a)

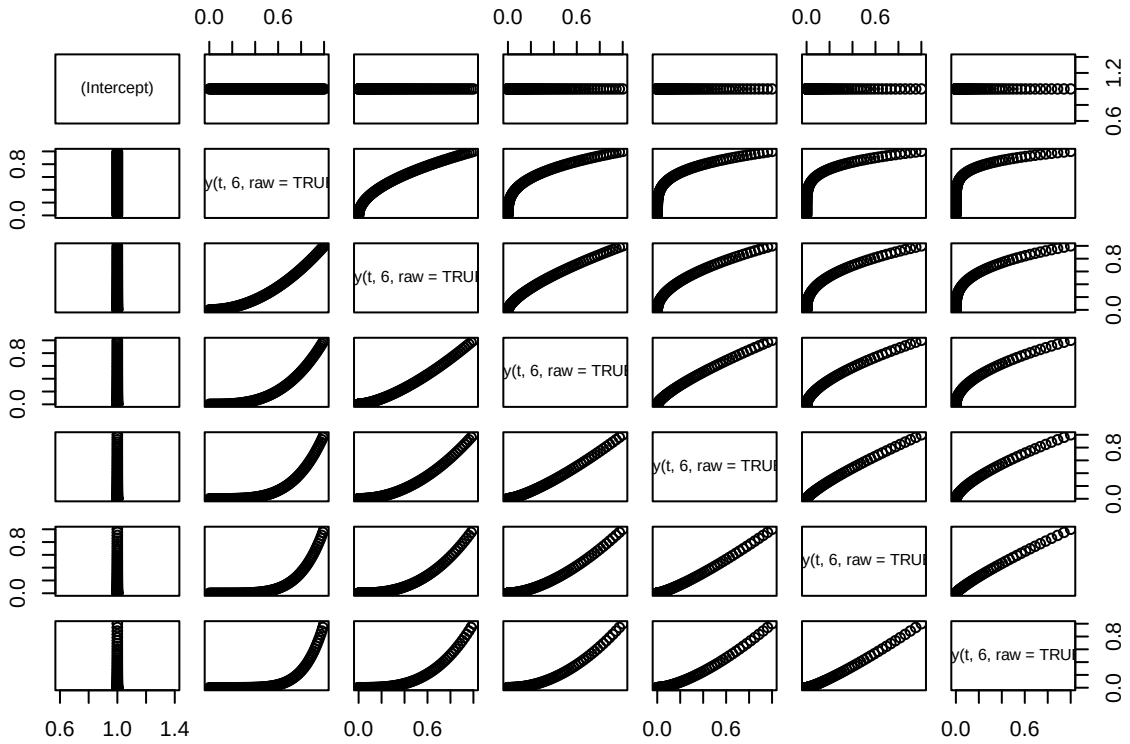
```
# Design matrix
X.raw <- model.matrix(~poly(t, 6, raw = TRUE), data)
# Correlation matrix
X.raw.cor <- cor(X.raw)
```

```
## Warning in cor(X.raw): the standard deviation is zero
```

```
X.raw.cor
```

```
##                         (Intercept) poly(t, 6, raw = TRUE)1
## (Intercept)                       1                      NA
## poly(t, 6, raw = TRUE)1          NA               1.0000000
## poly(t, 6, raw = TRUE)2          NA               0.9676503
## poly(t, 6, raw = TRUE)3          NA               0.9155254
## poly(t, 6, raw = TRUE)4          NA               0.8648570
## poly(t, 6, raw = TRUE)5          NA               0.8194120
## poly(t, 6, raw = TRUE)6          NA               0.7793700
##                         poly(t, 6, raw = TRUE)2 poly(t, 6, raw = TRUE)3
## (Intercept)                                  NA                      NA
## poly(t, 6, raw = TRUE)1               0.9676503               0.9155254
## poly(t, 6, raw = TRUE)2               1.0000000               0.9859506
## poly(t, 6, raw = TRUE)3               0.9859506               1.0000000
## poly(t, 6, raw = TRUE)4               0.9581845               0.9921440
## poly(t, 6, raw = TRUE)5               0.9268554               0.9749697
## poly(t, 6, raw = TRUE)6               0.8956223               0.9539097
##                         poly(t, 6, raw = TRUE)4 poly(t, 6, raw = TRUE)5
## (Intercept)                                  NA                      NA
## poly(t, 6, raw = TRUE)1               0.8648570               0.8194120
## poly(t, 6, raw = TRUE)2               0.9581845               0.9268554
## poly(t, 6, raw = TRUE)3               0.9921440               0.9749697
## poly(t, 6, raw = TRUE)4               1.0000000               0.9949854
## poly(t, 6, raw = TRUE)5               0.9949854               1.0000000
## poly(t, 6, raw = TRUE)6               0.9833317               0.9965231
##                         poly(t, 6, raw = TRUE)6
## (Intercept)                                  NA
## poly(t, 6, raw = TRUE)1               0.7793700
## poly(t, 6, raw = TRUE)2               0.8956223
## poly(t, 6, raw = TRUE)3               0.9539097
## poly(t, 6, raw = TRUE)4               0.9833317
## poly(t, 6, raw = TRUE)5               0.9965231
## poly(t, 6, raw = TRUE)6               1.0000000
```

```
pairs(X.raw)
```



Comments:

- In X.Cor we see all correlation values among the columns are close to 1, so they are close to linear.
- To interpret the scatterplots, index them by rows and columns according to the powers p of t^p in the design matrix for all $t \in [0, 1, 2, 3, 4, 5, 6]$. Note the plots are somewhat symmetrical about the diagonal since reflected plots (t^i, t^j) and (t^j, t^i) represent the same data just with inverted axes
- The observation on the correlation matrix above is supported by the scatterplots as most of them seem to follow linear trends. With more clear linear relationships showing for higher powers, ie. the plot for (t^5, t^6) looks like a straight line. The plots between smaller powers still follow clear patterns but not as linear as for higher powers ie, t^1, t^6 don't seem linear but they do have a non-random relationships.
- The plots between the ones columnd and other colums $(1, t^i)$ show vertical and horizontal lines which means the ones columns is independent (NaN correlations as well)
- Across the diagonal, correlations are one.

b)

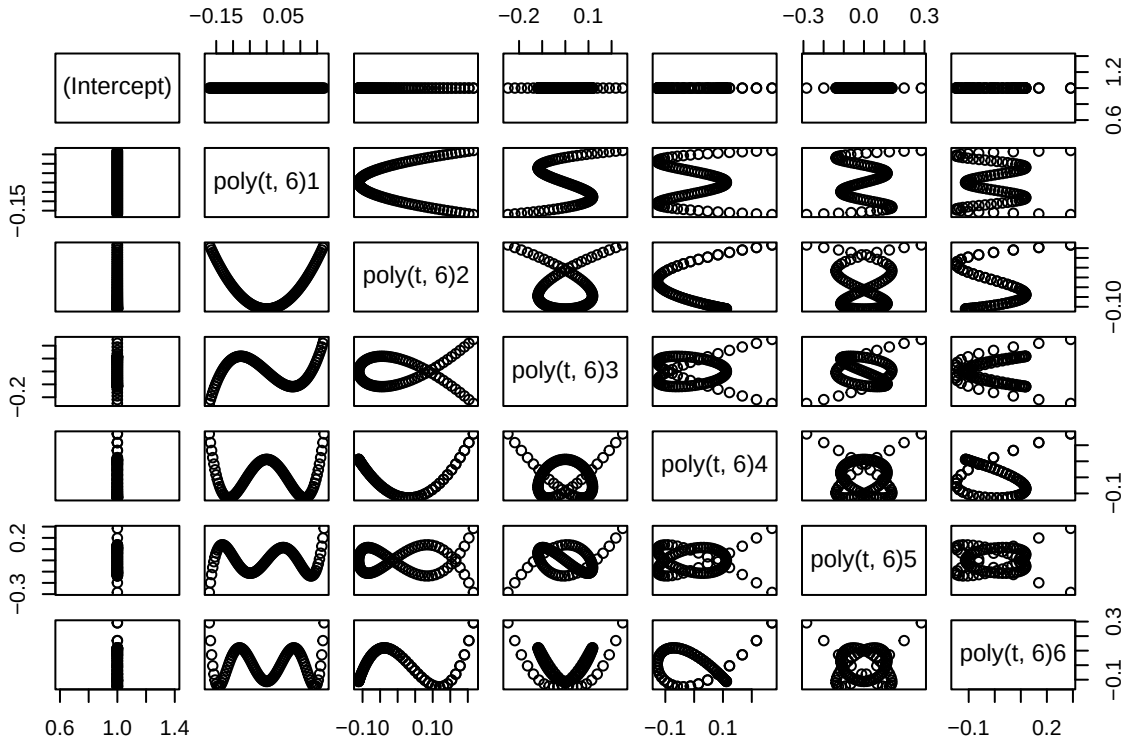
```
# Design matrix
X.ortho <- model.matrix(~poly(t, 6), data)
# Correlation matrix
X.ortho.cor <- cor(X.ortho)
```

```
## Warning in cor(X.ortho): the standard deviation is zero
```

```
X.ortho.cor
```

```
##             (Intercept)   poly(t, 6)1  poly(t, 6)2   poly(t, 6)3   poly(t, 6)4
## (Intercept)           1            NA           NA            NA            NA
## poly(t, 6)1          NA  1.000000e+00 4.576350e-17 -2.296476e-17  2.732867e-17
## poly(t, 6)2          NA  4.576350e-17 1.000000e+00  3.865858e-17  2.812488e-17
## poly(t, 6)3          NA -2.296476e-17 3.865858e-17  1.000000e+00  2.449619e-17
## poly(t, 6)4          NA  2.732867e-17 2.812488e-17  2.449619e-17  1.000000e+00
## poly(t, 6)5          NA -5.187230e-17 5.509102e-18 -7.013433e-18 -2.099964e-17
## poly(t, 6)6          NA  3.463348e-17 1.822815e-17  3.290554e-17  1.275293e-17
##               poly(t, 6)5   poly(t, 6)6
## (Intercept)            NA            NA
## poly(t, 6)1 -5.187230e-17  3.463348e-17
## poly(t, 6)2  5.509102e-18  1.822815e-17
## poly(t, 6)3 -7.013433e-18  3.290554e-17
## poly(t, 6)4 -2.099964e-17  1.275293e-17
## poly(t, 6)5  1.000000e+00 -2.291055e-17
## poly(t, 6)6 -2.291055e-17  1.000000e+00
```

```
pairs(X.ortho)
```



Comments:

- The plots show no linear relationship between any two columnds of X.ortho. There are patterns between

them but some are polynomial and some of them are not even functions.

- The correlation matrix has non-diagonal values in the order of 10^{-17} which supports the observation that no two columns of the design matrix are linearly dependent.

c)

Orthogonal polynomials definitely address concerns about multicolinearity which justifies inferences and also reduces the length of confidence intervals.

On the other hand, regular polynomials have coefficients that can be interpreted in terms of the explanatory variates in meaningful ways (ie, β_i represents an increase in one unit for variable x_i when all else is kept constant), whereas the coefficients of orthogonal polynomials can not be easily interpreted.

For fitting, I would prefer regular polynomials to keep the interpretability of their coefficients. For prediction, we should use orthogonal polynomials.

d)

```
Yt <- data$Yt
t <- data$t
lm_d <- lm(Yt ~ poly(t, 6, raw = TRUE))
summary(lm_d)
```

```
##
## Call:
## lm(formula = Yt ~ poly(t, 6, raw = TRUE))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.239422 -0.048626 -0.003498  0.054021  0.219662
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                9.16410    0.05419 169.117  < 2e-16 ***
## poly(t, 6, raw = TRUE)1    9.94042    1.53971   6.456 4.67e-09 ***
## poly(t, 6, raw = TRUE)2  -42.41134   13.67064  -3.102  0.00254 **
## poly(t, 6, raw = TRUE)3   77.67696   51.82887   1.499  0.13730
## poly(t, 6, raw = TRUE)4  -49.42553   94.68793  -0.522  0.60291
## poly(t, 6, raw = TRUE)5  -15.60068   82.35631  -0.189  0.85016
## poly(t, 6, raw = TRUE)6   21.47312   27.35505   0.785  0.43444
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08721 on 94 degrees of freedom
## Multiple R-squared:  0.8841, Adjusted R-squared:  0.8767
## F-statistic: 119.5 on 6 and 94 DF,  p-value: < 2.2e-16
```

Comment:

- The $p-$values for coefficients of degrees greated than 3 are large and thus insignificant. So there is no evidence here to reject the null hypotheses: $H_{0,j} : \beta_j = 0, \forall j \in [3, 4, 5, 6]$.
- The $p-$value of the F statistic is very small and thus highly significant. It says there is strong evidence to reject the null hypothesis that our model is better than the intercept-only model.
- Both of the above observations are contradictory and they suggest there is multicolinearity between some of the explanatory variables $\{1, t, t^2, t^3, t^4, t^5, t^6\}$ just as we suspected from the scatterplots in part (a).

e)

```
lm_e <- lm(Yt ~ poly(t, 6, raw = FALSE))
summary(lm_e)
```

```
##
## Call:
## lm(formula = Yt ~ poly(t, 6, raw = FALSE))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.239422 -0.048626 -0.003498  0.054021  0.219662
##
## Coefficients:
##                            Estimate Std. Error  t value Pr(>|t|)
## (Intercept)                9.998733   0.008677 1152.291  < 2e-16 ***
## poly(t, 6, raw = FALSE)1   1.733601   0.087205   19.880  < 2e-16 ***
## poly(t, 6, raw = FALSE)2  -0.102555   0.087205   -1.176    0.243
## poly(t, 6, raw = FALSE)3   1.433507   0.087205   16.438  < 2e-16 ***
## poly(t, 6, raw = FALSE)4  -0.022033   0.087205   -0.253    0.801
## poly(t, 6, raw = FALSE)5   0.615297   0.087205    7.056 2.89e-10 ***
## poly(t, 6, raw = FALSE)6   0.068454   0.087205    0.785    0.434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08721 on 94 degrees of freedom
## Multiple R-squared:  0.8841, Adjusted R-squared:  0.8767
## F-statistic: 119.5 on 6 and 94 DF,  p-value: < 2.2e-16
```

Comment:

- The $p-$values for coefficients of degrees $\{2, 4, 6\}$ are large and thus insignificant. So there is no evidence here to reject the null hypotheses: $H_{0,j} : \beta_j = 0, \forall j \in [2, 4, 6]$. A next step in trying to find a good model would be to drop these powers.
- The $p-$value of the F statistic is very small and thus highly significant. It says there is strong evidence to reject the null hypothesis that our model is better than the intercept-only model.

f)

```
res_d <- lm_d$residuals
res_e <- lm_e$residuals # orthogonal
lm_res<-lm(res_d ~res_e)
lm_res$coefficients
```

```
##   (Intercept)         res_e
## -4.883151e-18  1.000000e+00
```

```
head(lm_res$residuals)
```

```
##             1             2             3             4             5
##  1.268080e-14 -2.205226e-14  1.051645e-14  1.432843e-15 -7.925527e-16
##             6
##  3.453844e-16
```

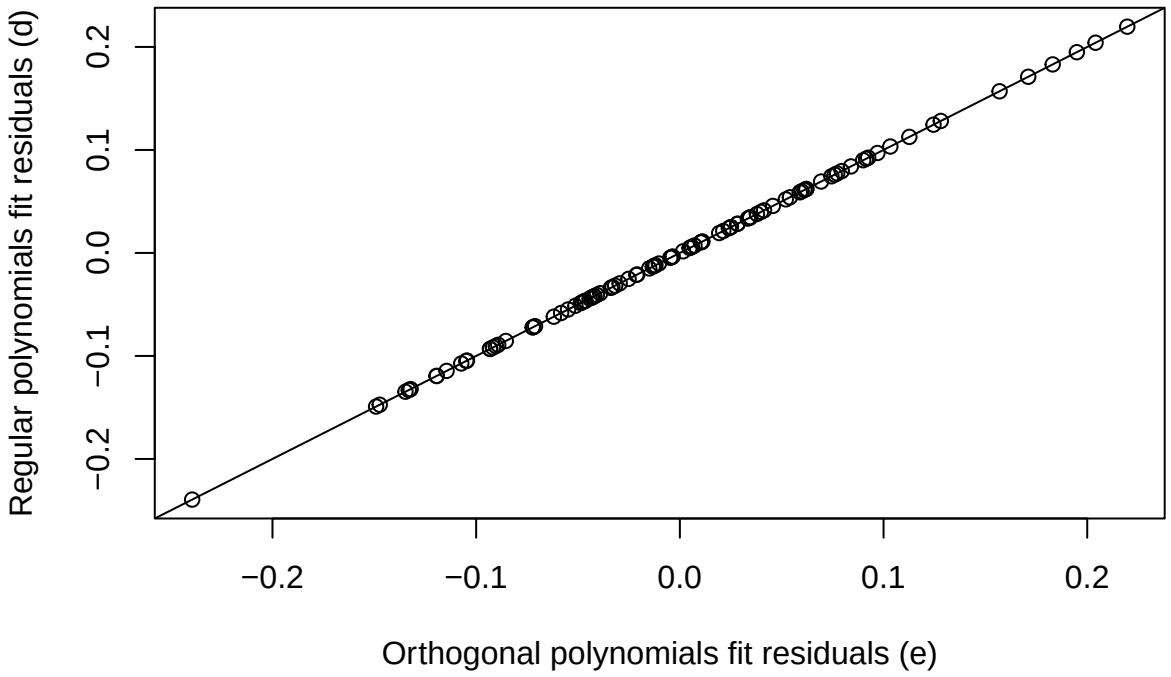
```
fit_e <-lm_e$fitted.values # orthogonal
fit_d <-lm_d$fitted.values
lm_fit<-lm(fit_d ~fit_e)
lm_fit$coefficients
```

```
##   (Intercept)         fit_e
## -1.555436e-14  1.000000e+00
```

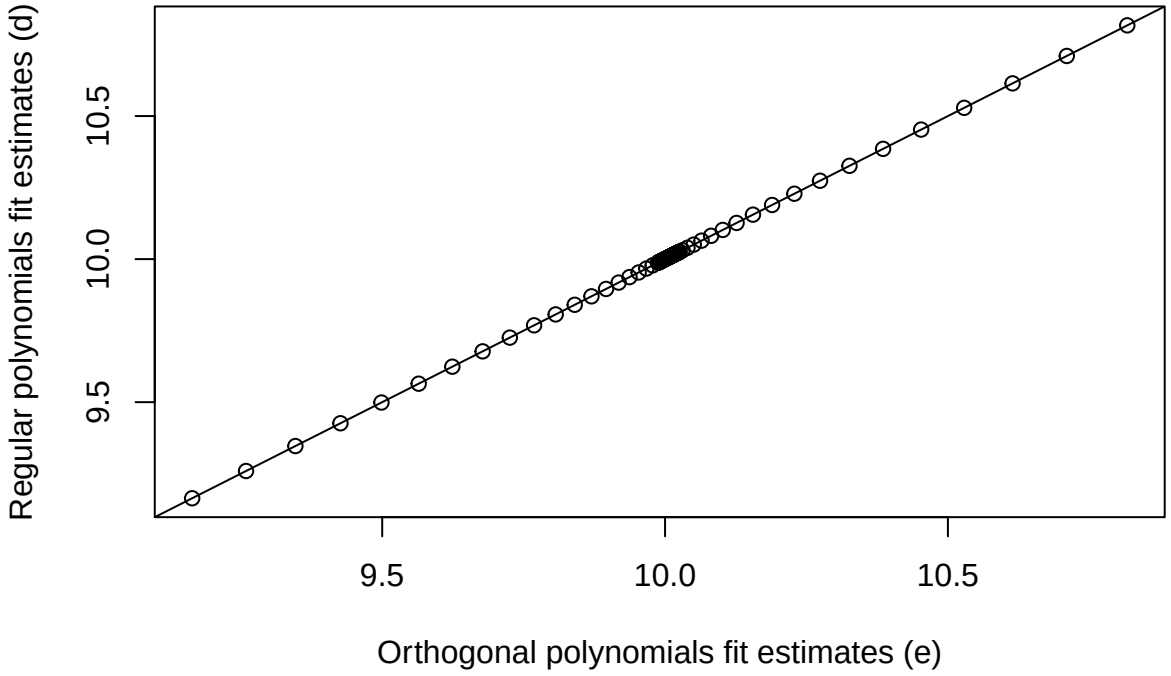
```
head(lm_fit$residuals)
```

```
##             1             2             3             4             5
## -1.411859e-14  2.544206e-14 -9.685812e-15 -9.983593e-16  6.902515e-16
##             6
##  5.871003e-16
```

```
plot(res_e, res_d,
     xlab = "Orthogonal polynomials fit residuals (e)",
     ylab = "Regular polynomials fit residuals (d)")
abline(lm_res)
```




```
plot(fit_e, fit_d,
     xlab = "Orthogonal polynomials fit estimates (e)",
     ylab = "Regular polynomials fit estimates (d)")
abline(lm_fit)
```



Observations:

- The residuals for both models follow a near-perfect linear relationship. The slope coefficient of linear model for residuals (lm_res) is virtually 1, there is however a very small intercept between the two, suggesting that residuals for the orthogonal polynomials model in part e) are slighly larger than those for the regular polynomials model in part e. The relationship is not perfectly linear as the residuals of lm_res are not zero.

- The fitted values for both models follow a near-perfect linear relationship. The slope coefficient of linear model for fits (lm_fit) is virtually 1, there is however a very small intercept between the two, suggesting that the fitted values for the orthogonal polynomials model in part e) are slighly larger than those for the regular polynomials model in part e. The relationship is not perfectly linear as the residuals of lm_fit are not zero.