Project #4

Term Structure Estimation

Purpose

This project will use recent market data to support a decision regarding the provision of captive financing to customers. We will focus on building and interpreting a forward curve, testing more than one model, plotting results, and applying the results to a a collateral management scenario.

Problem

As we researched how to provide captive financing and insurance for our customers, we found that we needed to understand the relationships among lending rates and various terms and conditions of typical equipment financing contracts. Now we will extend that analysis to the long term provision of financing terms for customers. This will involve building a monitoring and management system of revolving credit for customer accounts.

We will focus on one question:

> *What is the influence of the term structure on the management of collateral to support long term financial arrangements with customers?*

Data

The data set `termstrc.csv` contains data from the Wall Street Journal's market data site, which we will use to get some high level insights. The cross-sectional data covers the maturity date and price of STRIPS principal-only prices, otherwise known as zero coupon bond prices.

Variable	Description	Units of Measure
PRICE	market value of zero coupon bond	percent of face value
MATURITY	date at which zero coupon bond expires	date month/day/year

Source: Wall Street Journal marketdata.

Work Flow

1. Prepare and explore the data.

- Describe the STRIPS data and its collection. For example, visit the Wall Street Journal marketdata site. Include any information on the site to enhance the interpretation of results.
- Use `read.csv` to read the data into `R`. Be sure to set the working directory where the data resides. Use `na.omit()` to clean the data.

```r
# set working directory to source
# file location; /data is
# subdirectory with csv file
x.data <- read.csv("data/termstr20170127.csv")
head(x.data, n = 3)
```

```
##    PRICE MATURITY
## 1 99.991       18
## 2 99.795       46
## 3 99.769       77
```

```r
tail(x.data, n = 3)
```

```
##      PRICE MATURITY
## 189 39.621    10699
## 190 39.415    10791
## 191 39.302    10883
```

```r
names(x.data)
```

```
## [1] "PRICE"    "MATURITY"
```

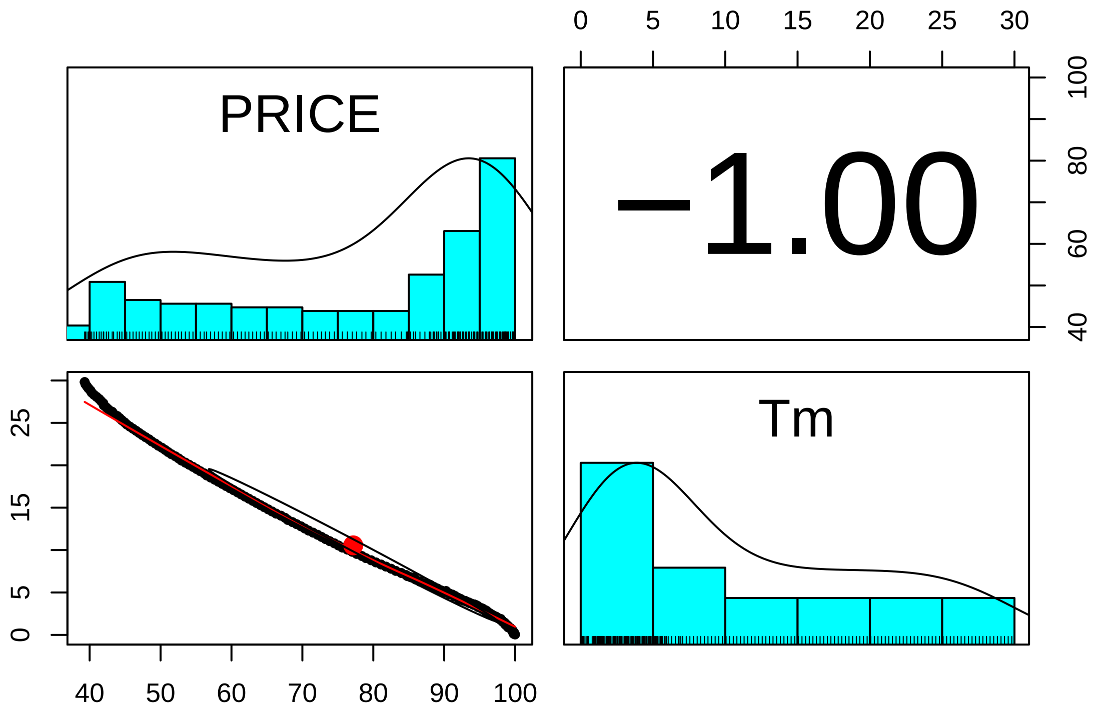
```r
x.data <- x.data[order(x.data$MATURITY),
    ]
x.data$Tm <- x.data$MATURITY/365  # express MATURITY in years
x.data <- x.data[, -2]
str(x.data)
```

```
## 'data.frame':    191 obs. of  2 variables:
##  $ PRICE: num  100 99.8 99.8 99.8 99.7 ...
##  $ Tm   : num  0.0493 0.126 0.211 0.2932 0.3781 ...
```

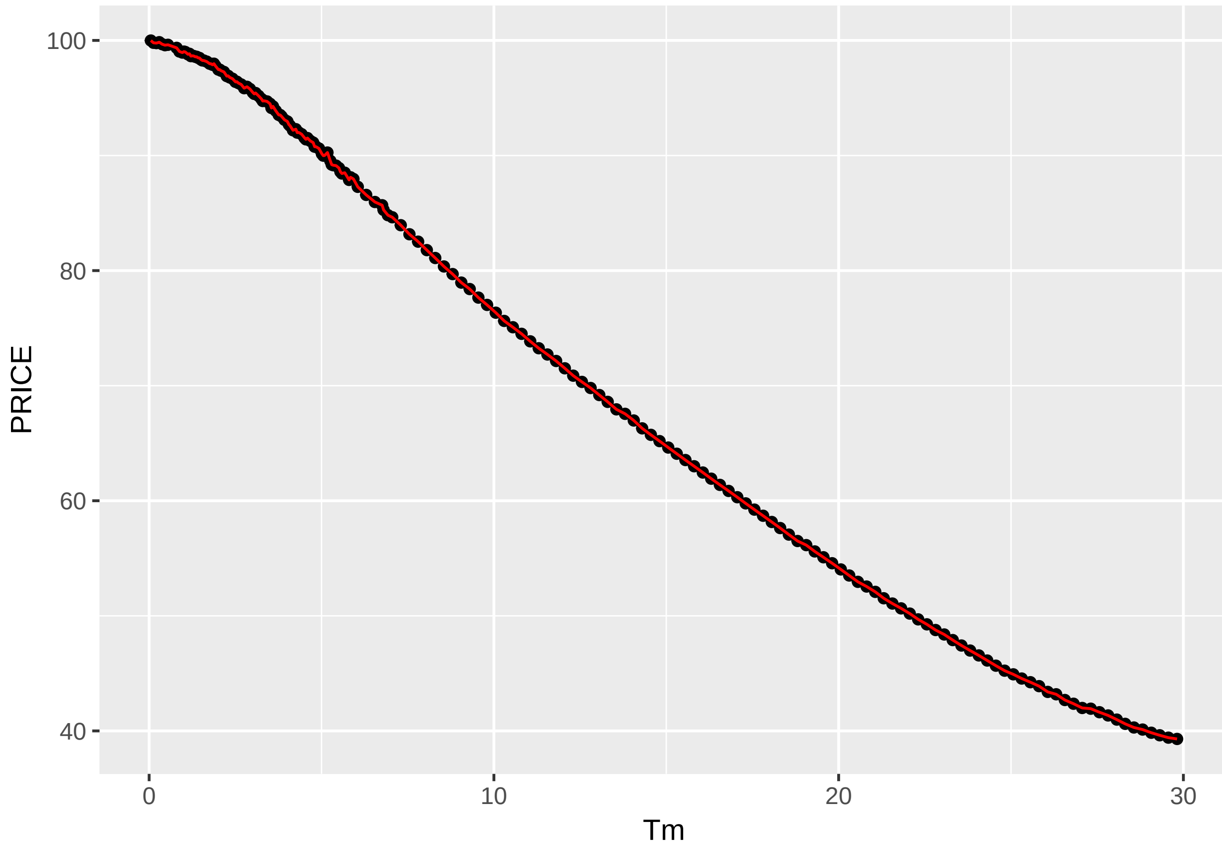
```r
summary(x.data)
```

```
##      PRICE             Tm
##  Min.   :39.30   Min.   : 0.04931
##  1st Qu.:58.43   1st Qu.: 3.06849
##  Median :85.96   Median : 6.54795
##  Mean   :77.17   Mean   :10.55737
##  3rd Qu.:95.39   3rd Qu.:17.93425
##  Max.   :99.99   Max.   :29.81644
```

```r
require(psych)
pairs.panels(x.data)
```




```r
require(ggplot2)
ggplot(data = x.data, aes(x = Tm, y = PRICE)) +
    geom_point() + geom_line(colour = "red")
```




```
# plot(x.data$MATURITY, x.data$PRICE,
# main = 'STRIPS', xlab = 'Maturity',
# ylab = 'Price')
```

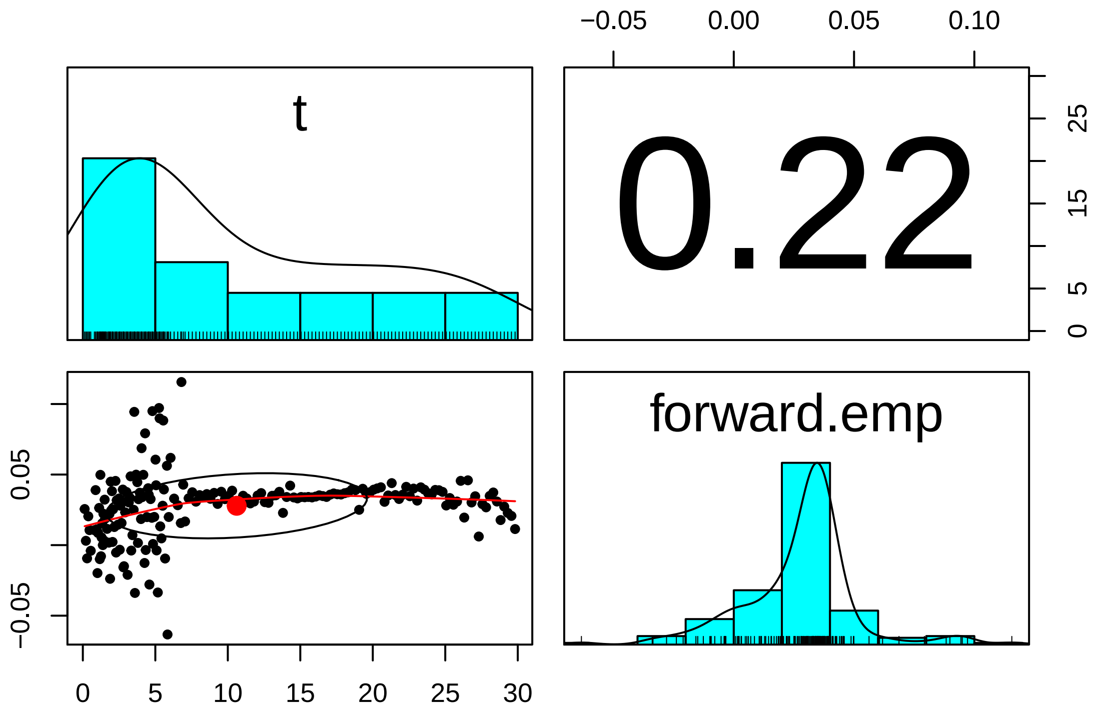
2. Analyze the data

- Estimate the empirical forward curve and interpret.

```
t <- x.data$Tm[-1]  # seq(0,30,length = 100)
forward.emp <- -diff(log(x.data$PRICE))/diff(x.data$Tm)
forward.emp.df <- data.frame(t = t, forward.emp = forward.emp)
str(forward.emp.df)
```

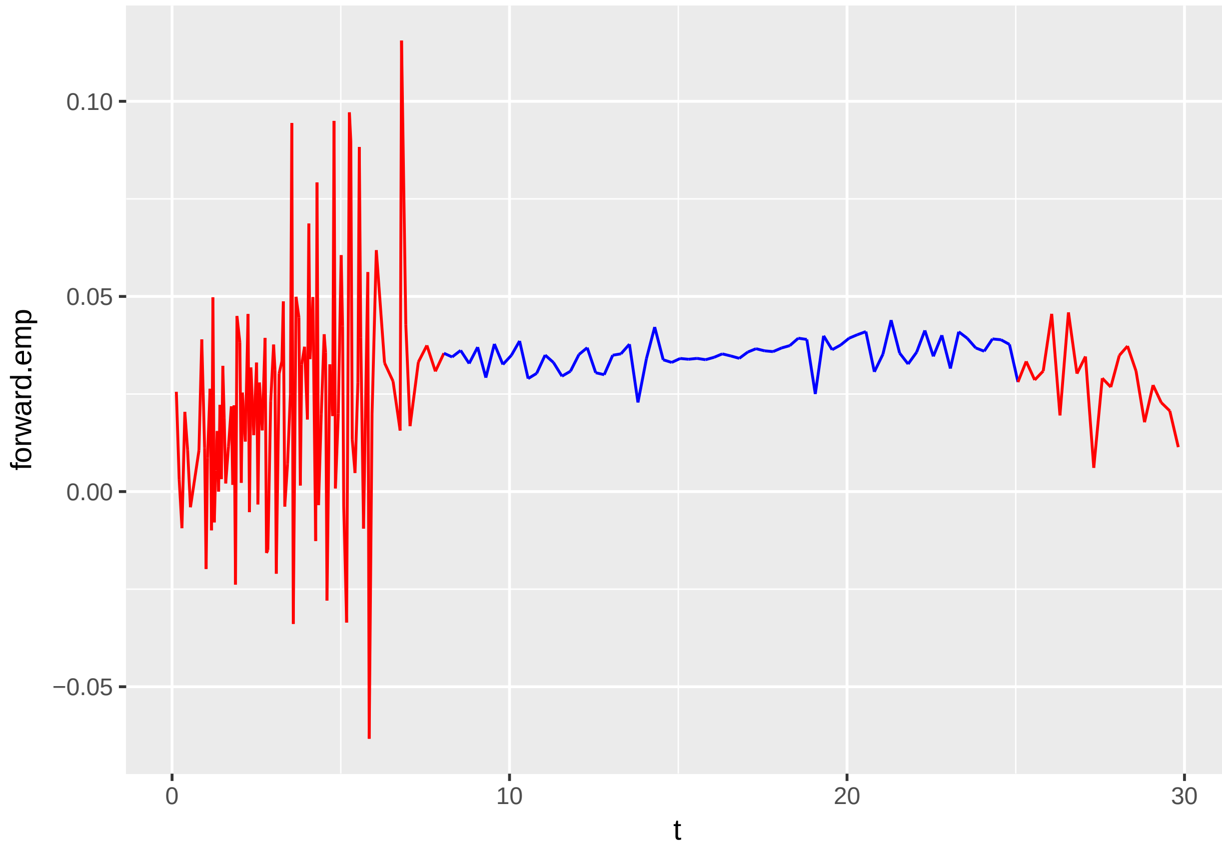
```
## 'data.frame':    190 obs. of  2 variables:
##  $ t          : num  0.126 0.211 0.293 0.378 0.46 ...
##  $ forward.emp: num  0.02558 0.00307 -0.00939 0.02042 0.01062 ...
```

```
pairs.panels(forward.emp.df)
```



We see significant clustering and volatility across maturities less than eight years. More stability is found from years eight through twenty-five. This suggests at least two knots, one at maturity of eight years and the other at twenty-five years. Effectively there appear to be at least three distributions of forward rates conditional on maturities in this data.

```
col.emp <- ifelse(forward.emp.df$t <
    8, "red", ifelse(forward.emp.df$t >
    25, "red", "blue"))
forward.emp.df <- data.frame(t = t, forward.emp = forward.emp,
    col.emp = col.emp)
ggplot(forward.emp.df, aes(x = t, y = forward.emp)) +
    geom_line(colour = col.emp, group = 1) +
    scale_colour_identity()
```



- Fit a cubic spline to the zero price and maturity combinations.

```
fit.spline <- nls(PRICE ~ 100 * exp(-theta_0 *
    Tm - (theta_1 * Tm^2)/2 - (theta_2 *
    Tm^3)/3 - (Tm < 8) * (theta_3 * (Tm -
    8)^3)/3 - (Tm > 25) * (theta_4 *
    (Tm - 25)^3)/3), data = x.data, start = list(theta_0 = 0.01,
    theta_1 = 0.0024, theta_2 = -1e-04,
    theta_3 = 1e-05, theta_4 = 1e-04))
```

- Fit 5% and 95% envelopes.

```
require(quantreg)
taus <- c(0.05, 0.5, 0.75)
fit.spline.q.05 <- nlrq(PRICE ~ 100 *
    exp(-theta_0 * Tm - (theta_1 * Tm^2)/2 -
        (theta_2 * Tm^3)/3 - (Tm < 8) *
        (theta_3 * (Tm - 8)^3)/3 - (Tm >
        25) * (theta_4 * (Tm - 25)^3)/3),
    data = x.data, tau = taus[1], start = list(theta_0 = 0.01,
        theta_1 = 0.0024, theta_2 = -1e-04,
        theta_3 = 1e-05, theta_4 = 1e-04))
fit.spline.q.50 <- nlrq(PRICE ~ 100 *
    exp(-theta_0 * Tm - (theta_1 * Tm^2)/2 -
        (theta_2 * Tm^3)/3 - (Tm < 8) *
        (theta_3 * (Tm - 8)^3)/3 - (Tm >
        25) * (theta_4 * (Tm - 25)^3)/3),
```

```
    data = x.data, tau = taus[2], start = list(theta_0 = 0.01,
        theta_1 = 0.0024, theta_2 = -1e-04,
        theta_3 = 1e-05, theta_4 = 1e-04))
fit.spline.q.75 <- nlrq(PRICE ~ 100 *
    exp(-theta_0 * Tm - (theta_1 * Tm^2)/2 -
        (theta_2 * Tm^3)/3 - (Tm < 8) *
        (theta_3 * (Tm - 8)^3)/3 - (Tm >
        25) * (theta_4 * (Tm - 25)^3)/3),
    data = x.data, tau = taus[3], start = list(theta_0 = 0.01,
        theta_1 = 0.0024, theta_2 = -1e-04,
        theta_3 = 1e-05, theta_4 = 1e-04))
```

- Fit quadratic comparison.

```
fit.quad <- nls(PRICE ~ 100 * exp(-theta_0 *
    Tm - (theta_1 * Tm^2)/2 - (theta_2 *
    Tm^3)/3), data = x.data, start = list(theta_0 = 0.01,
    theta_1 = 0.0024, theta_2 = -1e-04))
```

- What anomalies appear based on these procedures?

```
knitr::kable(summary(fit.spline)$coefficients,
    digits = 4)
```

	Estimate	Std. Error	t value	Pr(>|t|)
theta_0	0.0144	3e-04	54.7512	0.0000
theta_1	0.0028	1e-04	39.7473	0.0000
theta_2	-0.0001	0e+00	-26.5398	0.0000
theta_3	0.0001	0e+00	8.7560	0.0000
theta_4	0.0006	3e-04	2.2262	0.0272

```
(sigma.spline <- (summary(fit.spline)$sigma)^0.5)
```

```
## [1] 0.6778881
```

```
knitr::kable(summary(fit.spline.q.05)$coefficients,
    digits = 4)
```

	Value	Std. Error	t value	Pr(>|t|)
theta_0	0.0156	0.0011	13.6210	0.0000
theta_1	0.0028	0.0003	8.5454	0.0000
theta_2	-0.0001	0.0000	-6.4759	0.0000
theta_3	0.0000	0.0000	0.5548	0.5797
theta_4	0.0005	0.0005	1.0171	0.3104

```
knitr::kable(summary(fit.spline.q.50)$coefficients,
    digits = 4)
```

	Value	Std. Error	t value	Pr(>|t|)
theta_0	0.0148	4e-04	35.0484	0.0000
theta_1	0.0026	1e-04	23.5406	0.0000
theta_2	-0.0001	0e+00	-15.4073	0.0000

	Value	Std. Error	t value	Pr(>|t|)
theta_3	0.0001	0e+00	4.9993	0.0000
theta_4	0.0000	3e-04	0.0702	0.9441

```
knitr::kable(summary(fit.spline.q.75)$coefficients,
    digits = 4)
```

	Value	Std. Error	t value	Pr(>|t|)
theta_0	0.0139	4e-04	36.6154	0.0000
theta_1	0.0027	1e-04	33.3238	0.0000
theta_2	-0.0001	0e+00	-22.5341	0.0000
theta_3	0.0001	0e+00	8.3496	0.0000
theta_4	0.0001	2e-04	0.3105	0.7565

```
knitr::kable(summary(fit.quad)$coefficients,
    digits = 4)
```

	Estimate	Std. Error	t value	Pr(>|t|)
theta_0	0.0137	3e-04	51.5601	0
theta_1	0.0029	1e-04	43.4853	0
theta_2	-0.0001	0e+00	-30.7040	0

```
(sigma.quad <- (summary(fit.quad)$sigma)^0.5)
```

```
## [1] 0.7453241
```

- Build forward curves.

```
coef.spline <- summary(fit.spline)$coef[,
    1]
forward.spline <- coef.spline[1] + coef.spline[2] *
    t + coef.spline[3] * t^2 + (t < 8) *
    coef.spline[4] * (t - 8)^2 + (t >
    25) * coef.spline[5] * (t - 25)^2
coef.spline <- summary(fit.spline.q.05)$coef[,
    1]
forward.spline.q.05 <- coef.spline[1] +
    coef.spline[2] * t + coef.spline[3] *
    t^2 + (t < 8) * coef.spline[4] *
    (t - 8)^2 + (t > 25) * coef.spline[5] *
    (t - 25)^2
coef.spline <- summary(fit.spline.q.50)$coef[,
    1]
forward.spline.q.50 <- coef.spline[1] +
    coef.spline[2] * t + coef.spline[3] *
    t^2 + (t < 8) * coef.spline[4] *
    (t - 8)^2 + (t > 25) * coef.spline[5] *
    (t - 25)^2
coef.spline <- summary(fit.spline.q.75)$coef[,
    1]
```

```
forward.spline.q.75 <- coef.spline[1] +
    coef.spline[2] * t + coef.spline[3] *
    t^2 + (t < 8) * coef.spline[4] *
    (t - 8)^2 + (t > 25) * coef.spline[5] *
    (t - 25)^2
coef.quad <- summary(fit.quad)$coef[,
    1]
forward.quad <- coef.spline[1] + coef.spline[2] *
    t + coef.spline[3] * t^2
str(forward.spline)
```

```
##  num [1:190] 0.0189 0.0191 0.0192 0.0193 0.0195 ...
```

```
str(forward.spline.q.50)
```

```
##  num [1:190] 0.0206 0.0207 0.0208 0.0209 0.021 ...
```

```
str(forward.quad)
```

```
##  num [1:190] 0.0143 0.0145 0.0147 0.0149 0.0152 ...
```

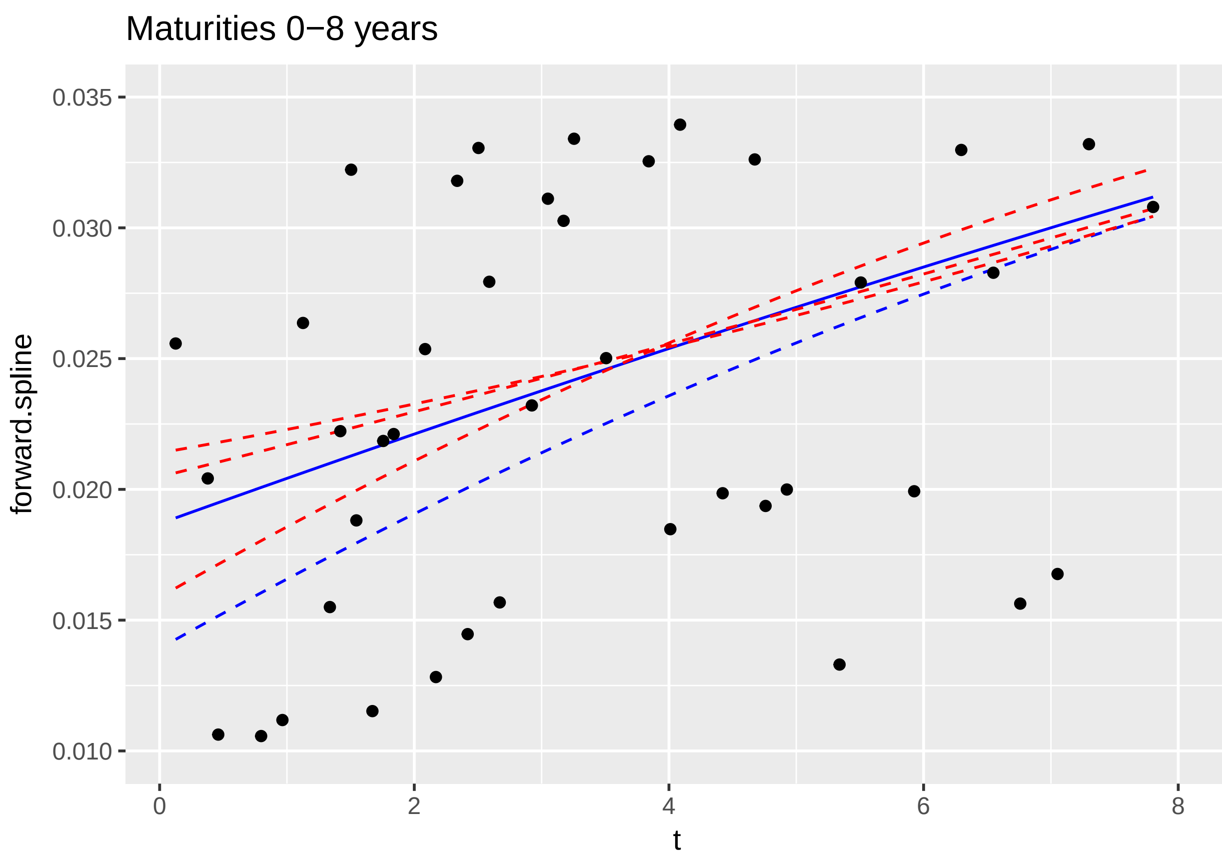
3. Interpret results

Pull the coefficients from a `summary()` of the `fit.spline` object for the plot. Zoom into the curves to see the action.

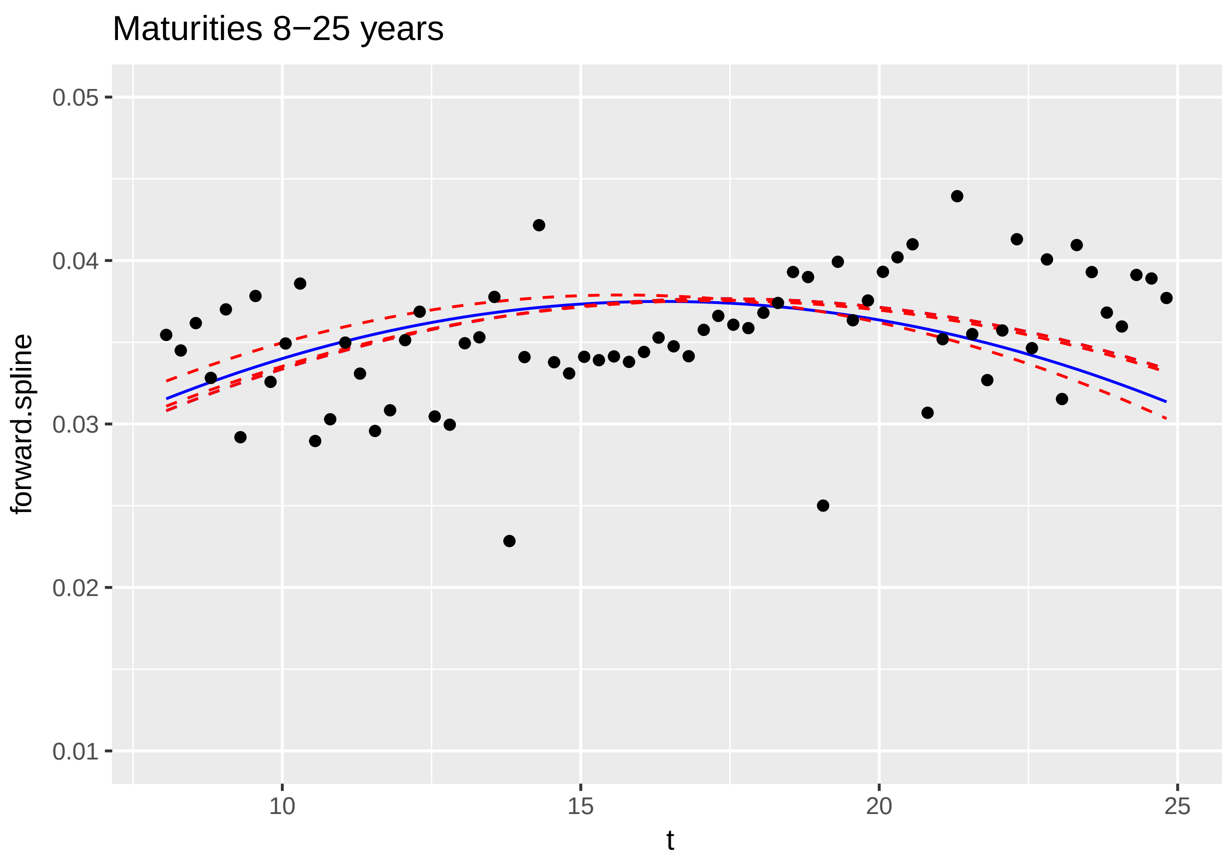
```
x.spl <- data.frame(t = t, forward.spline = forward.spline,
    forward.spline.q.05 = forward.spline.q.05,
    forward.spline.q.50 = forward.spline.q.50,
    forward.spline.q.75 = forward.spline.q.75,
    forward.quad = forward.quad, forward.emp = forward.emp)
str(x.spl)
```

```
## 'data.frame':    190 obs. of  7 variables:
##  $ t                  : num  0.126 0.211 0.293 0.378 0.46 ...
##  $ forward.spline     : num  0.0189 0.0191 0.0192 0.0193 0.0195 ...
##  $ forward.spline.q.05: num  0.0162 0.0165 0.0167 0.0169 0.0171 ...
##  $ forward.spline.q.50: num  0.0206 0.0207 0.0208 0.0209 0.021 ...
##  $ forward.spline.q.75: num  0.0215 0.0216 0.0216 0.0217 0.0218 ...
##  $ forward.quad       : num  0.0143 0.0145 0.0147 0.0149 0.0152 ...
##  $ forward.emp        : num  0.02558 0.00307 -0.00939 0.02042 0.01062 ...
```

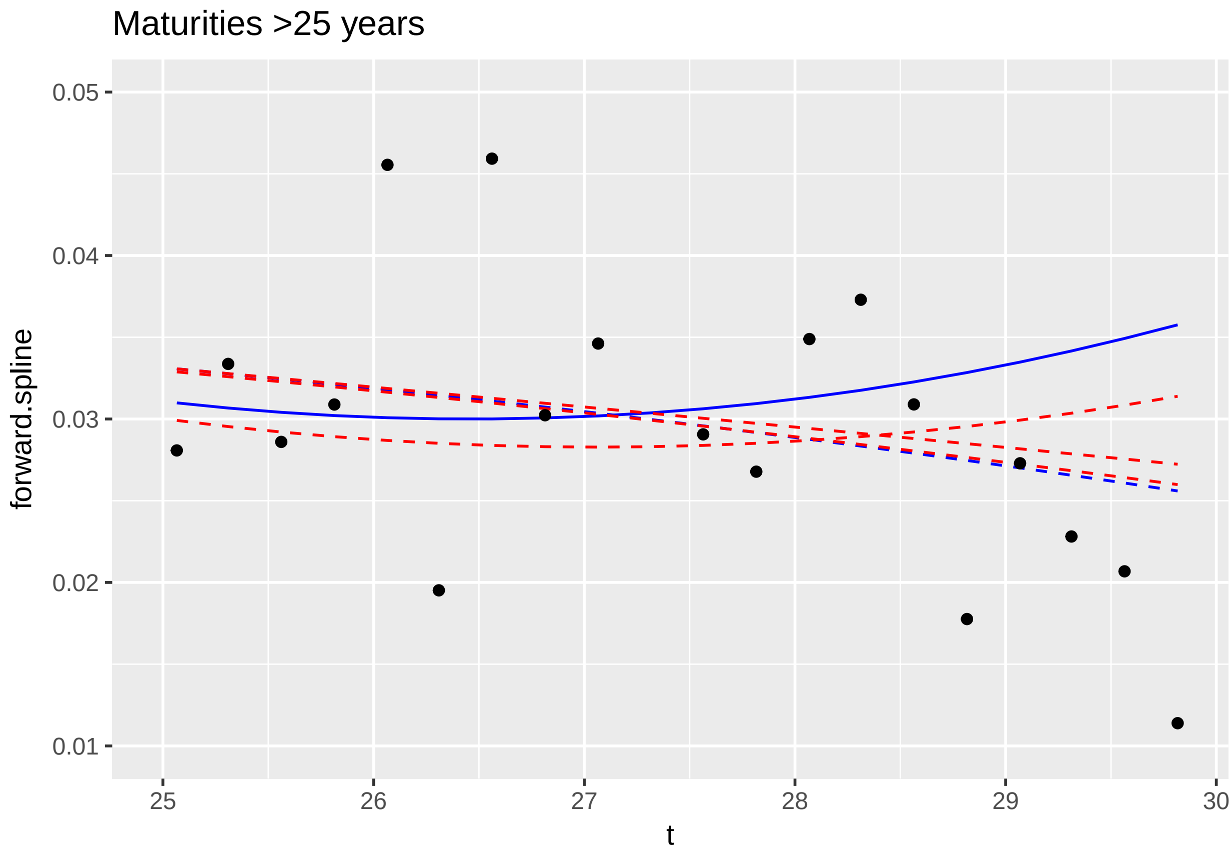
```
# 0-8 years maturity
ggplot(data = x.spl, aes(x = t, y = forward.spline)) +
    geom_line(colour = "blue") + geom_line(aes(x = t,
    y = forward.quad), colour = "blue",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.05), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.50), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.75), colour = "red",
    linetype = "dashed") + geom_point(aes(x = t,
    y = forward.emp)) + ylim(0.01, 0.035) +
    xlim(min(t), 8) + ggtitle("Maturities 0-8 years")
```




```
# 8-25 years maturity
ggplot(data = x.spl, aes(x = t, y = forward.spline)) +
    geom_line(colour = "blue") + geom_line(aes(x = t,
    y = forward.quad), colour = "blue",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.05), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.50), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.75), colour = "red",
    linetype = "dashed") + geom_point(aes(x = t,
    y = forward.emp)) + ylim(0.01, 0.05) +
    xlim(8, 25) + ggtitle("Maturities 8-25 years")
```

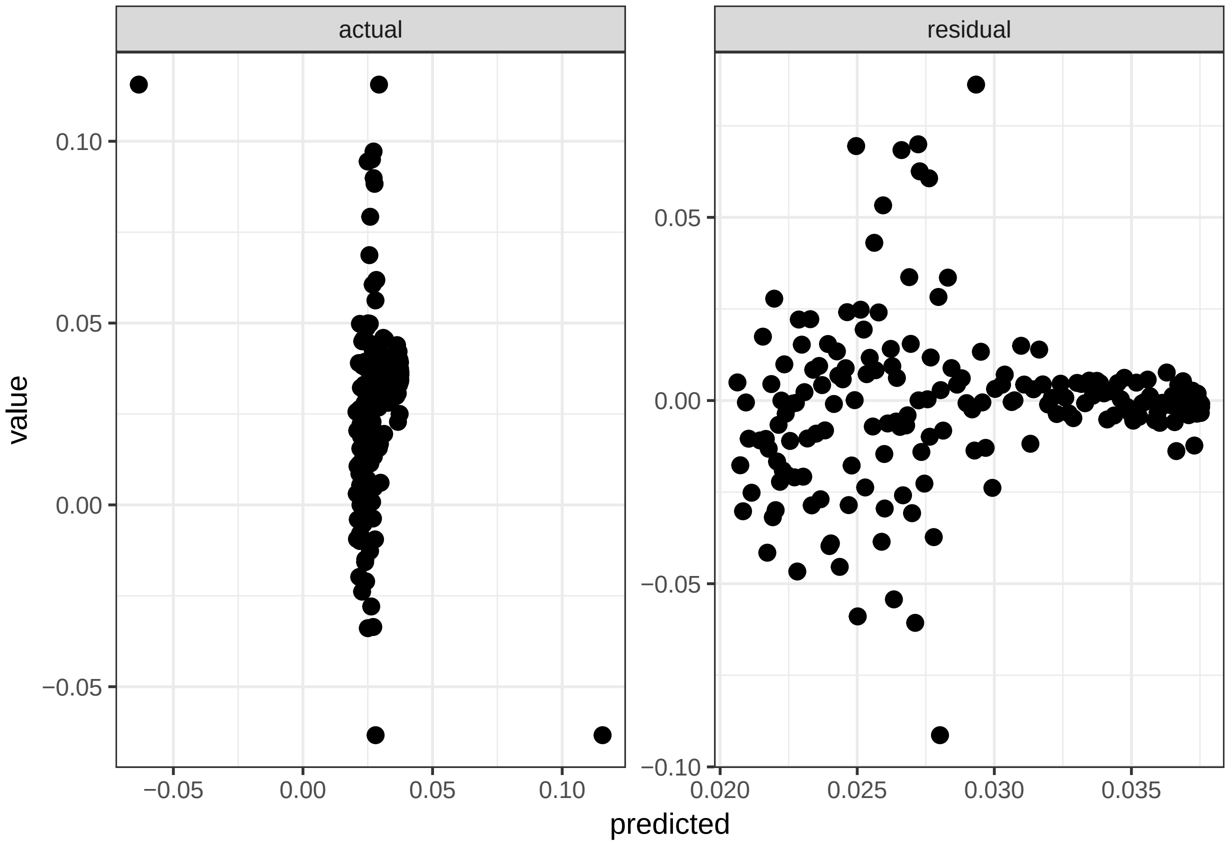



```r
# > 25 years maturity
ggplot(data = x.spl, aes(x = t, y = forward.spline)) +
    geom_line(colour = "blue") + geom_line(aes(x = t,
    y = forward.quad), colour = "blue",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.05), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.50), colour = "red",
    linetype = "dashed") + geom_line(aes(x = t,
    y = forward.spline.q.75), colour = "red",
    linetype = "dashed") + geom_point(aes(x = t,
    y = forward.emp)) + ylim(0.01, 0.05) +
    xlim(25, max(t)) + ggtitle("Maturities >25 years")
```



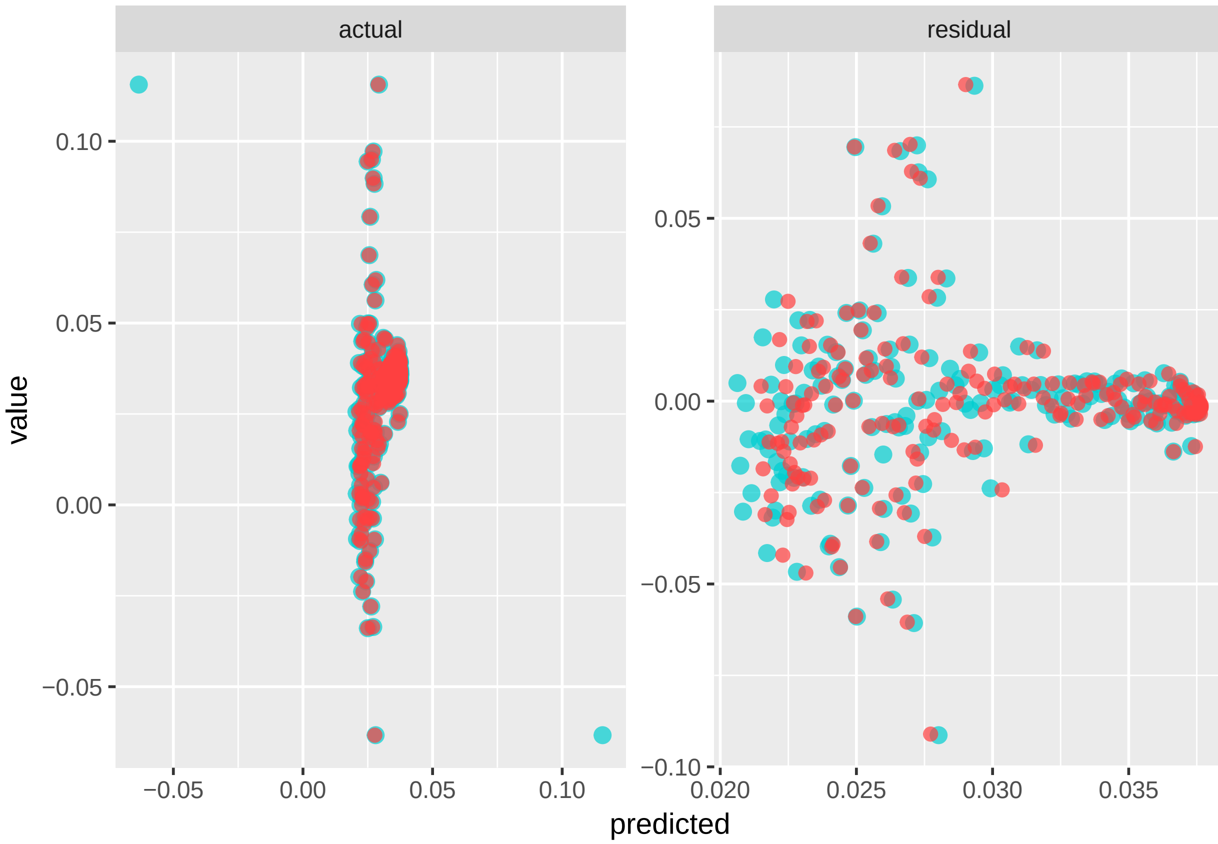
- Compare residuals to predicted for the median (50%) and quadratic models.

```
require(reshape2)
actual <- forward.emp
predicted <- forward.spline.q.50
residual <- actual - predicted
results <- data.frame(actual = actual,
    predicted = predicted, residual = residual)
# Insert comment here
min_xy <- min(min(results$actual), min(results$predicted))
max_xy <- max(max(results$actual), max(results$predicted))
# Insert comment here
plot.melt <- melt(results, id.vars = "predicted")
# Insert comment here
plot.data <- rbind(plot.melt, data.frame(predicted = c(min_xy,
    max_xy), variable = c("actual", "actual"),
    value = c(max_xy, min_xy)))
# Insert comment here
p <- ggplot(plot.data, aes(x = predicted,
    y = value)) + geom_point(size = 2.5) +
    theme_bw()
p <- p + facet_wrap(~variable, scales = "free")
p
```




```
# This function takes in 3 inputs
# INPUT1: a vector that holds actual
# values INPUT2: a vector that holds
# predicted values from a model
# INPUT3: a vector that holds
# predicted values from another model
# OUTPUT: a plot of actuals vs.
# predicted and residuals vs.
# predicted of both predicting models

p <- function(actual, pred.model1, pred.model2) {

    library(ggplot2)
    library(reshape2)

    # calculate the residuals from actual
    # vs. pred.model1 then create a data
    # frame
    residual <- actual - pred.model1
    results <- data.frame(actual = actual,
        predicted = pred.model1, residual = residual)

    # convert the results data frame to
    # long format
    plot.melt <- melt(results, id.vars = "predicted")
```

```
    # create 2 extra data points for
    # plotting purposes only
    min_xy <- min(min(results$actual),
        min(results$predicted))
    max_xy <- max(max(results$actual),
        max(results$predicted))

    # bind plot.melt and the 2 new data
    # points into a single data frame for
    # plotting
    plot.data <- rbind(plot.melt, data.frame(predicted = c(min_xy,
        max_xy), variable = c("actual",
        "actual"), value = c(max_xy,
        min_xy)))

    # calculate the residuals from actual
    # vs. pred.model2. then create a data
    # frame
    residual <- actual - pred.model2
    results <- data.frame(actual = actual,
        predicted = pred.model2, residual = residual)

    # convert the results data frame to
    # long format
    plot.melt2 <- melt(results, id.vars = "predicted")

    # plot both of plot.data and
    # plot.melt2 data frames
    p <- ggplot(plot.data, aes(x = predicted,
        y = value)) + geom_point(size = 2.5,
        col = "darkturquoise", alpha = 0.7) +
        geom_point(data = plot.melt2,
            aes(x = predicted, y = value),
            col = "brown1", alpha = 0.7,
            size = 2) + facet_wrap(~variable,
        scales = "free")

    return(p)

}

# plot forward.spline.q.50 and
# forward.spline.q.75
p(forward.emp, forward.spline.q.50, forward.spline.q.75)
```




```
# call this p() function on any other
# combinations of predictive models
# you want
```

4. Generate scenarios

- Suppose we just bought a 10 year maturity zero-coupon bond to satisfy collateral requirements for workers' compensation in the (great) State of New York.
- The forward rate can been estimated using a spline, a pure quadratic, and across quantiles. We believe that the 50%tile of zero-coupon bond prices is the appropriate threshold for estimation.

$$r_{0.50}(T) = \theta_{0,0.50}T + \theta_{1,0.50}\frac{T^2}{2} + \theta_{2,0.50}\frac{T^3}{3} + \theta_{3,0.50}\frac{(T-8)^3_+}{3} + \theta_{4,0.50}\frac{(T-25)^3_+}{3}$$

We will estimate this curve and explain in detail the choice of a particular model.

3. In 6 months there is a very high likelihood we will need to exit all business in New York State, have no employees that can claim workers' compensation. If we do so, we will sell the 10 year maturity zero-coupon bond. We project that the forward curve will be at the 75%tile of the current data, so that

$$r_{0.75}(T) = \theta_{0,0.75}T + \theta_{1,0.75}\frac{T^2}{2} + \theta_{2,0.75}\frac{T^3}{3} + \theta_{3,0.75}\frac{(T-8)^3_+}{3} + \theta_{4,0.50}\frac{(T-25)^3_+}{3}$$

We will estimate this curve and explain in detail the choice of a particular model.

4. How much would we gain or lose on this transaction at our exit?

Let's recall the following:

1. The forward rate is the rate of change of the yield-to-maturity
2. This means we integrate (i.e., take the cumulative sum of) forward rates to get the yield

3. The cumulative sum would then be some maturity times the components of the yield curve adjusted for the slope of the forward curve (the terms in Tm).
4. This adjustment is just one-half (1/2) of the slope term.

- Set up today's yield curve and the curve 6 months out.
- Using these yields we can compute the bond prices for today and for 6 months out as well.
- Our exit transaction is long today's version of the bond and short the 6 month version. We then calculate the six month (0.5 year maturity) return.

$$R_{6m} = \frac{P(T-0.5) - P(T)}{P(T)} = \frac{P(T-0.5)}{P(T)} - 1$$

4. Interpret and present results.

- We will produce an `R Markdown` document with code chunks to document and interpret our results.
- The first section of the document will summarize observations and recommendations driven by the analysis, disclose assumptions, limitations, and issues within the analysis.
- Subsequent sections that discuss background and context for decisions, the data to be used, and the work flow we have defined.
- We will use the following rubric to assess our performance in producing this summary document.

Rubric

General

The assignment solution is due *24 hours prior to the Week 6 Live Session.*

You will only receive credit for this assignment if the assignment is on time, barring any crises.

You will only receive credit for this assignment if you attempt to answer all questions and address all sections with substantive answers, relevant code, and graphics as needed.

Specific

Grades for assignments will follow this rubric:

- **Words:** The text is laid out cleanly, with clear divisions and transitions between sections and subsections. The writing itself is well-organized, free of grammatical and other mechanical errors, divided into complete sentences, logically grouped into paragraphs and sections, and easy to follow from the presumed level of knowledge.
- **Numbers:** All numerical results or summaries are reported to suitable precision, and with appropriate measures of uncertainty attached when applicable.
- **Pictures:** All figures and tables shown are relevant to the argument for ultimate conclusions. Figures and tables are easy to read, with informative captions, titles, axis labels, and legends, and are placed near the relevant pieces of text.
- **Code:** The code is formatted and organized so that it is easy for others to read and understand. It is indented, commented, and uses meaningful names. It only includes computations which are actually needed to answer the analytical questions, and avoids redundancy. Code borrowed from the notes, from books, or from resources found online is explicitly acknowledged and sourced in the comments. Functions or procedures not directly taken from the notes have accompanying tests which check whether the code does what it is supposed to. All code runs, and the `R Markdown` file `knits` to `pdf_document` output, or other output agreed with the instructor.

- **Modeling:** Model specifications are described clearly and in appropriate detail. There are clear explanations of how estimating the model helps to answer the analytical questions, and rationales for all modeling choices. If multiple models are compared, they are all clearly described, along with the rationale for considering multiple models, and the reasons for selecting one model over another, or for using multiple models simultaneously.
- **Inference:** The actual estimation and simulation of model parameters or estimated functions is technically correct. All calculations based on estimates are clearly explained, and also technically correct. All estimates or derived quantities are accompanied with appropriate measures of uncertainty.
- **Conclusions:** The substantive, analytical questions are all answered as precisely as the data and the model allow. The chain of reasoning from estimation results about the model, or derived quantities, to substantive conclusions is both clear and convincing. Contingent answers (for example, "if X, then Y, but if A, then B, else C") are likewise described as warranted by the model and data. If uncertainties in the data and model mean the answers to some questions must be imprecise, this too is reflected in the conclusions.
- **Sources:** All sources used, whether in conversation, print, online, or otherwise, are listed and acknowledged where they used in code, words, pictures, and any other components of the analysis.