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

# Class 28: Modeling IV

May 3, 2018

# General

# Annoucements

- Homework 5 due Friday, May 4th by 11:59pm
    - **Reminder:** 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
- Final Interviews on May 15th, 1:30pm – 4:15pm in this classroom (1004 Exploratory Hall)

# Selecting models for the *Mario Kart* eBay prices dataset

# Review: Can we predict accurately eBay prices?

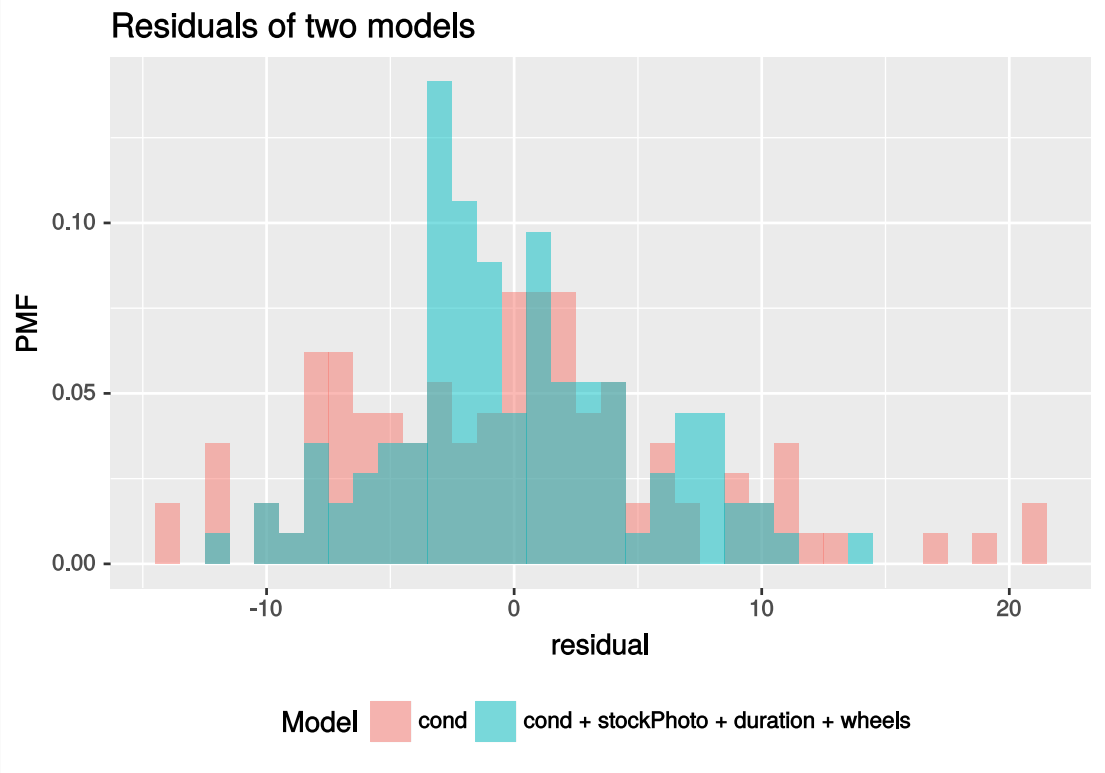
- Data scraped from eBay listings for the video game *Mario Kart Wii*
- Can we predict each game's final selling price using other information on a eBay listing page?


## Goal

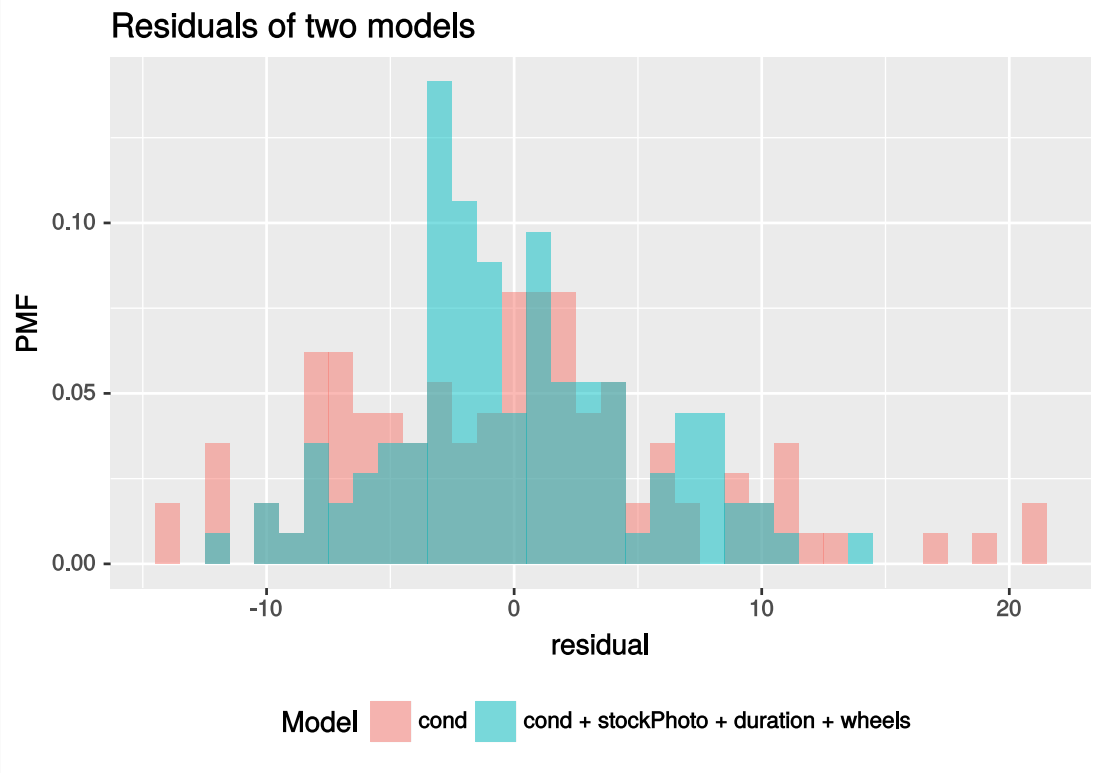
**Build a model that predicts the dataset variable `totalPr` using the other columns**

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

# How can we compare models?

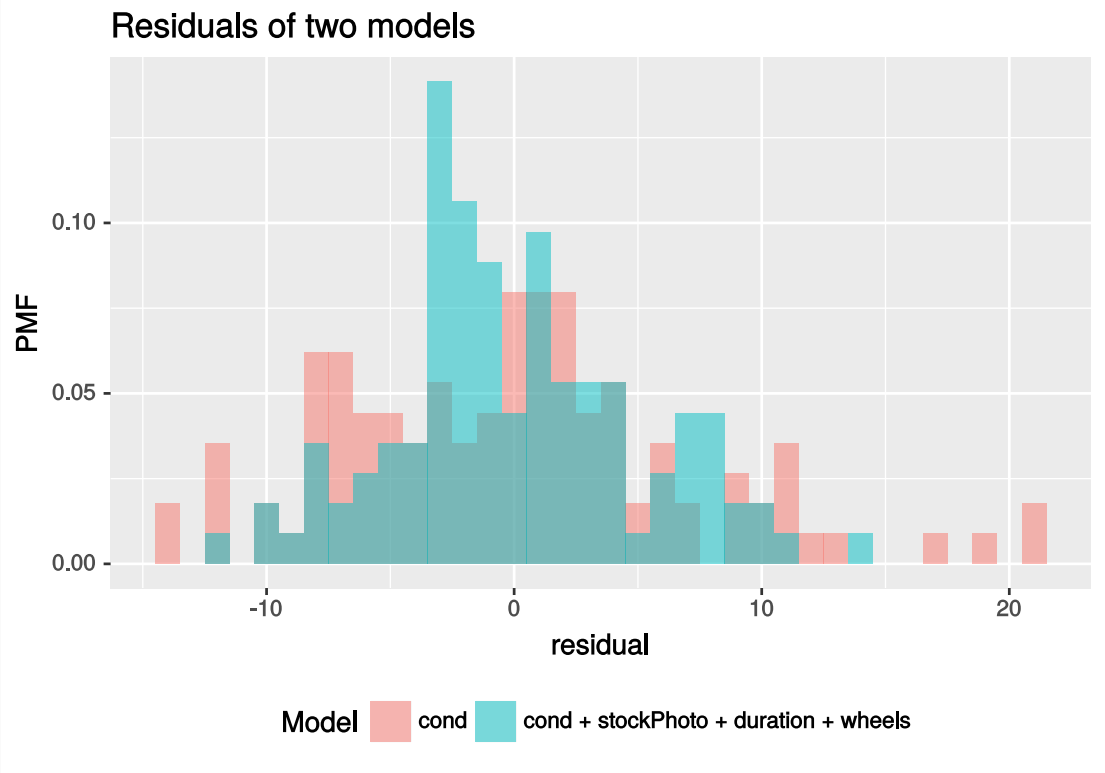


# How can we compare models?




- Model `totalPr ~ cond + stockPhoto + duration + wheels` *seems* better... but we'd like to evaluate this more precisely

# How can we compare models?




- Model `totalPr ~ cond + stockPhoto + duration + wheels` *seems* better... but we'd like to evaluate this more precisely
- **k-fold cross-validation** is a popular method for comparing the predictive power of different models

# What is k-fold cross-validation?

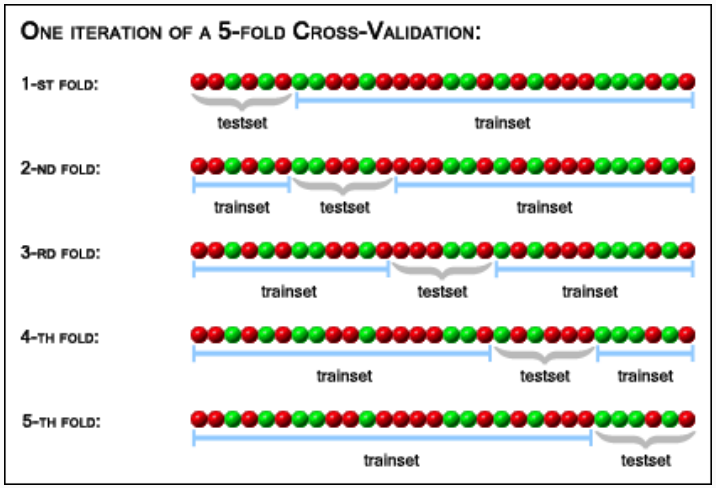
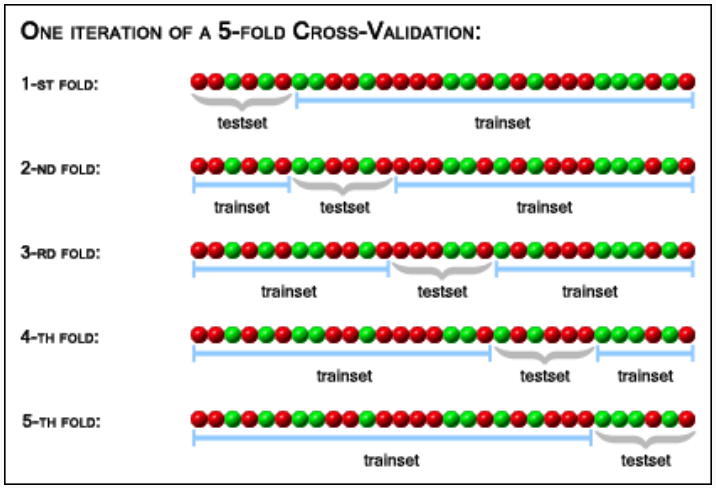



Image: "Cross-Validation Explained", *ProClassify User's Guide*, http://genome.tugraz.at/proclassify/help/pages/images/xv_folds.gif

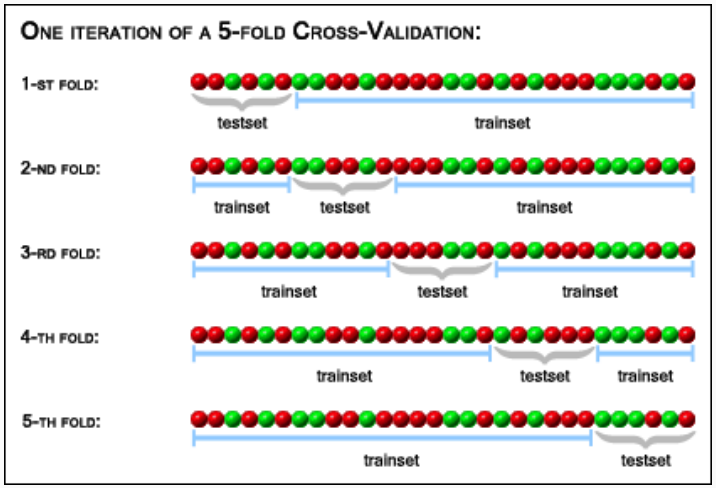
# What is k-fold cross-validation?




- 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

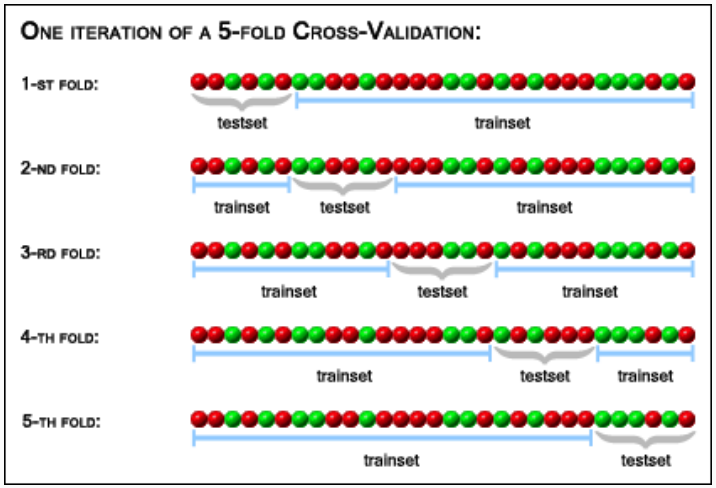
# What is k-fold cross-validation?




- 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.

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

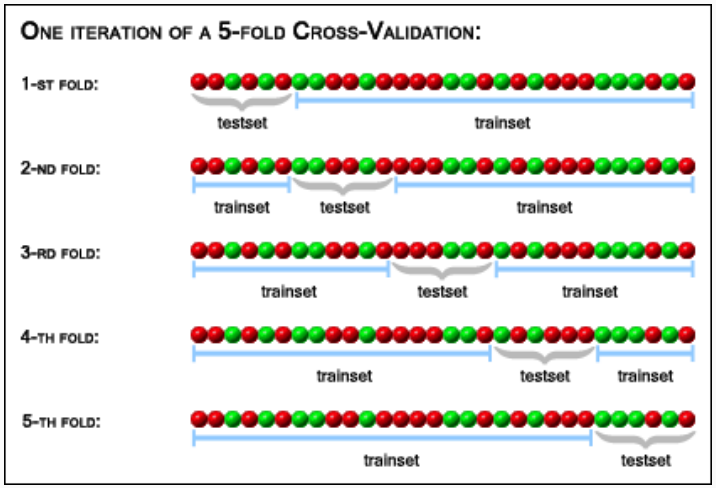
# What is k-fold cross-validation?




- 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

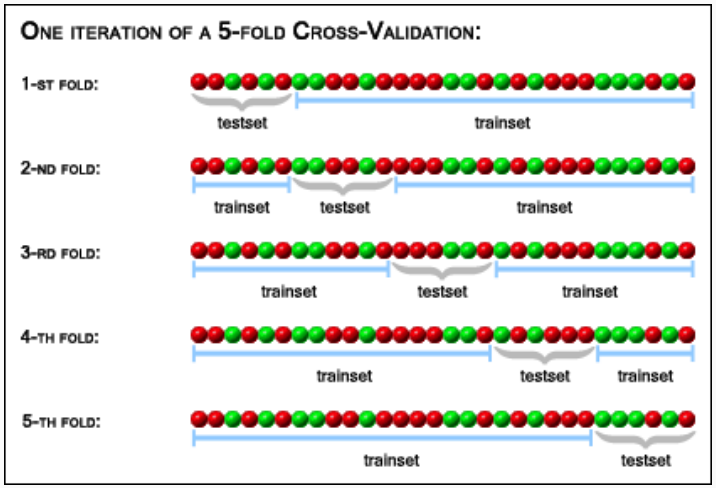
# What is k-fold cross-validation?




- 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

# What is 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

# Intermission: Fill out evaluation forms

# Cross-validating our models

# Cross-validating our models

- Load the function `rep_kfold_cv()` so you can practice model selection:

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

# Cross-validating our models

- Load the function `rep_kfold_cv()` so you can practice model selection:

```
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 |

# What's in the training dataset again?

| totalPr | cond | stockPhoto | duration | wheels |
|---|---|---|---|---|
| 41.00 | used | no | 1 | 1 |
| 51.55 | new | yes | 3 | 1 |
| 53.76 | new | yes | 1 | 2 |
| 64.95 | new | yes | 1 | 2 |
| 35.99 | used | yes | 5 | 0 |
| 56.01 | new | yes | 1 | 2 |
| 45.21 | used | yes | 3 | 1 |
| 44.00 | used | no | 7 | 1 |
| 45.00 | new | yes | 3 | 0 |
| 44.00 | new | yes | 3 | 1 |

# Applying cross-validation to our models

```
totalPr ~ cond
```

# Applying cross-validation to our models

`totalPr ~ cond`

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

# Applying cross-validation to our models

`totalPr ~ cond`

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

| r_squared | mse | adjusted_mse |
|---|---|---|
| 0.2246425 | 53.42795 | 53.30145 |

# Applying cross-validation to our models

`totalPr ~ cond`

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

| r_squared | mse | adjusted_mse |
|---|---|---|
| 0.2246425 | 53.42795 | 53.30145 |

`totalPr ~ cond + stockPhoto + duration + wheels`

# Applying cross-validation to our models

`totalPr ~ cond`

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

| r_squared | mse | adjusted_mse |
|---|---|---|
| 0.2246425 | 53.42795 | 53.30145 |

`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

`totalPr ~ cond`

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

| r_squared | mse | adjusted_mse |
|---|---|---|
| 0.2246425 | 53.42795 | 53.30145 |

`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 |
|---|---|---|
| 0.566543 | 27.45162 | 27.28987 |

# Applying cross-validation to our models

`totalPr ~ cond`

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

| r_squared | mse | adjusted_mse |
|---|---|---|
| 0.2246425 | 53.42795 | 53.30145 |

`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 |
|---|---|---|
| 0.566543 | 27.45162 | 27.28987 |

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

# Finding the best model

- You have been assigned 3 models according to your Table number in the chart on the right.
- Use `rep_kfold_cv()` to run your assigned models using the RMarkdown file in your activity repo
- Be ready to report your **Adjusted MSE** values
- Save, commit, and push when you're done

| # | cond | stockPhoto | wheels | duration | Adj.MSE |
|---|---|---|---|---|---|
| NA | X | | | | 53.3 |
| 1 | | X | | | |
| 1 | | X | X | | |
| 2 | | | X | | |
| 2 | | X | | X | |
| 3 | | | | X | |
| 3 | | | X | X | |
| 4 | X | X | | | |
| 4 | X | X | X | | |
| 5 | X | | X | | |
| 5 | X | X | | X | |
| 6 | X | | | X | |
| 6 | X | | X | X | |
| All | | X | X | X | |
| NA | X | X | X | X | 27.3 |

# Considering a minor in CDS?

GEORGE MASON UNIVERSITY

College of Science

## Minor in Computational & Data Sciences (CDS)

*"In a world driven by data, university graduates with a solid base of data knowledge have a distinct edge in the job market."*

The Department of Computational and Data Science, College of Science, offers a Computational and Data Science Minor designed to equip graduates with knowledge and skills to enhance their careers.

### Benefits of a CDS Minor

Students enrolled in the CDS minor have the benefit of following their passion with their undergraduate degree while adding the data knowledge and skills employers want and entrepreneurs need.

Enrolling gives you access to CDS Department resources including:

- career coaching & career fairs
- job opportunities & networking
- internship opportunities

# Considering a minor in CDS?

## CDS Minor Courses (15 credits total)

The CDS Minor consists of **5 courses**:

- CDS 101 - Introduction to Computational and Data Sciences **or**
  CDS 130 - Computing for Scientists
- 3 CDS courses of the following:
  - CDS 230: Modeling and Simulation I
  - CDS 251: Intro to Scientific Programming **(Spring 2016)**
  - CDS 301: Scientific Information and Data Visualization
  - CDS 302: Scientific Data and Databases **(Spring 2016)**
  - CDS 303: Scientific Data Mining
  - CDS 411: Modeling and Simulation II
- One 300+ level Science course

## Enrolling in the CDS Minor

- **Minor Declaration Form**
  **https://registrar.gmu.edu/wp-content/uploads/UMD.pdf**
  Please email it to Dr. Joseph Marr, **jmarr2@gmu.edu**
- Register for courses! (You may take a few courses before you declare).
- Have questions? Contact Dr. Joseph Marr, jmarr2@gmu.edu

# 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.