# Homework 1

Ian Effendi

## Certication

> I certify that I indeed finished reading Ch. 2 from *An Introduction to Statistical Learning*, by James Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani — Ian Effendi (`iae2784@rit.edu`)

The assignment tasks are completed below and provided in a brief report that highlights used R code and results.

```
# Loading dependencies.
devtools::load_all("./")
library(dplyr)
library(ggplot2)
```

## 1. Read in the data from the object, which was saved with `save(galaxies, file="galaxies.RData")` R command.

```
# Load the galaxies.RData file into memory.
load(file = "data/galaxies.RData")

# Confirm the class type of the loaded data.
class(galaxies)
#> [1] "data.frame"
```

## 2. Perform an EDA on the data.

### Describe shape of data

```
# Show first 6 rows.
head(galaxies)
#>     Galaxy velocity distance
#> 1  NGC0300      133     2.00
#> 2  NGC0925      664     9.16
#> 3 NGC1326A     1794    16.14
#> 4  NGC1365     1594    17.95
#> 5  NGC1425     1473    21.88
#> 6  NGC2403      278     3.22
```

`galaxies.RData` contains a 24 x 3 `data.frame` with a selection of unique galaxies from the *New General Catalogue of Nebulae and Clusters of Stars* (NGC) and the supplemental *Index Catalogues* (IC). Data has been provided by course instructor for STAT 745.

### Describe features

An example record contains the following information:

```
galaxies[1,]
#>    Galaxy velocity distance
#> 1 NGC0300      133        2
```

`Galaxy` (`fctr`) represents the unique identifier name given to galaxies from the NGC/IC catalogues. We will not be using the identifier itself in our regression problem, but they identify each of the 24 observed galaxies in the dataset.

`velocity` is an integer (`int`) representing the recessional velocity, expressed in km per second ($\frac{km}{s}$).

`distance` is a double (`dbl`) representing the proper distance of the galaxy from the observer, measured in mega-parsecs ($Mpc$ ).

Note: $1\,Mpc \approx 3.086 * 10^{19}\,km$.

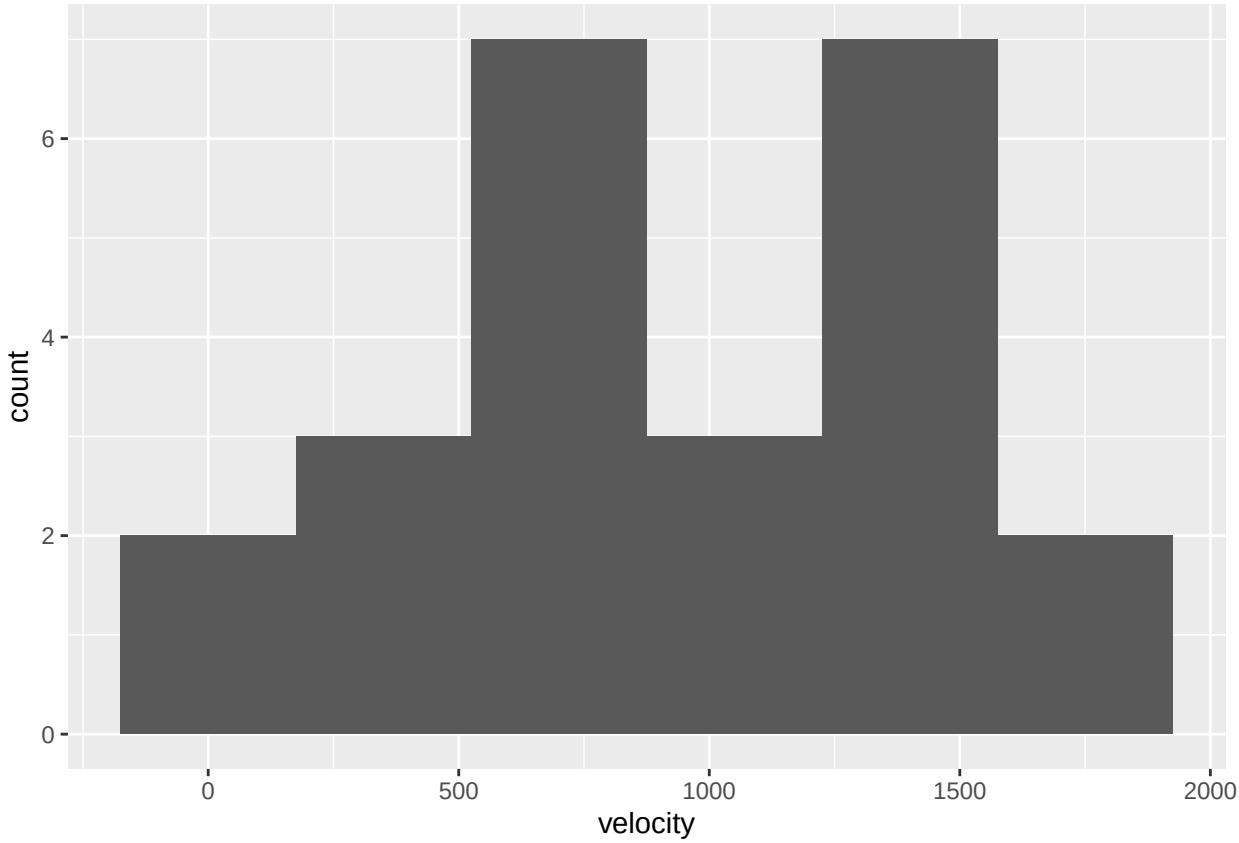
### Summary statistics

```
# Remove missing values prior to computation.
remove_na <- TRUE

# Generate summary statistics about the sample velocity.
summarise(galaxies,
          count = n(),
          std.dev = sd(velocity, na.rm = remove_na),
          avg.velocity = mean(velocity, na.rm = remove_na),
          med.velocity = median(velocity, na.rm = remove_na),
          min.velocity = min(velocity, na.rm = remove_na),
          max.velocity = max(velocity, na.rm = remove_na))
#>   count std.dev avg.velocity med.velocity min.velocity max.velocity
#> 1    24 512.814      924.375          827           80         1794
```

**`velocity`** The average recessional velocity is $\approx 924 \frac{km}{s}$, with a median velocity of $827 \frac{km}{s}$.

```
# Generate histogram.
ggplot(data = galaxies) +
  geom_histogram(mapping = aes(x = velocity), binwidth = 350)
```



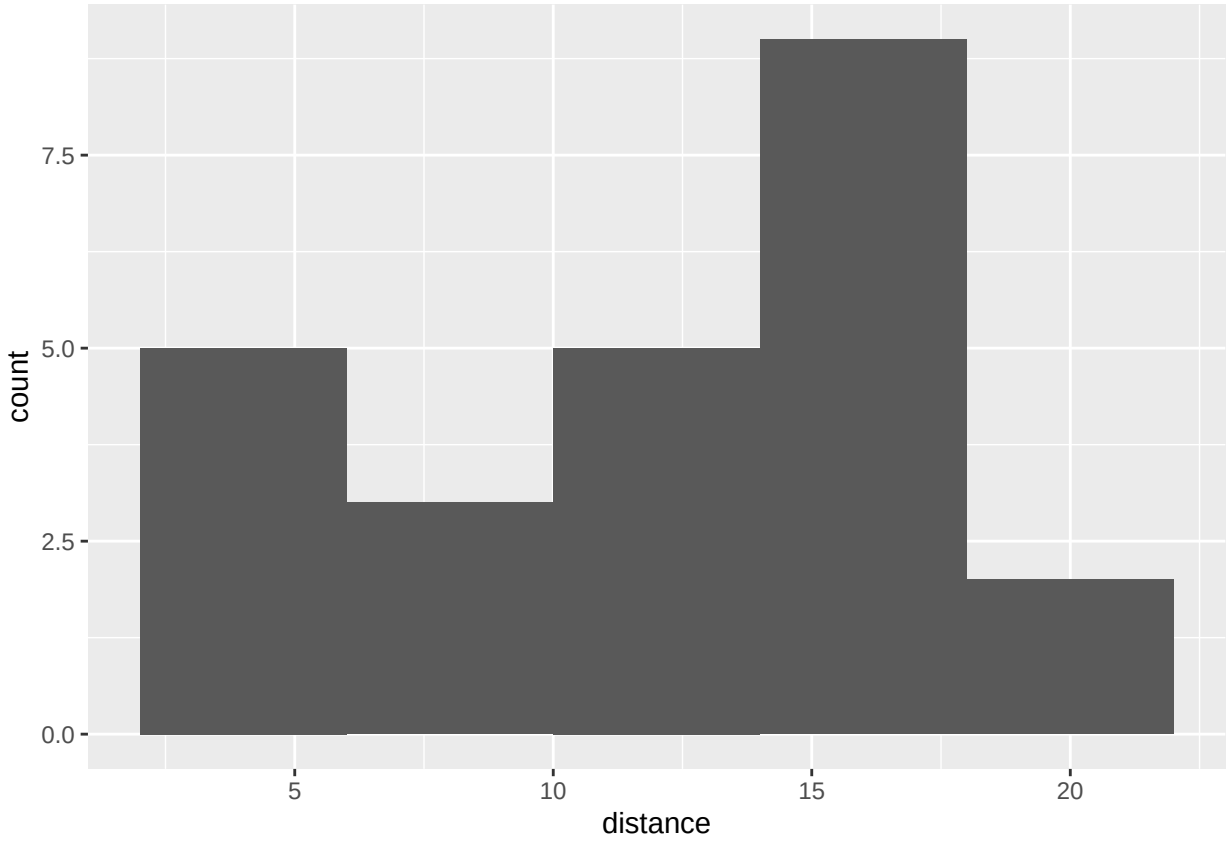

The above shows the distribution of `velocity`.

```
# Remove missing values prior to computation.
remove_na <- TRUE

# Generate summary statistics about the sample velocity.
summarise(galaxies,
          count = n(),
          std.dev = sd(distance, na.rm = remove_na),
          avg.distance = mean(distance, na.rm = remove_na),
          med.distance = median(distance, na.rm = remove_na),
          min.distance = min(distance, na.rm = remove_na),
          max.distance = max(distance, na.rm = remove_na))
#>   count  std.dev avg.distance med.distance min.distance max.distance
#> 1    24 5.814649     12.05458        13.08            2        21.98
```

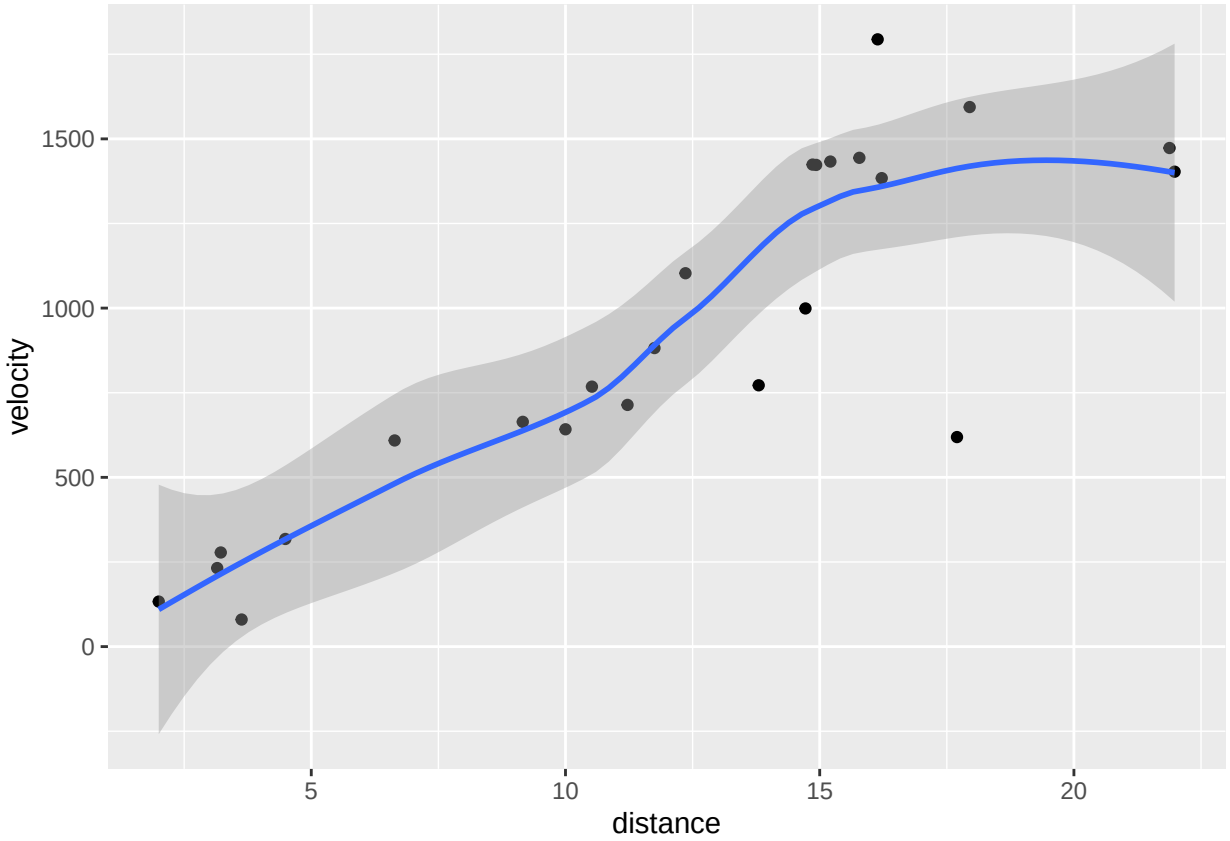
**distance** The average distance is $\approx 12.05\,Mpc$, with a median distance of $13.08\,Mpc$.

```
# Generate histogram.
ggplot(data = galaxies) +
  geom_histogram(mapping = aes(x = distance), binwidth = 4)
```




The above shows the distribution of `distance`.

```
# Generates scatterplot of the response and predictor.
ggplot(data = galaxies) +
  geom_point(mapping = aes(x = distance, y = velocity)) +
  geom_smooth(mapping = aes(x = distance, y = velocity))
#> `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




From a scatterplot of the data, we can visually identify a positive correlation between velocity and distance.

## 3. Fit a linear, no-intercept model (called Hubble's law).

The assignment considers `distance` to be the predictor and `velocity` to be the response, in the context of linear regression. Additionally, there is no intercept for this model, as shown below:

$velocity = \beta_1 * distance + \epsilon$

We can fit our model as such:

```
# Fit a no-intercept linear regression model.
model <- lm(velocity ~ 0 + distance, data = galaxies)

# Summarize information on the model's performance and coefficients.
summary(model)
#>
#> Call:
#> lm(formula = velocity ~ 0 + distance, data = galaxies)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -736.5 -132.5  -19.0  172.2  558.0
#>
#> Coefficients:
#>          Estimate Std. Error t value Pr(>|t|)
#> distance   76.581      3.965   19.32 1.03e-15 ***
#> ---
```

```
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 258.9 on 23 degrees of freedom
#> Multiple R-squared:  0.9419, Adjusted R-squared:  0.9394
#> F-statistic: 373.1 on 1 and 23 DF,  p-value: 1.032e-15
```

As we can see, there is no intercept in the model, so there is no $\beta_0$ coefficient.

The coefficient of regression for the `distance` factor is $\beta_1 = 76.681$.

## 4. Assess the quality of the model fit, but do not explore other models.

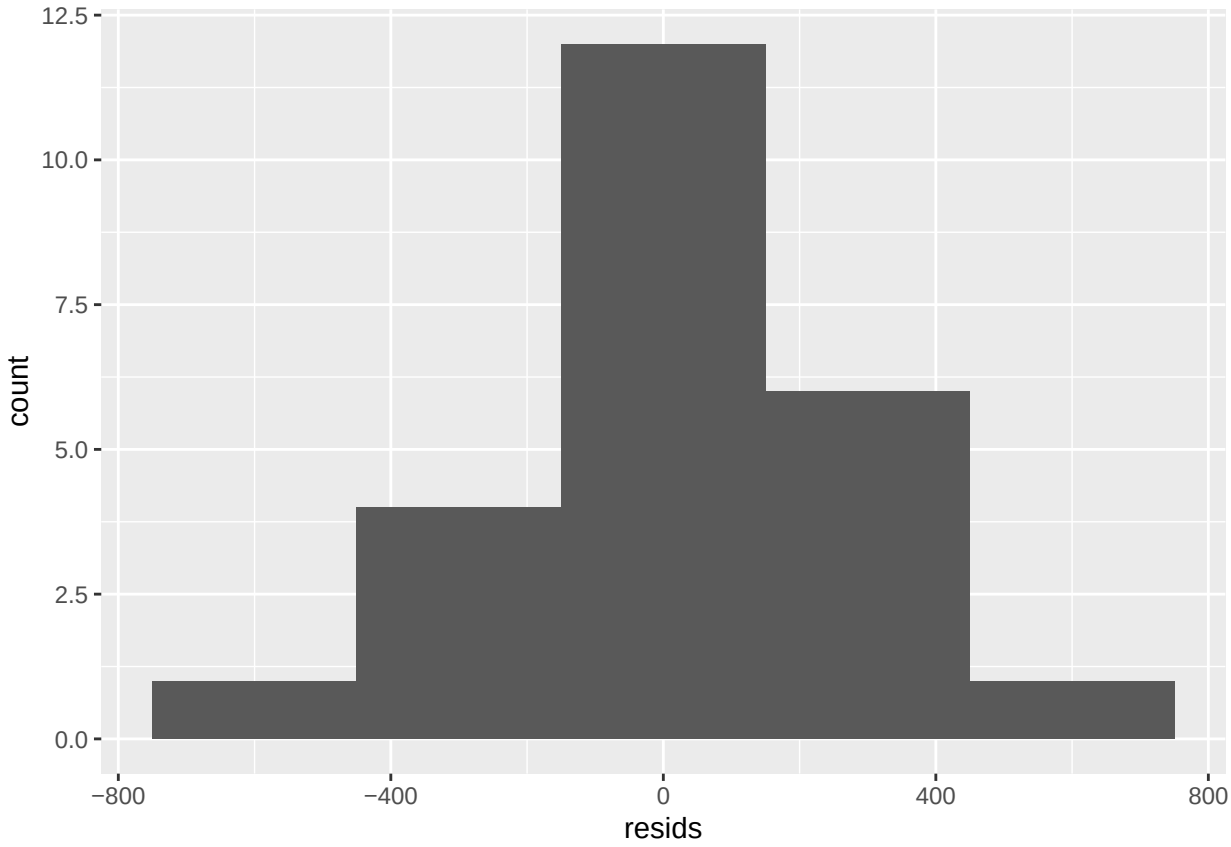
**Residuals** ($e_i$)

$$e_i = y_i - \hat{y}_i$$

A residual is the difference between our predicted value and the actual value for the $i$th observation. Intuitively, it can be considered as the vertical distance between a point and the line of the linear model.

```
# Add residuals to the model.
resids = resid(model)
galaxies.model = cbind(galaxies, resids)

# Plot residuals against velocity.
ggplot(data = galaxies.model) +
  geom_histogram(mapping = aes(x = resids), binwidth = 300)
```




The residuals appear to be normally distributed.

**Measure of Determination:** $R^2$

$$R^2 = \frac{\sum_{i=1}^{n}(y-\bar{y})^2 - \sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n}(y-\bar{y})^2}$$

$R^2$ is the coefficient of determination and it is a measure that describes the proportion of total variability explained by the regression model. $R^2$ can take on a value between $[0, 1]$, with values closer to 1 representing a better fit for the data.

In this case, $R^2 \approx 0.94$, which indicates a very strong fit for the data. If we had more observations to setup a test/validation structure, we could weigh concerns of over-fitting as well.

**Residual Standard Error** ($RSE$)

$$RSE = \frac{\sum_{i=1}^{n}(e_i^2)}{n-(1+k)}$$

The residual standard error is the standard deviation of the residuals. Typically, a smaller residual standard error means our predictions are better.

For regression of `velocity` on `distance` gives an $RSE = 258.9$. The advantage of $RSE$ is it is in the units of our response: our predictions deviate by $\approx 259\frac{km}{s}$ from the actual values.

By collecting more data and refitting the model, we may be able to improve its performance for the purposes of predicting velocity in the future.

## 5. Estimate $\beta_1$ (called Hubble's constant), including units. Hubble's constant is given in $km * sec^{-1} * Mpc^{-1}$. A `mega-parsec` ($Mpc$) is $3.086 * 10^{19} km$. Velocity data is given in $\frac{km}{s}$ and distance in $Mpc$.

Hubble's law is formally defined as such:

$$v = H_0 D$$

$v$, represents the recessional velocity and $D$ represents the proper distance. $H_0$, which represents Hubble's constant of proportionality, corresponds to the $\beta_1$ coefficient of regression in our no-intercept linear model, which is shown below:

```
model$coefficients
#> distance
#> 76.58117
```

Therefore, $H_0 \approx 76.85\frac{(km/s)}{Mpc}$.

This is equivalent to $\approx 2.50 * 10^{-18} km/s$.

## 6. Find $\beta_1^{-1}$ (which approximates the age of the universe) in seconds, and then transform it into `years`.

Hubble time, representing the approximate age of the universe can be calculated as the reciprocal of Hubble's constant.

$$t_H = d/v = 1/H_0$$

Given our linear model provides us directly with $\beta_1 = H_0$, we can calculate $t_H$ as the reciprocal of our coefficient of regression for `distance`.

$$t_H = \beta_1^{-1}$$

$\beta_1 = 76.5811720291694 \frac{(km/s)}{Mpc}$

$t_H = \beta_1^{-1} = 0.0130580399007096 \frac{Mpc}{(km/s)}$

This suggests that Hubble time is $t_H \approx 1.277x10^{10} years$.