Lab 2

PSTAT 115, Spring 2024

Invalid Date

Goal:

- Brief review: sufficient statistics.
- Bayestian and Frequentist differences.
- An example of Bayesian analysis.
- An example of data generating process (DGP).

Sufficient Statistics:

- Let $L(\theta) = p(y_1, y_2, ..., y_n|\theta)$ be the likelihood and $s(y_1, ..., y_n)$ be a statistic
- **factorization theorem**: s is sufficient if $L(\theta) = h(y_1, ..., y_n)g(s, \theta)$, where h is not a function of θ
- $L(\theta) \propto g(s, \theta)$
- Many possible sufficient statistics, for example, the original sample is always a sufficient statistic.
- Often seek a minimal sufficient statistic.

Examples

a) Let $Y_1, Y_2, \ldots, Y_n$ be a random sample in which Y_i possesses the probability density function $f(y_i|\theta) = \frac{1}{\theta} e^{\frac{-y_i}{\theta}}$ where $0 \leq y_i < \infty$ and $\theta > 0, i = 1, 2, \ldots, n$. Find the likelihood, log-likelihood, proportional simplification. What is the MLE? Also find a sufficient statistic.

b) Let $X_1, X_2, \ldots, X_n$ $i.i.d \sim N(\mu, \sigma^2)$ with unknown mean and known variance.Find the likelihood, log-likelihood, proportional simplification. What is the MLE? Also find a sufficient statistic.

Comparison between Bayesian and Frequentist Statistics:

- In frequentist inference, unknown parameters treated as constants
  - Estimators are random (due to sampling variability)
- In Bayesian inference, unknown parameters are random variables.
  - Need to specify a prior distribution for θ - The prior can be uninformative or informative.
  - Gather data
  - Update your prior distribution with the data using Bayes' theorem, resulting in posterior distribution. The posterior distribution is a probability distribution that represents your updated beliefs about the parameter after having seen the data.
  - Analyze the posterior distribution and summarize it (mean, median, sd, quantiles…)
- Confidence level differences
  - Bayesian confidence intervals describe information about the location of the true value of theta after observing Y. (Post experimental coverage)
  - Frequentist confidence intervals describe the probability that the interval will cover the true value before the data is observed. (Pre experimental coverage)

Bayes Rule for Bayesian Statistics

Bayes Rule

$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$

- $P(A \mid B)$ is the conditional probability of A given B.

- $P(B \mid A)$ is the conditional probability of B given A.
- $P(A)$ and $P(B)$ are called the marginal (unconditional) probabilities of A and B.

Bayes Rule for Bayesian Statistics

$$P(\theta \mid y) = \frac{P(y \mid \theta)\, P(\theta)}{P(y)} = \frac{P(y \mid \theta)\, P(\theta)}{\displaystyle\int_{\Theta} p(y \mid \tilde{\theta})\, p(\tilde{\theta})\, d\tilde{\theta}} \propto P(y \mid \theta)\, P(\theta)$$

- $P(\theta \mid y)$ is the posterior distribution.
- $P(y \mid \theta)$ is the likelihood (sampling model).
- $P(\theta)$ is the prior distribution.
- $P(y) = \displaystyle\int_{\Theta} p(y \mid \tilde{\theta})\, p(\tilde{\theta})\, d\tilde{\theta}$ is the model evidence.

i.e. Posterior is proportional to likelihood times prior

Bayesian Example

An example from BDA3 (the Gelman book in the syllabus) is a spelling correction problem. The problem setting is, assume that a user searched for "radom". The question is, what is the probability that they typed something else?

First assume that there are only three words to consider, "radom", "random", and "radon". Assume also that you've been given the likelihoods for typing "radom" given you actually wanted to type each of those three words and also the prior probability that each of those words was what was typed.

Likelihoods:

$$p(\text{typed radom}|\text{wanted to type random}) = 0.00193$$
$$p(\text{typed radom}|\text{wanted to type radon}) = 0.000143$$
$$p(\text{typed radom}|\text{wanted to type radom}) = 0.975$$

Priors:

$$p(\text{wanted to type random}) = 7.60 * 10^{-5}$$
$$p(\text{wanted to type radon}) = 6.05 * 10^{-6}$$

$$p(\text{wanted to type radom}) = 3.12 * 10^{-7}$$

Now we can use Bayes' rule to compute the posterior.

```
likelihood = c(0.00193, 0.000143, 0.975)
prior = c(7.60e-5, 6.05e-6, 3.12e-7)

unnormalized_posterior = likelihood * prior
posterior = unnormalized_posterior / sum(unnormalized_posterior)
posterior
```

```
[1] 0.324696347 0.001915128 0.673388524
```

Posterior:

$$p(\text{wanted to type random}|\text{typed radom}) = 0.325$$
$$p(\text{wanted to type radon}|\text{typed radom}) = 0.002$$
$$p(\text{wanted to type radom}|\text{typed radom}) = 0.673$$

Questions

- Was this what you expected?
- How do you think the prior was estimated?
- How do you think the likelihood was estimated?

Data Generating Process (DGP)

Dataset and Background

If you ever want to just experiment with numbers, there are a bunch of handy datasets built into common R packages. To see them type:

```
data()
```

Let's use one of these to talk about generating models + covariates.

ChickWeight is a dataset of weights collected over time for a number of different chicks on different diets. Let's load up the ChickWeight dataset into a tibble to make it easy to work with:

```
df = ChickWeight %>% as.tibble
```

```
Warning: `as.tibble()` was deprecated in tibble 2.0.0.
i Please use `as_tibble()` instead.
i The signature and semantics have changed, see `?as_tibble`.
```

The %>% notation is a pipe in the tidyverse package. It is shorthand for:

```
df = as.tibble(ChickWeight)
```

Using tibbles (instead of base R dataframes) and pipes (instead of function calls) can make it easier to do basic data transformations. You can do all of this in base R as well.

First let's print the ChickWeight tibble and see what's in it:

```
df
```

```
# A tibble: 578 x 4
   weight  Time Chick Diet
    <dbl> <dbl> <ord> <fct>
 1     42     0 1     1
 2     51     2 1     1
 3     59     4 1     1
 4     64     6 1     1
 5     76     8 1     1
 6     93    10 1     1
 7    106    12 1     1
 8    125    14 1     1
 9    149    16 1     1
10    171    18 1     1
# i 568 more rows
```

Usually we'd look at the data before writing out a generating model. Just as an exercise though, let's do the reverse. We'll write out a generating model and then make some plots to see how reasonable our assumptions were.

- What is it that we're trying to model?
  - Weight
- What other data did we collect to explain what we're modeling (covariates)?
  - Time
  - Chicken ID
  - Diet

The simplest thing we might assume is that errors in our measurements are normally distributed. This means:

$$\text{weight} \sim N(\mu, \sigma)$$

In this case, weight is fixed data, and we'll need to estimate μ and σ (they are the estimands).

- What might be a limitation of the normal assumption here?
  - Our chickens grow over time. Lumping all the weights together at these different times won't produce a normal distribution.
  - There are multiple diets. Should every diet have the same mean?
  - Our outcomes appear to be rounded (looks like we'd never measure a weight of 75.33333 for instance)
  - Weights will be non-negative

How might we incorporate the chickens growing over time? We might assume that chickens grow linearly with time:

$$\text{weight}_i \sim N(\alpha t_i + \beta, \sigma)$$

In this case the new estimands are α, β, and σ. weight_i and t_i are both data.

How might we expect the different diets to affect things? Maybe it doesn't affect the time zero weight of the chicken (β), but it does affect the slope (α)? We can code in a diet dependence easily enough:

$$\text{weight}_{i,d} \sim N(\alpha_d t_i + \beta, \sigma)$$

The generating model also requires us to specify how the chickens are given a diet. We might assume this is uniformly sampled from the available diets, or something else. This is more important if we actually need to estimate the diet the chicken ate. In this case, the diet is given.

Pseudocode

Assume that the diet is categorically distributed with 4 categories and corresponding probabilities $p = (p_1, p_2, p_3, p_4) = (0.4, 0.2, 0.2, 0.2)$. The covariate *time* and parameters $\alpha_1, \cdots, \alpha_k$, β are given.

```
for(i in 1:nrow(ChickWeight)) {
  Draw diet_i from Categorical(4, p)
  Set alpha_i = alpha_k if diet_i = k, k = 1, 2, 3, 4
  Draw weight_i from Normal(alpha_i * t_i + beta, sigma)
}
```

We can use this to generate data as:

```
p=c(0.4, 0.2, 0.2, 0.2)
alpha = c(0.8, 1.0, 1.2, 1.4)
beta   <- 40
sigma  <- 5

diet = c()
weight = c()

for(i in 1:nrow(ChickWeight)) {

  diet[i] = sample(1:4, size = 1, replace = TRUE, prob = p)
  alpha_i = alpha[diet[i]]

  t_i = ChickWeight$Time[i]

  weight[i] = rnorm(1, mean = alpha_i*t_i + beta, sd = sigma)
}
```

```
tibble(
  time   = ChickWeight$Time,
  diet   = factor(diet, levels = 1:4, labels = 1:4),
  weight = weight
)
```

```
# A tibble: 578 x 3
    time diet  weight
   <dbl> <fct>  <dbl>
 1     0 4       39.9
```

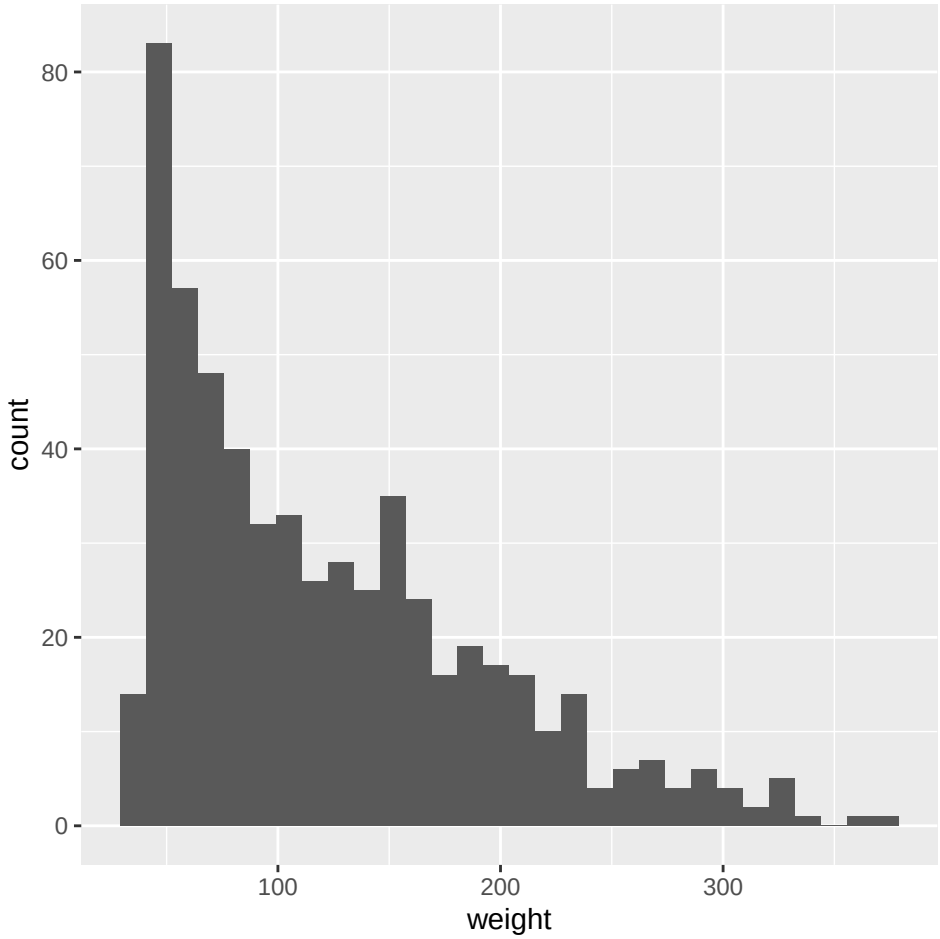
```
 2     2 4       51.8
 3     4 1       39.3
 4     6 1       37.5
 5     8 4       44.3
 6    10 4       53.7
 7    12 1       49.1
 8    14 2       53.1
 9    16 1       49.7
10    18 2       56.5
# i 568 more rows
```

Exploring our data

Since it is the weights we are concerned with, let's look at that first.

```
df %>%
  ggplot(aes(weight)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

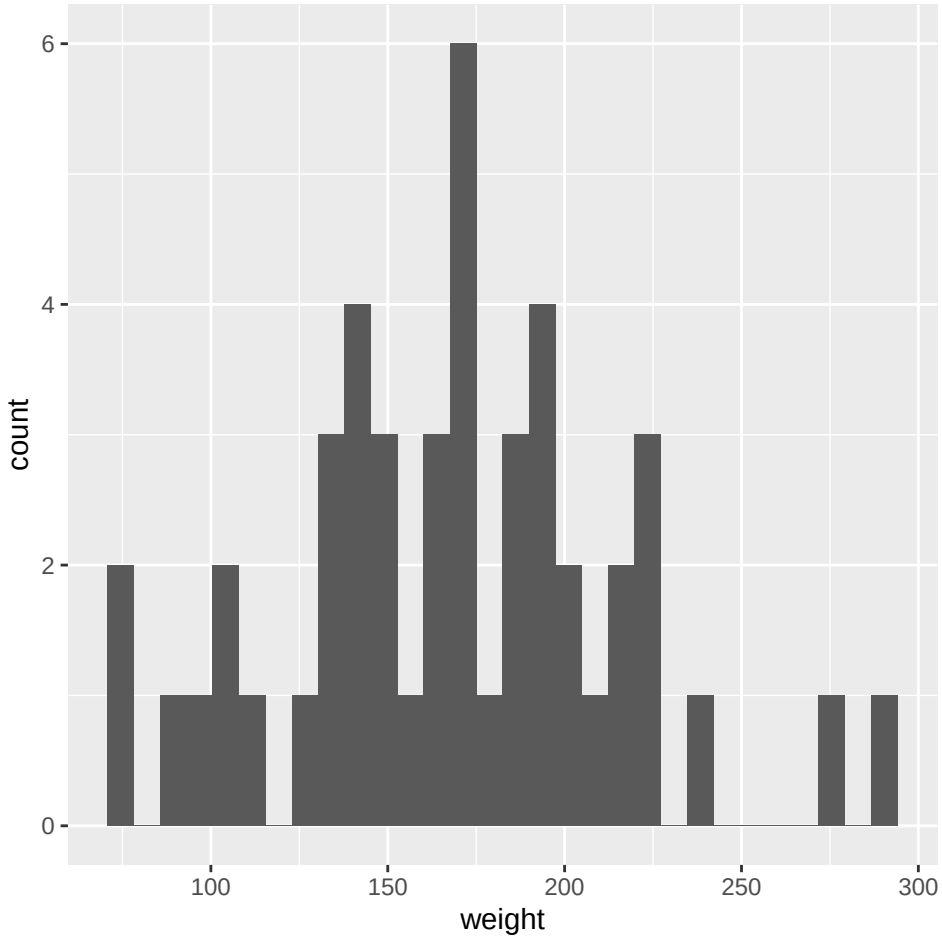


Perhaps unsurprisingly, this plot is mostly unintelligible. There's all sorts of things mixed together, so let's try to use some more plots to break it apart.

Let's have a look at $p(\text{weight}|\text{Time} = t)$.

```
df %>%
  filter(Time == 16) %>%
  ggplot(aes(weight)) +
  geom_histogram()
```

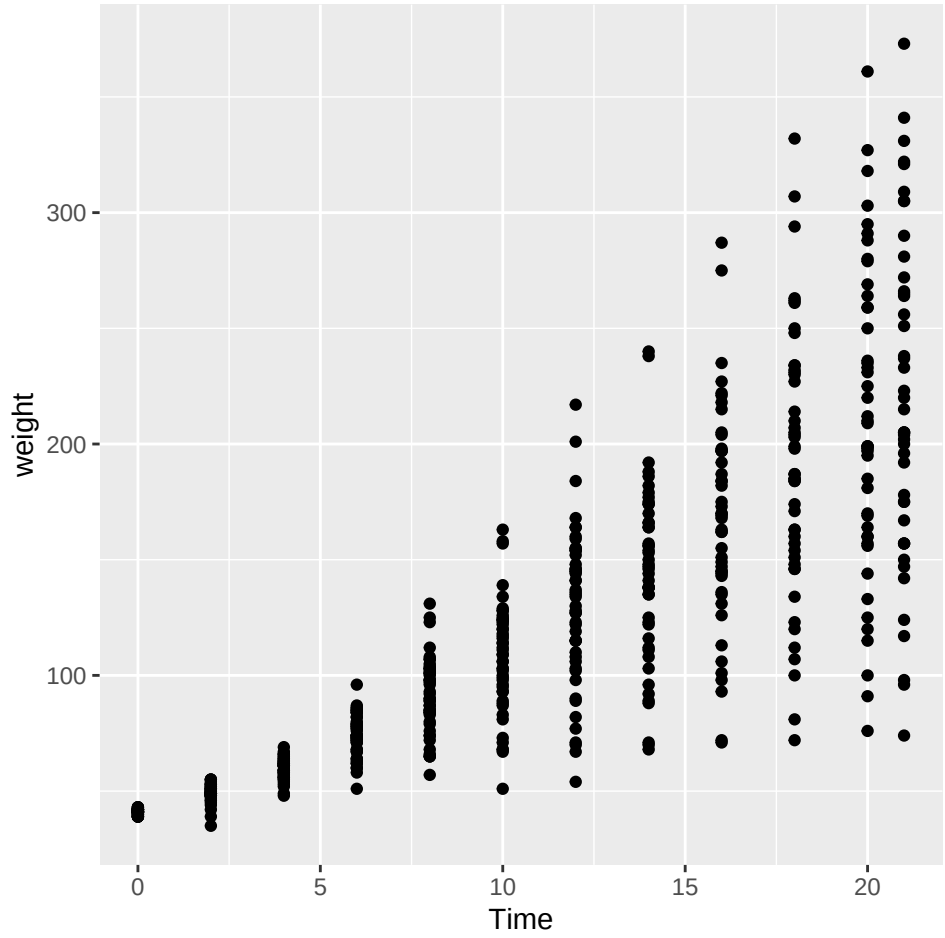
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



If we assume this conditional distribution is normal, then we're saying the output (weight), is characterized by two parameters μ and σ (that can be functions of the covariates). μ and σ are our estimands here – the things we'd try to estimate.

Let's try to get an idea of what $\mu(t)$ would look like by plotting weight vs. time:

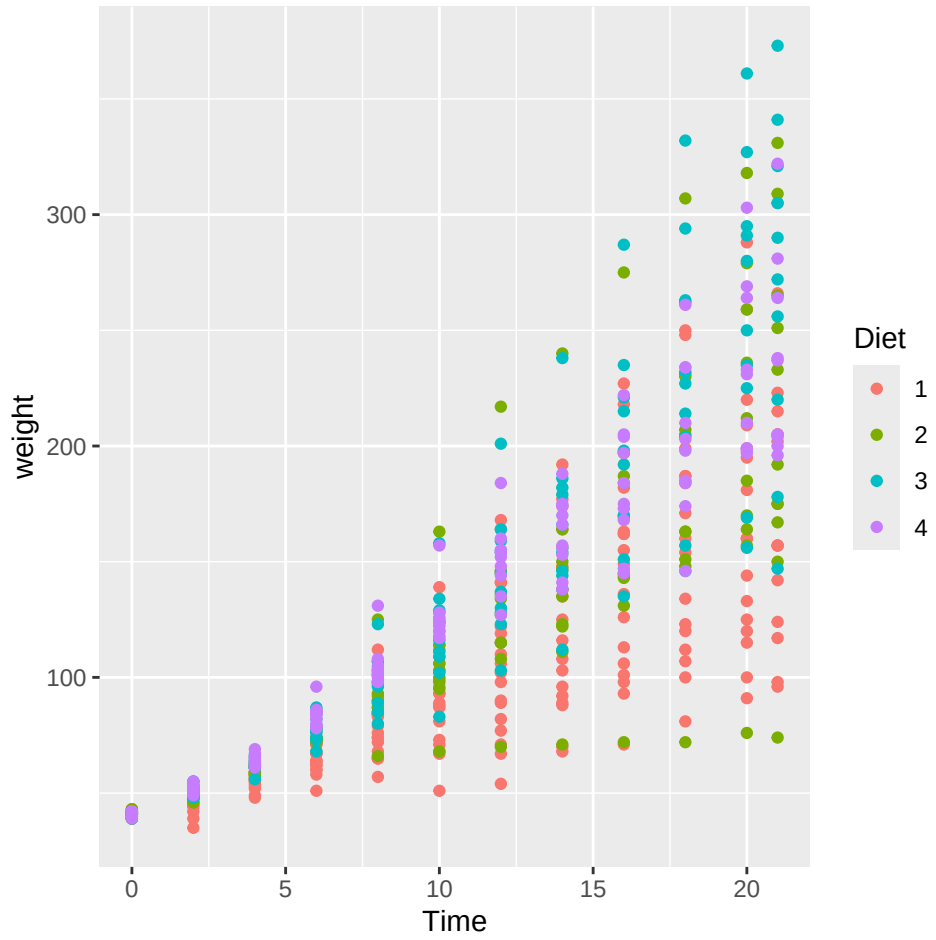
```
df %>%
  ggplot(aes(Time, weight)) +
  geom_point()
```



- How does this differ from what we proposed initially?
  - The noise is heteroskedastic (this means the amount of noise changes with another covariate – in this case time)

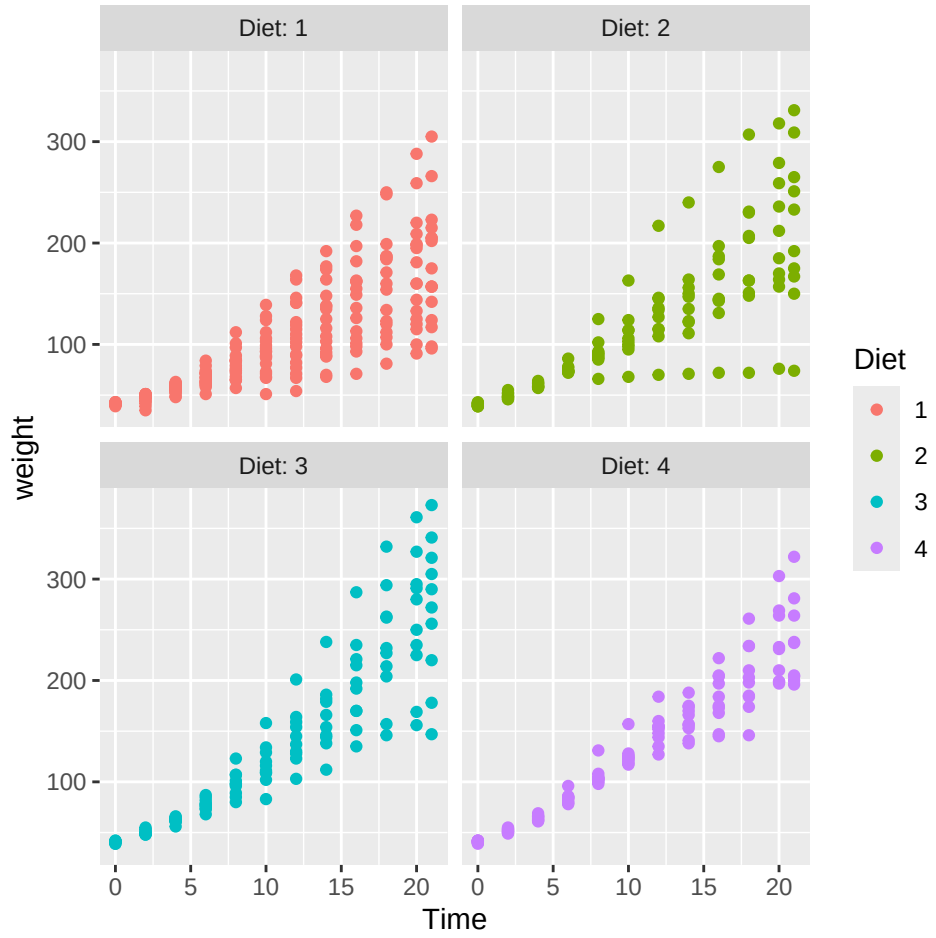
Let's try to get a handle on how diet affects any of this:

```
df %>%
  ggplot(aes(Time, weight)) +
  geom_point(aes(color = Diet))
```



That's kindof hard to see. Maybe we can break this out into different plots:

```
df %>%
  ggplot(aes(Time, weight)) +
  geom_point(aes(color = Diet)) +
  facet_wrap(~ Diet, nrow = 2, labeller = label_both)
```



- Does diet seem to have a strong affect on the growth of the chickens?
  - The effect isn't clear to me. Maybe someone else is more creative. At best maybe diet 4 produces more consistently sized chickens?
- What might be the next step in this analysis?
  - I'd feed food to more chickens. It's probably not that expensive to collect more data here.