BT1101

Lab 7: Logistic Regression and Time Series Forecasting

Installing and loading packages

```
# load required packages
# install any package below if it's first time loaded in your computer.
library(dplyr)
library(tidyr)
library(tseries)
library(TTR) # One alternative for time-series in R
library(forecast) # An alternative for time series in R
library(car) # "Companion to Applied Regression" package, for F-test for linear combination of regression coefs
library(wooldridge) # wooldridge data set will be used in this tutorial
library(ggplot2) # optional. we expect you to know base graphics, but allow ggplot if you find it easier
```

We expect you to know base graphics but allow ggplot if you find it easier

Part 1

Part One: Lab Session Completion and Discussion

Question 1

- Dataset required: `SGHDBp.csv`

Note: This dataset comes from publically available data from the Singapore Department of Statistics, or SingStat. https://data.gov.sg/dataset/hdb-resale-price-index.

First, load in the dataset for this question. There is only one variable, which is the average HDB resale price index. Q1 of 2009 is set as the "base" period, and thus has by definition an index value of 100. The index values of the rest of the years are relative to this base value (so a value of 120 means that the average HDB resale price index for that quarter is 120% (or 1.2x) that of the index of Q1/2009).

The code below will also "hold out" Years 2018 and 2019, to test the predictions of our model. This means that we fit the model using all the years except 2018 and 2019, and then once we have the fitted model, we see how well it does on 2018 and 2019.

Data Munging

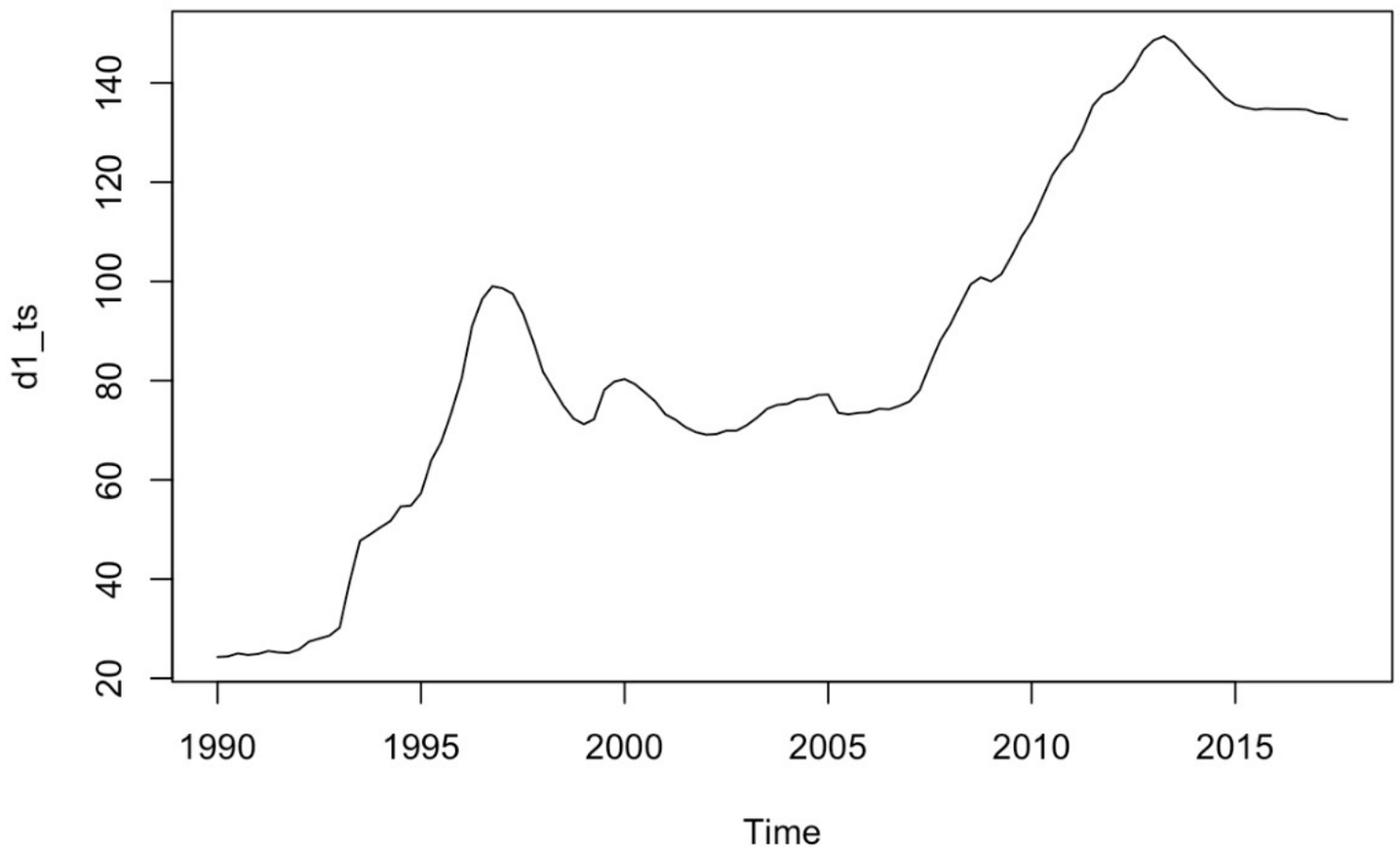
```
d1_wide = read.csv('SGHDBp.csv', header=T, na.strings = "NA")
# removing unused columns
d1_wide_HELDOUT <- d1_wide[,114:119] # HOLDING OUT values in 2018 and 2019
d1_wide <- d1_wide[,2:113] # keeping values up to and including 2016

# convert to a `ts` object:
d1_ts = ts(unlist(d1_wide[1,1:ncol(d1_wide)], use.names=F), frequency=4, start = c(1990, 1))

# also create a long form data frame. If you are interested in learning more about dplyr, try understanding what each step i
n this code does by running each line separately (without the last %>%), and inspecting the resulting file using head(d1_lon
g)
d1_long <- d1_wide %>%
  # gather() converts wide-form to long-form.
  gather(key="YearQuarter", value="PriceIndex") %>%
  # remove the annoying "X"
  mutate_at("YearQuarter", function(x) {sub(pattern="X", replacement="", x)}) %>%
  # split "YearQuarter" into a "Year" variable and a "Quarter" variable
  # and make a variable called "TimeIndex" that just goes 1, 2, 3, 4...
  mutate( Year = as.numeric(substr(YearQuarter, start=1, stop=4)),
          Quarter = substr(YearQuarter, start=6, stop=7),
          TimeIndex = 1:length(YearQuarter)) %>%
  # Rearrange the columns in a nicer order
  select("TimeIndex", "YearQuarter", "Year", "Quarter", "PriceIndex")
```

Time series object

	TimeIndex <int>	YearQuarter <chr>	Year <dbl>	Quarter <chr>	PriceIndex <dbl>
1	1	1990Q1	1990	1	24.3
2	2	1990Q2	1990	2	24.4
3	3	1990Q3	1990	3	25.0
4	4	1990Q4	1990	4	24.7
5	5	1991Q1	1991	1	24.9
6	6	1991Q2	1991	2	25.5

(1a) First, plot the data. There is only one variable, so just plot this against time on the horizontal axis. What do you notice? (Stationary? Trend? Seasons? Cycles?)

```
plot(d1_ts)
```

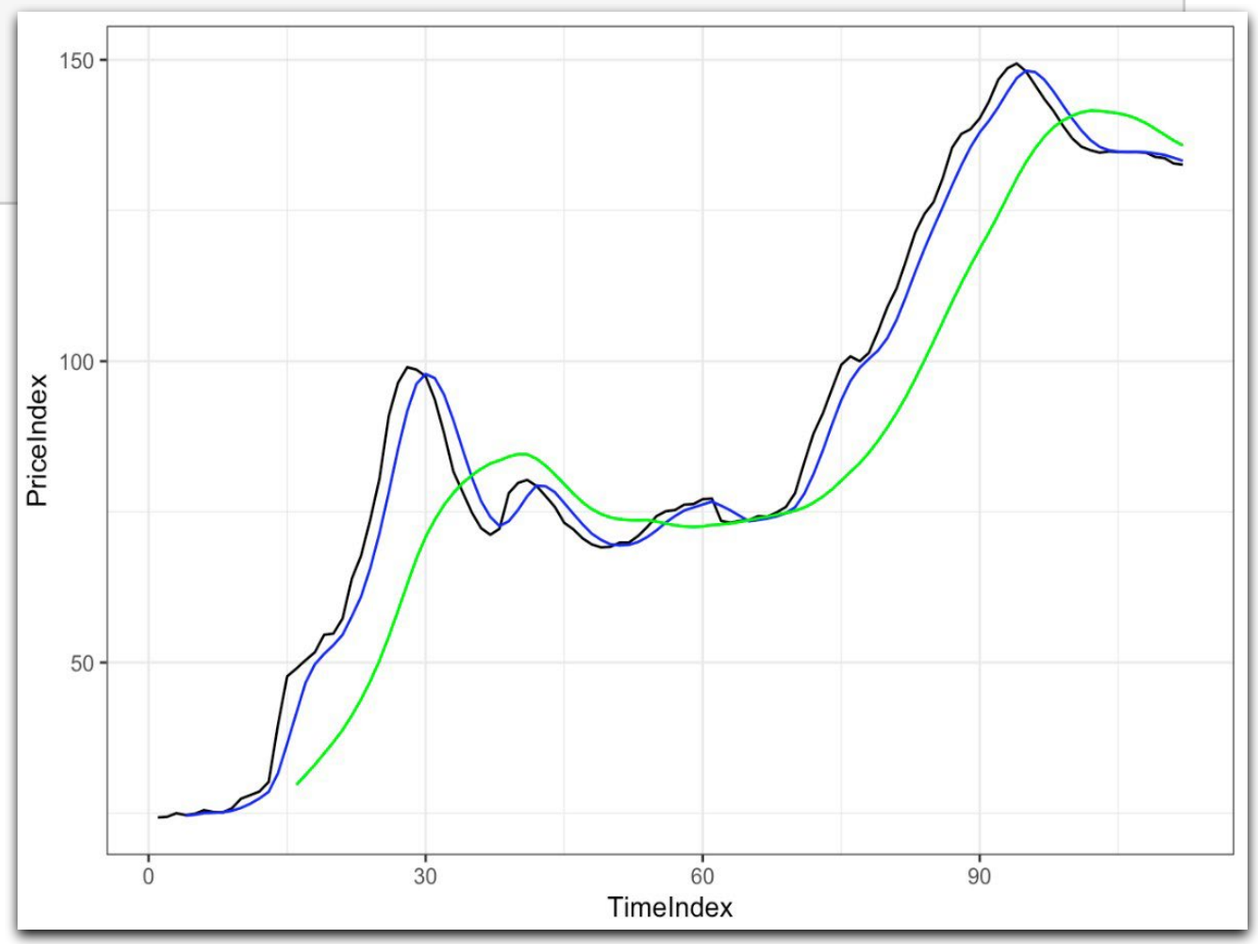


What do you notice? (Stationary? Trend? Seasons? Cycles?)

There seems to be an increasing trend. There does not seem to be any apparent seasonal effect.

(b) Calculate a Simple Moving Average model to the data, using the Equation we had in class, where $m_t = (y_t + y_{t-1} + \ldots + y_{t-K+1})/K$. Calculate one with window size of 4 periods (1 year). Calculate a second one of 16 periods (4 years). Plot these two (and the actual data) on the same plot. Discuss what you see.

```
d1_long$ma4 = TTR::SMA(d1_long$PriceIndex, n=4)
d1_long$ma16 = TTR::SMA(d1_long$PriceIndex, n=16)

ggplot(d1_long, aes(x=TimeIndex)) +
  geom_line(aes(y=PriceIndex)) +
  geom_line(aes(y=ma4), col="blue") +
  # geom_line(aes(y=SMApred16), col="purple") +
  geom_line(aes(y=ma16), col="green") +
  theme_bw()
```



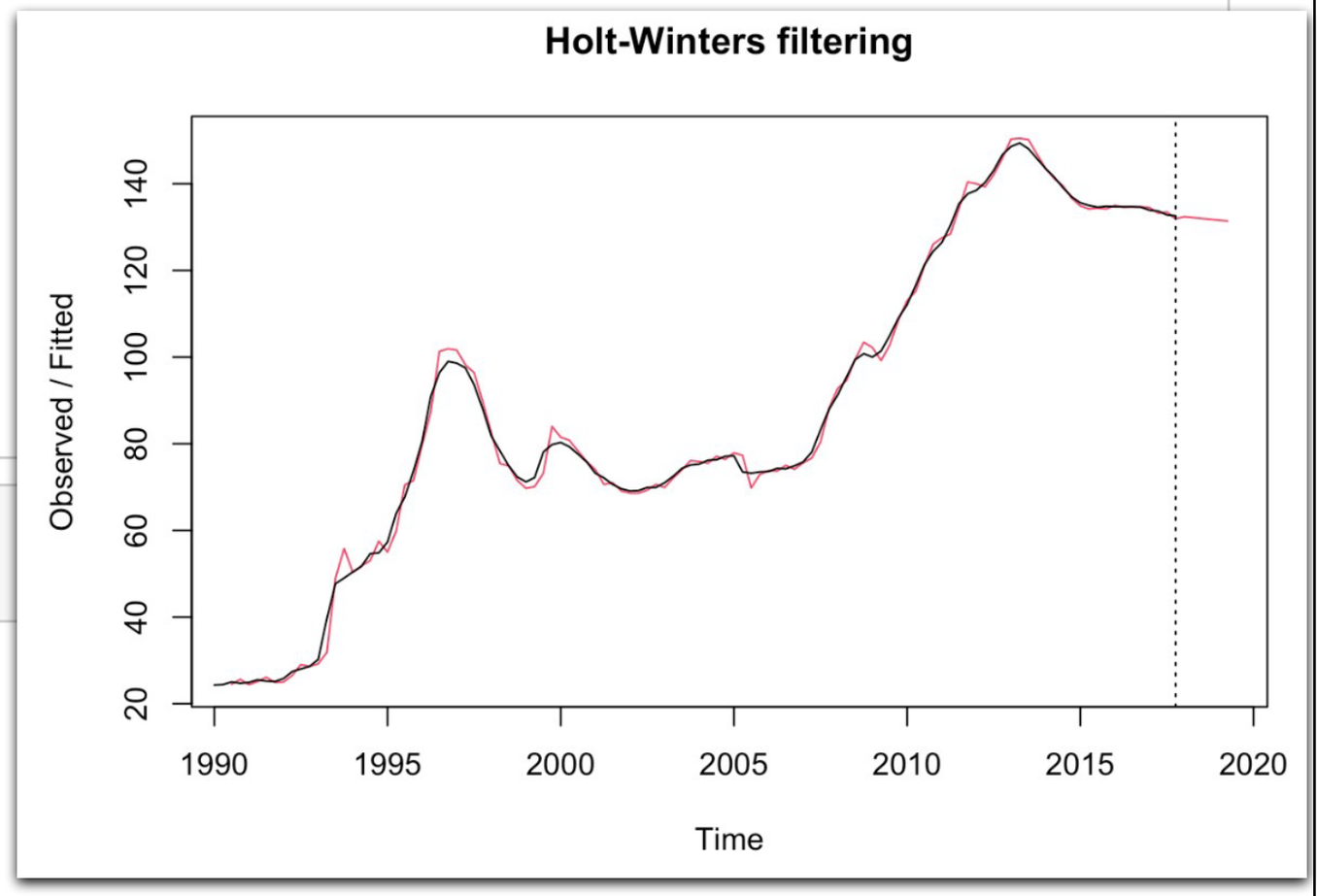
The lines do seem "offset" to the right. This is because moving average models can "follow" recent changes, but they're always a little slow, and playing catch-up. The peaks happen earlier in the real data

(c) Based on what you observed about the time-series in Q1a, fit a HoltWinters model to the data. Use the model to predict the next 6 periods (6 quarters), and plot the predictions.

```
hw1 = HoltWinters(d1_ts, gamma=FALSE)
hw1
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = d1_ts, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 1
##  gamma: FALSE
##
## Coefficients:
##    [,1]
## a 132.6
## b  -0.2
```

```
hw1_pred <- predict(hw1, n.ahead = 6)
plot(hw1, hw1_pred)
```



Check that students called HoltWinters(..., gamma=FALSE) and check for RMSE below

(d) Compare the HoltWinters models (Q1c) predictions with `d1_wide_HELDOUT`, which contains the actual values for 2018/2019. What is the mean sum of squared error for these 6 predicted data points? Take the square root of that, which gives the root-mean-squared-error, or RMSE. Report the RMSE.

$$RMSE = \sqrt{\frac{1}{n}\sum_i (\hat{y}_i - y_i)^2}$$

Make a plot of the Holt-Winters predictions and the actual values in `d1_wide_HELDOUT`, both on the y axis and with time on the horizontal axis. Use colors and/or linetypes to differentiate, and include a legend.

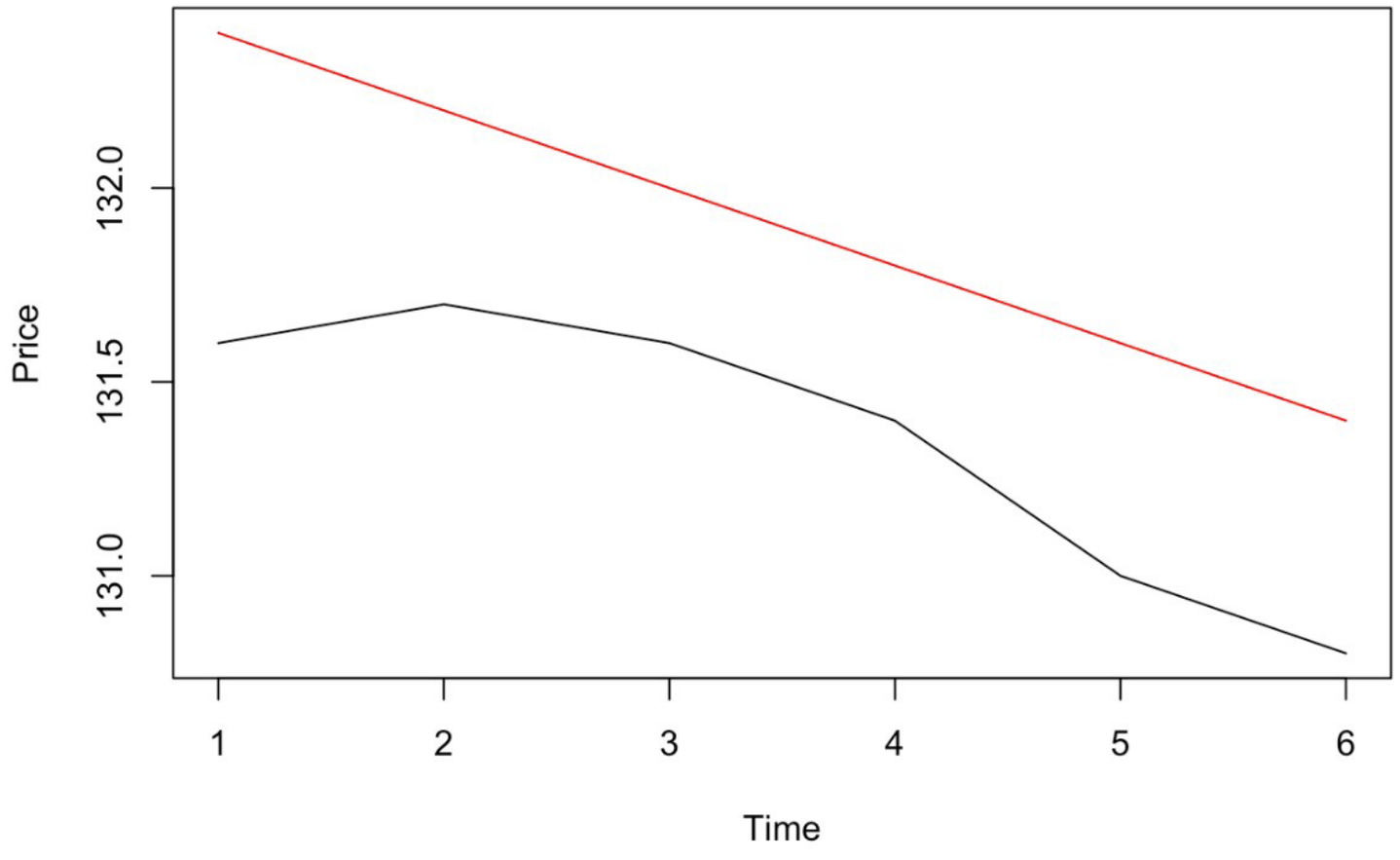
```
#sum_squared_errors_hw1 = mean(as.numeric((hw1_pred[1:6] - as.vector(d1_wide_HELDOUT))^2))
sum_squared_errors_hw1 = mean(as.numeric((hw1_pred[1:6] - d1_wide_HELDOUT)^2))
sum_squared_errors_hw1
```

```
## [1] 0.3216667
```

```
rmse_hw1 = sqrt(sum_squared_errors_hw1)
rmse_hw1
```

```
## [1] 0.5671567
```

```
plot_min_value = min(c(hw1_pred[1:6], unlist(as.vector(d1_wide_HELDOUT))))
plot_max_value = max(c(hw1_pred[1:6], unlist(as.vector(d1_wide_HELDOUT))))
plot(1:6,
     hw1_pred[1:6],
     type='l',
     col="red",
     ylim=c(plot_min_value, plot_max_value),
     xlab = "Time",
     ylab = "Price")
lines(1:6, as.vector(d1_wide_HELDOUT), type="l", col="black")
legend(x=1, y=135, legend=c("Holt-Winters", "Actual"), col=c("red", "black"), lty=1)
```



Discussion

```
# checking if students ran with the gamma.
hw1_with_gamma_pred = predict(HoltWinters(d1_ts), n.ahead=6)
#sum_squared_errors_hw1_with_gamma = mean(as.numeric((hw1_with_gamma_pred[1:6] - as.vector(d1_wide_HELDOUT))^2))
sum_squared_errors_hw1_with_gamma = mean(as.numeric(hw1_with_gamma_pred[1:6] - (d1_wide_HELDOUT))^2)
rmse_hw1_with_gamma_pred = sqrt(sum_squared_errors_hw1_with_gamma)
```

- Sum of squared errors = 0.3216667, RMSE = 0.5671567
- Interpret the magnitude of RMSE with respect to the actual value.
- Note: Need to correctly fit a HW model without the seasonal component (e.g. gamma=FALSE)
- Bonus: why no SMA forecast?

(1e) For the second-half of this question we shall be using a dataset that's available in R. Load in the dataset using `data(ChickWeight)`. The dataset will then be stored in a variable called `ChickWeight`

There are 4 variables in this long-form dataset, with 578 observations, that comes from a longitudinal experiment in which chicks (baby chickens) were given different types of diets since birth, and the chicks' weights were measured at various time-points. The variables are:

- `weight`. Body weight of the chick at that time point (in grams).
- `Time`. A numeric variable, measuring days since birth at the time of weight measurement.
- `Chick`. A `factor` that represents a unique Chick. There are in total 50 unique chicks.
- `Diet`. A `factor` with levels 1,2,3,4 that represents the diet that the chicks were fed.

```
data(ChickWeight)
head(ChickWeight)
```

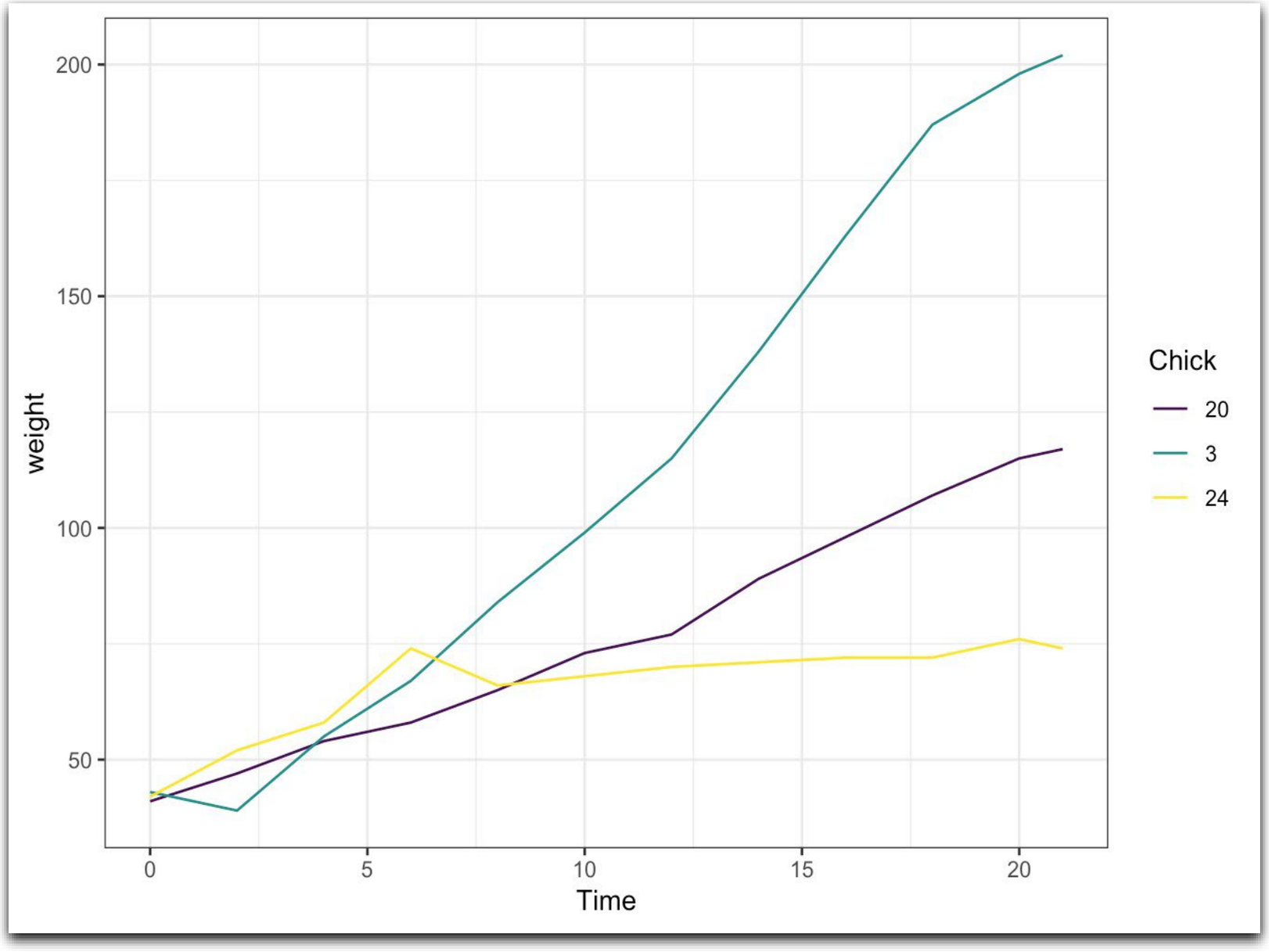
```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

First, let's plot some time-series data. Plot the weight-vs.-time graphs for Chick numbers: 3, 20, 24. Put them all on the same graph, make sure each chick's data is connected by a line, and label each line accordingly.

Which diet did each of these 3 chicks take?

```
subset2a = subset(ChickWeight, ChickWeight$Chick %in% c("3", "20", "24"))

ggplot(subset2a, aes(x=Time, y=weight, group=Chick, color=Chick)) + geom_line() +
  theme_bw()
```

Which diet did each of these 3 chicks take?



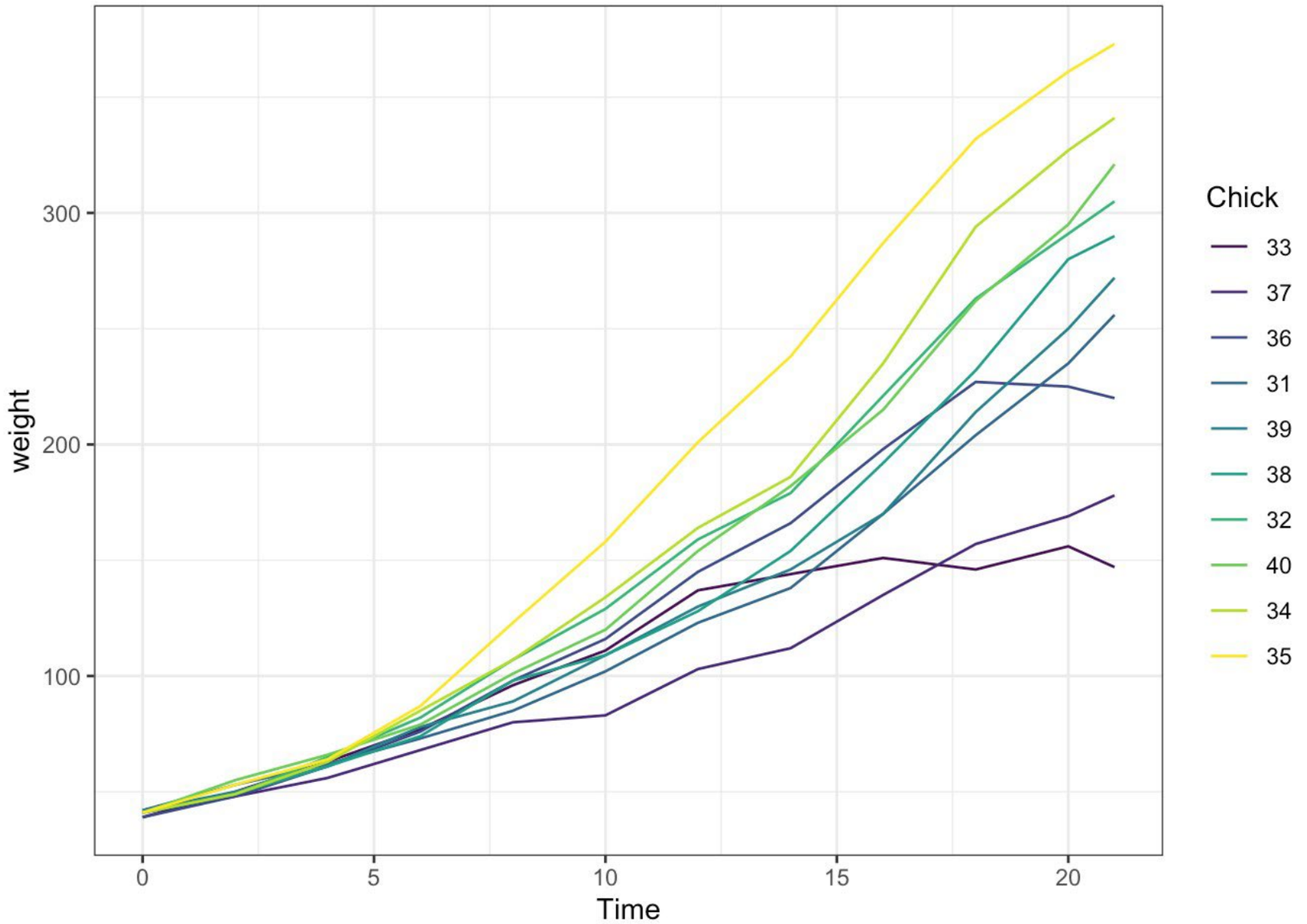
Chick 3 and 20 took Diet 1, and Chick 24 took Diet 2 respectively.

(1f) Make a subset of all the chicks that took `Diet 3`. For this subset, fit a linear model predicting `weight` just using `Time`. This is a regression-based time-series model where our predictor, our "X", is just an index now that represents time. Interpret the intercept and slope coefficients.

```
subset2b = ChickWeight %>% filter(Diet %in% c("3"))

summary(lm(weight ~ Time, subset2b))
```

```
##
## Call:
## lm(formula = weight ~ Time, data = subset2b)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.131  -19.056   -1.865   15.484  114.869
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   18.250      6.617   2.758  0.00674 **
## Time          11.423      0.515  22.181  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.22 on 118 degrees of freedom
## Multiple R-squared:  0.8066, Adjusted R-squared:  0.8049
## F-statistic:   492 on 1 and 118 DF,  p-value: < 2.2e-16
```

```
ggplot(subset2b, aes(x=Time, y=weight, group=Chick, color=Chick)) + geom_line() +
  theme_bw()
```



Intercept: 18.250. This means average weight of a chick at birth is 18.25g.
Slope: 11.423. This means that, on average, chicks put on 11.42g per day.

1(g) Now let's look at two groups. Make a subset of `chicks` who are on `Diet` 3 and `Diet` 1. Make a dummy variable to indicate which `Diet` they are on. (To give you some practice in manipulating variables, let's say that 3 is the reference group, and this dummy variable should be 1 if the Chick is on Diet 1 and 0 if the Chick is on Diet 3).

If I'm interested in seeing whether the `Diet` affects `chick`s' growth, what is the linear regression model I should test? Run this model, and interpret the coefficients on the variables in the model. Which of the two `Diet`s seem to be better for growth?

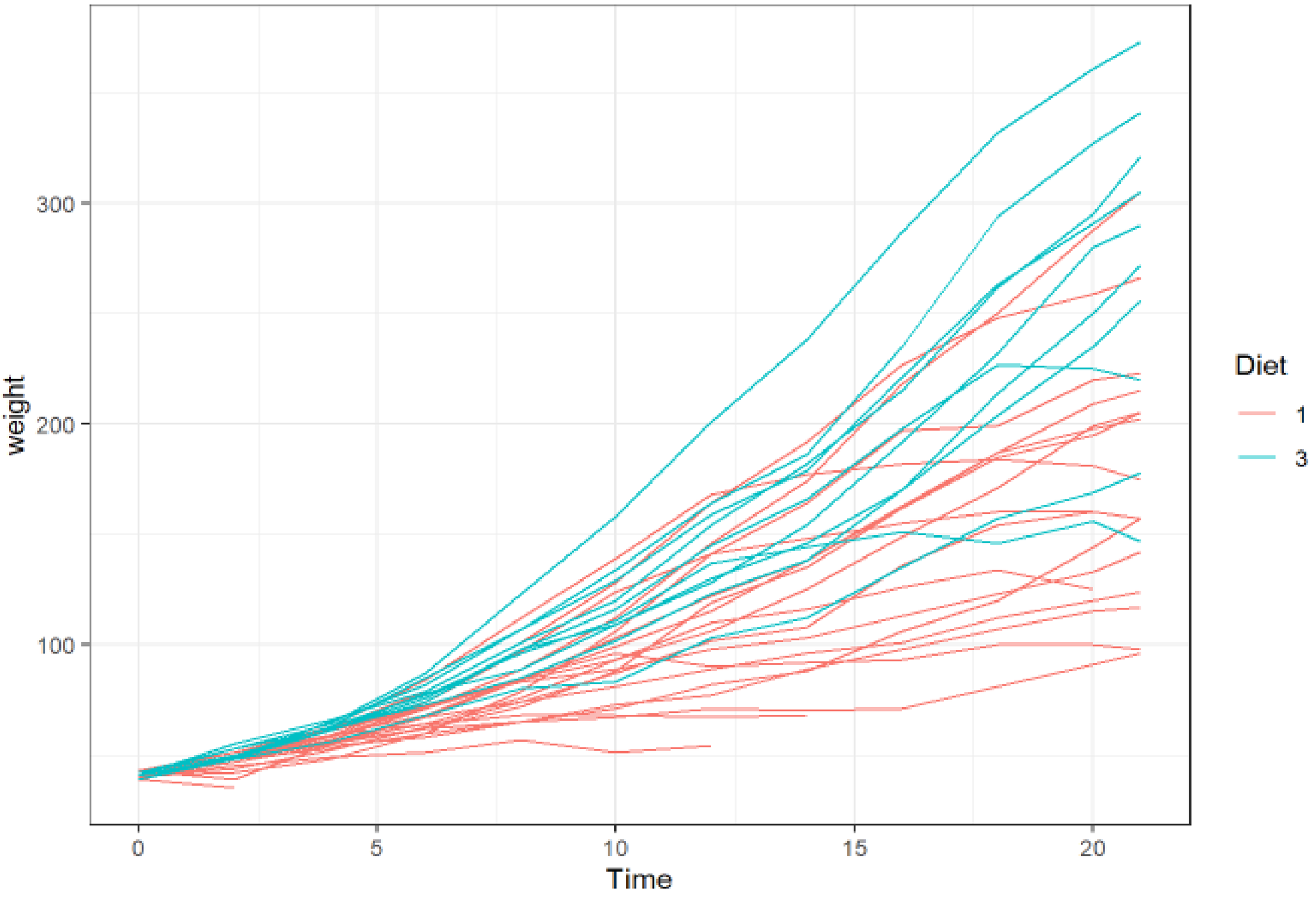
(Challenge: Try to plot all the `Chick`s in this analysis on the same plot. Similar to Q1f, make sure each `Chick` corresponds to one line. But for this graph, let's colour the lines by `Diet`.)

```r
subset2c = ChickWeight %>% filter(Diet %in% c("3", "1")) %>%
  mutate(Dummy = factor(Diet, levels=c("3", "1"), labels=c("3", "1")))

summary(lm(weight ~ Time*Dummy, subset2c))
```

```
##
## Call:
## lm(formula = weight ~ Time * Dummy, data = subset2c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.131  -17.609   -0.942   11.933  130.391
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.2503     6.0292   3.027  0.00266 **
## Time         11.4229     0.4693  24.342  < 2e-16 ***
## Dummy1       12.6807     7.4299   1.707  0.08880 .
## Time:Dummy1  -4.5811     0.5845  -7.838  6.1e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.83 on 336 degrees of freedom
## Multiple R-squared:  0.7631, Adjusted R-squared:  0.761
## F-statistic: 360.7 on 3 and 336 DF,  p-value: < 2.2e-16
```

- We specify an **interaction term** in our model
- Intercept is 18.250
  - When the value of all predictors are 0 and the `chick` is on `Diet` 3, the mean value of weight is 18.25g
  - What does it mean for all predictors to be zero in this case?
  - "The average weight of a chick at birth on Diet 3 is 18.25g" -> sensible?

```
ggplot(subset2c, aes(x=Time, y=weight, group=Chick, color=Diet)) + geom_line() +
  theme_bw()
```



1(h) Finally, let's look at all four `Diet`s. Now, let's use the full dataset. `ChickWeight$Diet` is already a factor, so let's just use `Diet` as the moderator, and see if `Diet` moderates growth rates.

You should be running the same `lm()` model.

What is the reference group of `ChickWeight$Diet`?

How do you interpret each interaction coefficients? Which seems to be the best `Diet` for growth (i.e., with the highest growth rate)? Which seems to be the worst?

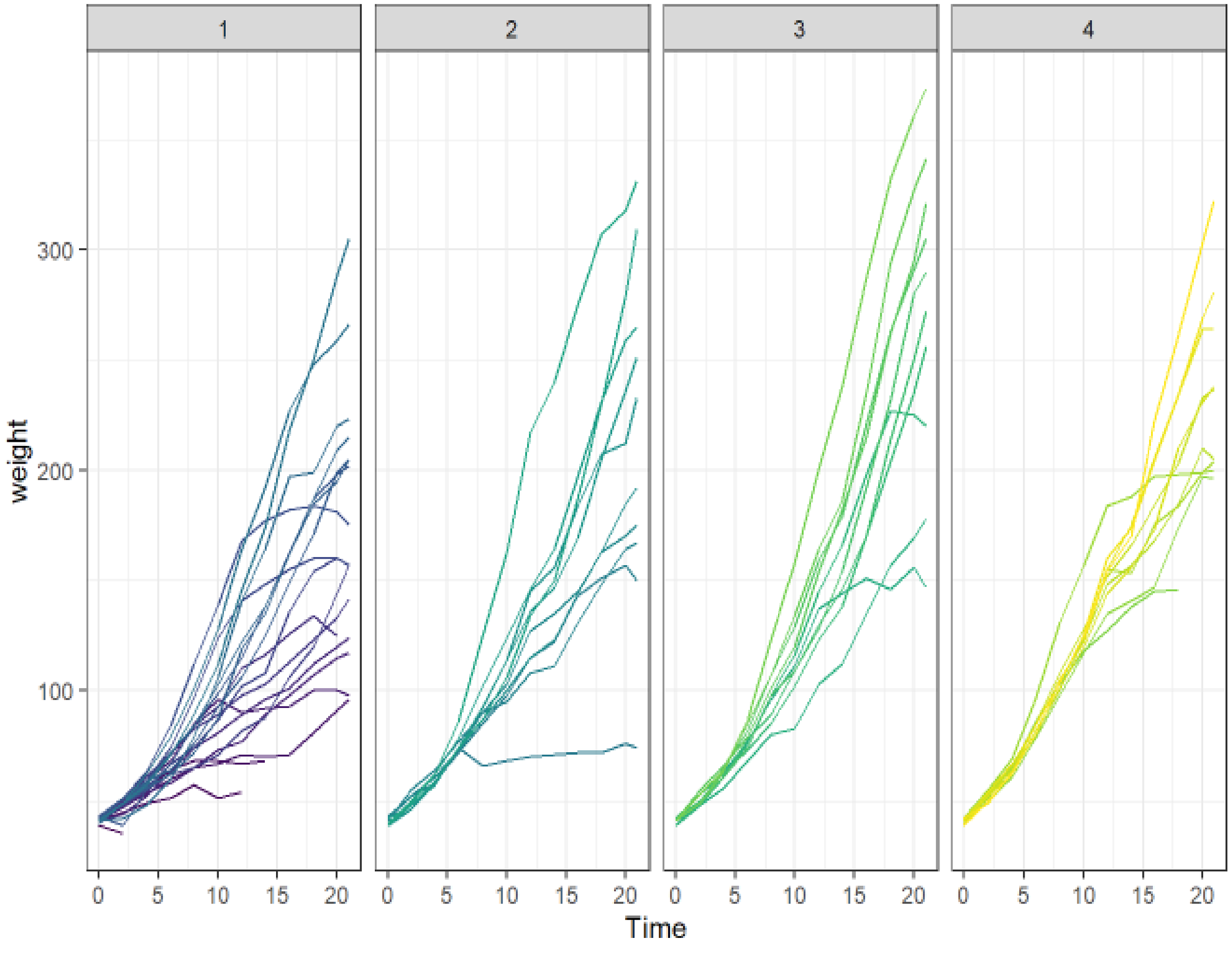
(Challenge points for plotting all these results. One way I would recommend visualizing them is putting all the `chick`s in one `Diet` on one graph, and have four graphs side-byside. If you use ggplot it's called facet-ing. Your tutor will show this graph in class using `ggplot`, but I will leave this as a bonus challenge for you.)

```
summary(lm(weight ~ Time*Diet, ChickWeight))
```

```
##
## Call:
## lm(formula = weight ~ Time * Diet, data = ChickWeight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -135.425  -13.757   -1.311   11.069  130.391
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.9310     4.2468   7.283 1.09e-12 ***
## Time          6.8418     0.3408  20.076  < 2e-16 ***
## Diet2        -2.2974     7.2672  -0.316  0.75202
## Diet3       -12.6807     7.2672  -1.745  0.08154 .
## Diet4        -0.1389     7.2865  -0.019  0.98480
## Time:Diet2    1.7673     0.5717   3.092  0.00209 **
## Time:Diet3    4.5811     0.5717   8.014 6.33e-15 ***
## Time:Diet4    2.8726     0.5781   4.969 8.92e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.07 on 570 degrees of freedom
## Multiple R-squared:  0.773,  Adjusted R-squared:  0.7702
## F-statistic: 277.3 on 7 and 570 DF,  p-value: < 2.2e-16
```

What if you wanted a different reference group? (When might you want to use a specific value as your reference group?)

```
ggplot(ChickWeight, aes(x=Time, y=weight, group=Chick, color=Chick)) + geom_line() +
  facet_grid(~Diet) + theme_bw() + theme(legend.position = "None")
```



Intercept: 30.93 This means average weight of a chick ON DIET 1 at birth is 30.93g.

Coefficient on Time: 6.8418

This means that, on average, chicks ON DIET 1 put on 6.84g per day.

Coefficient on Diet2: -2.2974.

This means that, on average, at birth chicks on Diet 2 are 2.29g lighter than chicks on Diet 1 (reference group).

Coefficient on Diet3: -12.6807.

This means that, on average, at birth chicks on Diet 3 are 12.68g lighter than chicks on Diet 1 (reference group).

Coefficient on Diet4: -0.1389.

This means that, on average, at birth chicks on Diet 4 are 0.14g lighter than chicks on Diet 1 (reference group).

Coefficient on Time:Diet2 : 1.76.

This means that, on average, chicks ON DIET 2 put on 1.76g per day MORE than chicks on Diet 1.

Coefficient on Time:Diet3 : 4.58.

This means that, on average, chicks ON DIET 3 put on 4.58g per day MORE than chicks on Diet 1.

Coefficient on Time:Diet4 : 2.87.

This means that, on average, chicks ON DIET 4 put on 2.87g per day MORE than chicks on Diet 1.