), alpha = 0.5, binwidth = 1,
    position = "identity", center = 0
  ) +
  theme(legend.position = "bottom")
```

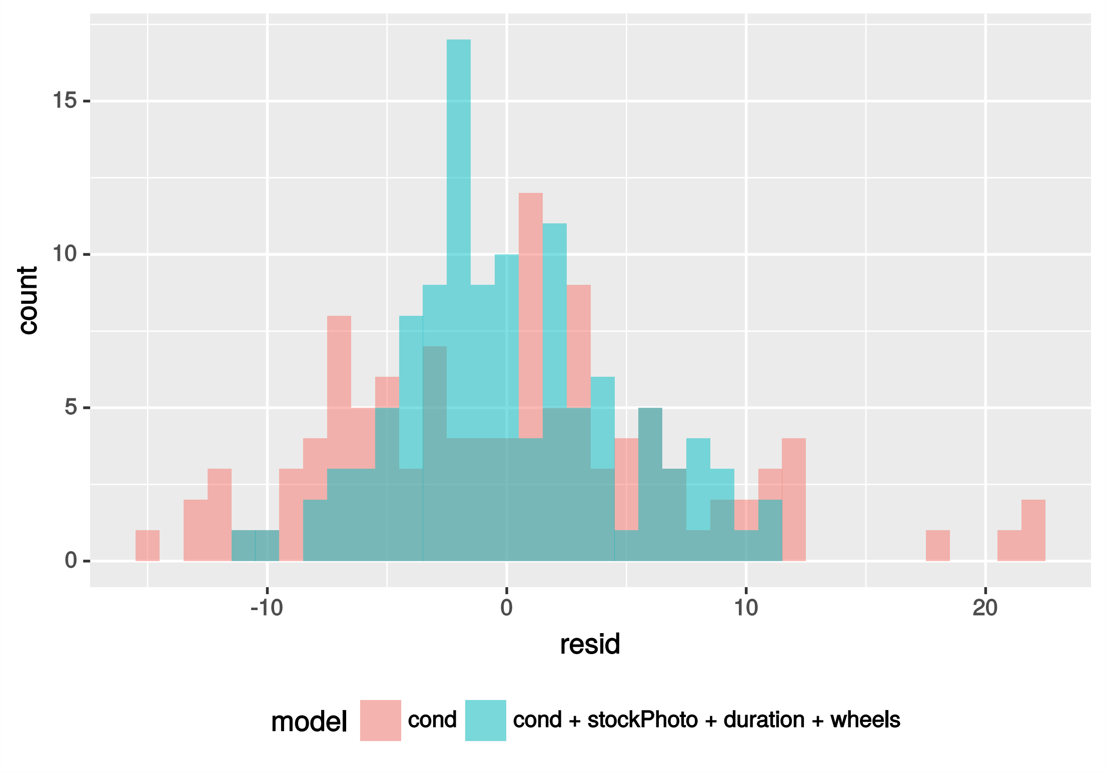
Comparing the two models

- Compare the residual histograms of the two models



Comparing the two models

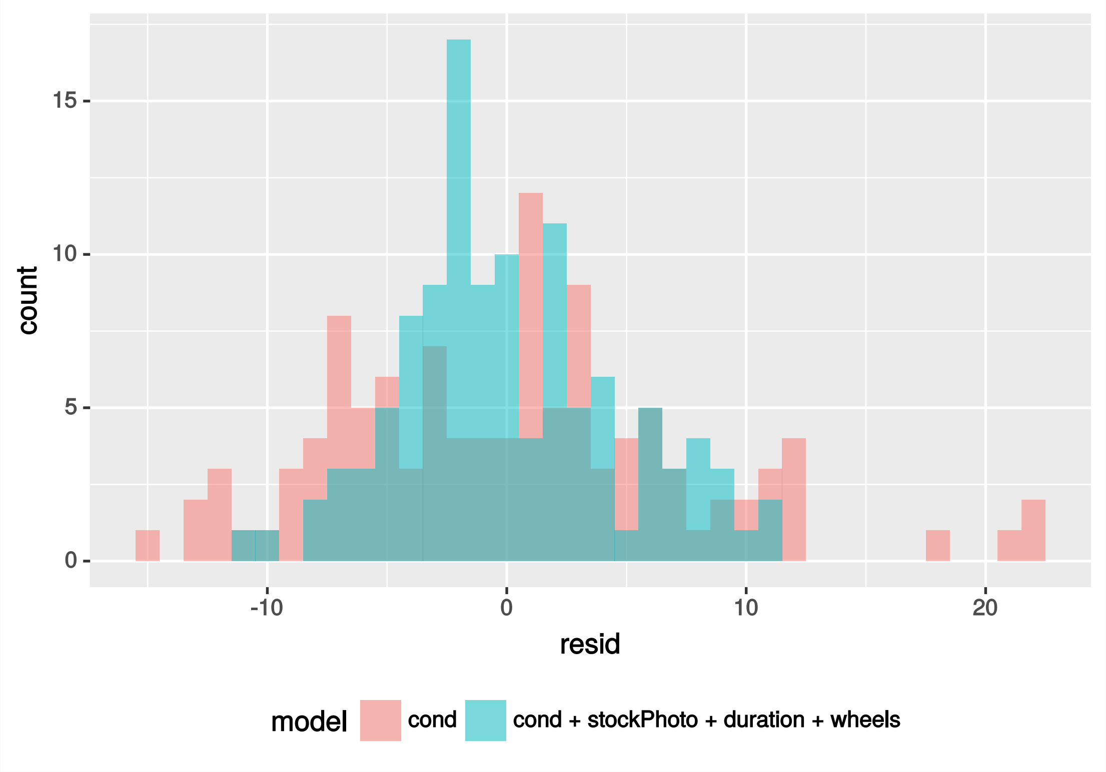
- Compare the residual histograms of the two models



- Multivariate model *seems* better

Comparing the two models

- Compare the residual histograms of the two models



- Multivariate model *seems* better, but it'd be better if we had an objective measure of model quality

Model selection

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure
- How should we compare and rank models?

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure
- How should we compare and rank models?
- This is what model selection is about, computing scores and measures of model performance for different models, and selecting the best choice.

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure

- How should we compare and rank models?

- This is what model selection is about, computing scores and measures of model performance for different models, and selecting the best choice.

- Bootstrapping is one option

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure
- How should we compare and rank models?
- This is what model selection is about, computing scores and measures of model performance for different models, and selecting the best choice.
- Bootstrapping is one option
- Cross-validation is another method that can compare relative model performance using only training data

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure
- How should we compare and rank models?
- This is what model selection is about, computing scores and measures of model performance for different models, and selecting the best choice.
- Bootstrapping is one option
- Cross-validation is another method that can compare relative model performance using only training data
- A popular flavor of cross-validation (especially among data scientists) is called **k-fold cross-validation**

Question, what kind of model is best?

- Comparing residuals can help us understand the relative performance of models, but it's just a qualitative measure
- How should we compare and rank models?
- This is what model selection is about, computing scores and measures of model performance for different models, and selecting the best choice.
- Bootstrapping is one option
- Cross-validation is another method that can compare relative model performance using only training data
- A popular flavor of cross-validation (especially among data scientists) is called **k-fold cross-validation**
- **Basic idea:** Estimate how robust your model is by systematically removing different chunks (the "folds") of the dataset, repeating the fitting process, then testing its predictive power on the folds

k-fold cross-validation

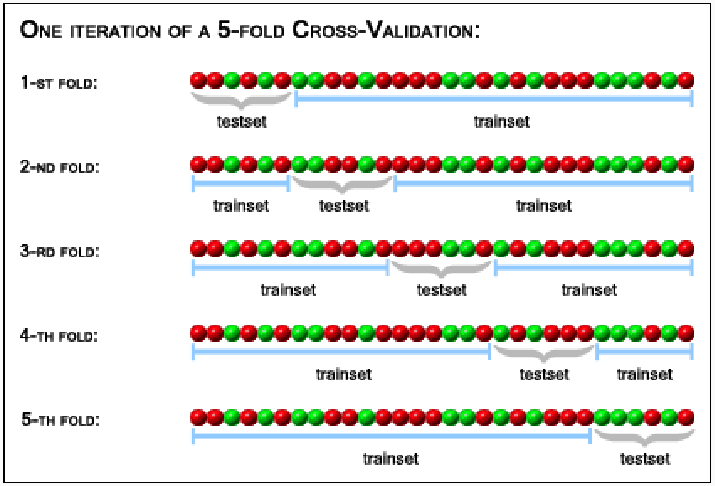
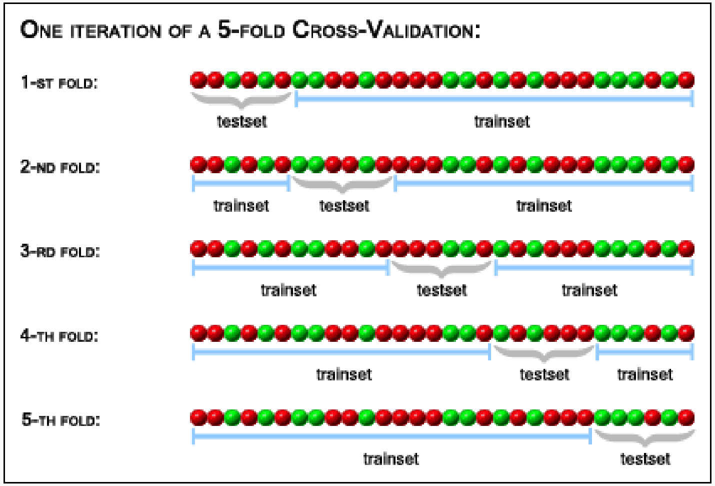


Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

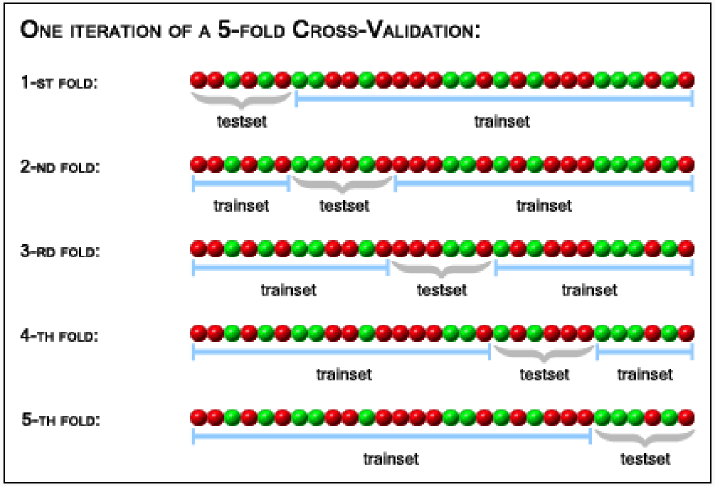
k-fold cross-validation



- The above example illustrates a *5-fold*, or $k = 5$, cross-validation.

Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

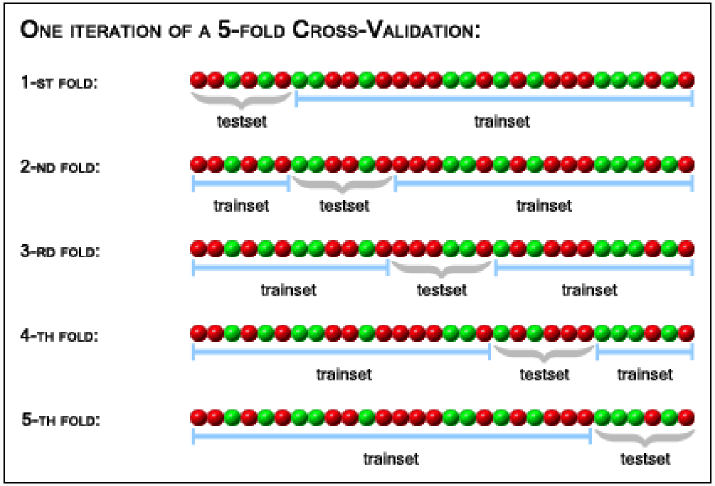
k-fold cross-validation



- The above example illustrates a *5-fold*, or $\boxed{k = 5}$, cross-validation.
- Each fold will act as a testing set, with the remaining $\boxed{k - 1}$ folds used to train the model.

Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

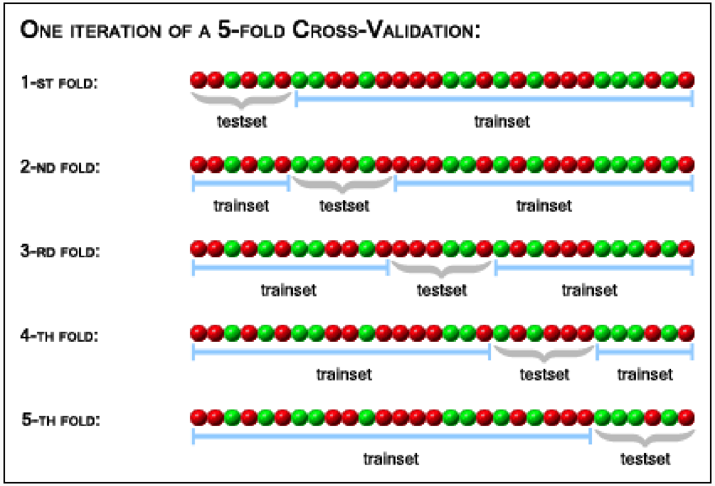
k-fold cross-validation



- The above example illustrates a *5-fold*, or $k = 5$, cross-validation.
- Each fold will act as a testing set, with the remaining $k - 1$ folds used to train the model.
- Fit model, predict values in testing set, then calculate the mean-squared prediction error (MSE)

Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

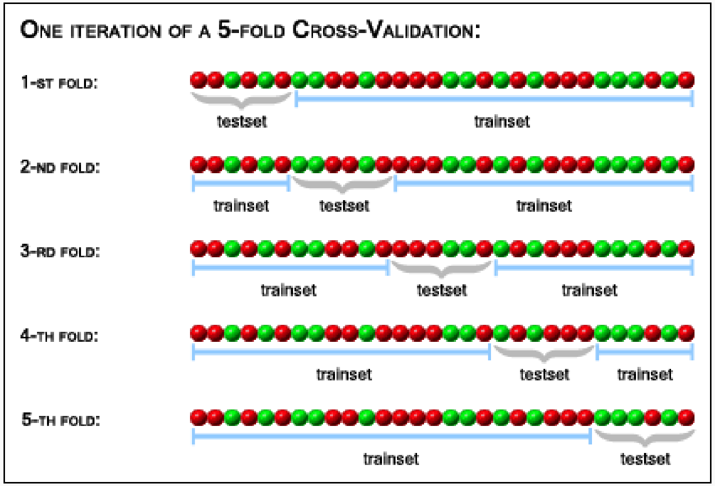
k-fold cross-validation



- The above example illustrates a *5-fold*, or $k = 5$, cross-validation.
- Each fold will act as a testing set, with the remaining $k - 1$ folds used to train the model.
- Fit model, predict values in testing set, then calculate the mean-squared prediction error (MSE)
- MSE gives an estimate of how well the model works as a predictor

Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

k-fold cross-validation



- The above example illustrates a *5-fold*, or $k = 5$, cross-validation.
- Each fold will act as a testing set, with the remaining $k - 1$ folds used to train the model.
- Fit model, predict values in testing set, then calculate the mean-squared prediction error (MSE)
- MSE gives an estimate of how well the model works as a predictor
- MSE is general-purpose and allows you to compare models of many types

Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

Cross-validating our models

Cross-validating our models

- The code for doing k-fold cross-validation, even with the `tidyverse` tools, is *just* complicated enough that it's beyond the scope of the course

Cross-validating our models

- The code for doing k-fold cross-validation, even with the `tidyverse` tools, is *just* complicated enough that it's beyond the scope of the course

- To let you practice model selection, run the following code to load in the function `rep_kfold_cv()`

```
load(url("http://spring18.cds101.com/files/R/repeated_kfold_cross_validation.RData"))
```

Cross-validating our models

- The code for doing k-fold cross-validation, even with the `tidyverse` tools, is *just* complicated enough that it's beyond the scope of the course
- To let you practice model selection, run the following code to load in the function `rep_kfold_cv()`

```
load(url("http://spring18.cds101.com/files/R/repeated_kfold_cross_validation.RData"))
```

- This function takes a linear regression model and cross-validates it automatically for you, you just supply the following inputs:

Input	Description
data	The training dataset
k	Number of folds to use
model	Model to cross-validate written in `lm()` syntax
cv_reps	Number of times to repeat cross-validation sequence to improve statistics

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.216622	56.6742	56.59307

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.216622	56.6742	56.59307

- Cross-validate the multivariate model `totalPr ~ cond + stockPhoto + duration + wheels`

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.216622	56.6742	56.59307

- Cross-validate the multivariate model `totalPr ~ cond + stockPhoto + duration + wheels`

```
rep_kfold_cv(
  data = train, k = 10,
  model = totalPr ~ cond + stockPhoto + duration + wheels, cv_reps = 3)
```

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.216622	56.6742	56.59307

- Cross-validate the multivariate model `totalPr ~ cond + stockPhoto + duration + wheels`

```
rep_kfold_cv(
  data = train, k = 10,
  model = totalPr ~ cond + stockPhoto + duration + wheels, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.6342139	23.00328	22.90515

Applying cross-validation to our models

- Cross-validate the univariate model `totalPr ~ cond`

```
rep_kfold_cv(data = train, k = 10, model = totalPr ~ cond, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.216622	56.6742	56.59307

- Cross-validate the multivariate model `totalPr ~ cond + stockPhoto + duration + wheels`

```
rep_kfold_cv(
  data = train, k = 10,
  model = totalPr ~ cond + stockPhoto + duration + wheels, cv_reps = 3)
```

r_squared	mse	adjusted_mse
0.6342139	23.00328	22.90515

- Scores indicate the multivariate model performs better than the univariate model

Credits

Mario Kart data set source: David M Diez, Christopher D Barr, and Mine Çetinkaya-Rundel. 2012. *openintro*: OpenIntro data sets and supplemental functions. http://cran.r-project.org/web/packages/openintro

Mario Kart example loosely adapted from content in chapters 6.1, 6.2, and 6.3 of the *Introductory Statistics with Randomization and Simulation* textbook by David M Diez, Christopher D Barr, and Mine Çetinkaya-Rundel and made available under the CC BY-NC-SA 3.0 Unported license.