HW3

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(readr)
library(ggplot2)
library(splines)
```

Question 1

Replicate the figure 5.6

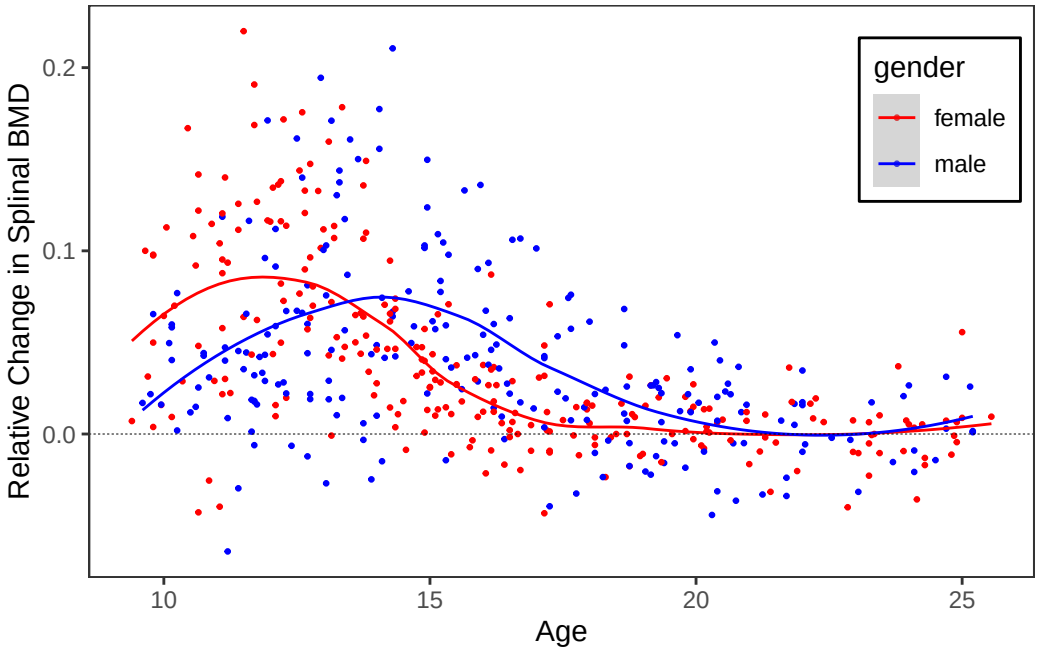
```
q1<-read_table(
"https://hastie.su.domains/ElemStatLearn/datasets/bone.data")%>%as.data.frame()
```

```
ggplot(data=q1)+
  geom_point(aes(x=age,y=spnbmd,color=gender),size=0.5)+
  geom_smooth(aes(x=age,y=spnbmd,color=gender),level=0,size=0.5)+
```

```
  scale_color_manual(values= c("male"="blue","female"="red"))+
  labs(x="Age",y="Relative Change in Splinal BMD")+
  theme_bw()+
  geom_hline(yintercept=0,linetype=2,size=0.1)+
  theme(
    panel.grid = element_blank(),
#    legend.background = element_blank(),
    legend.position = c(.9, .8),
   # legend.box.background = element_rect(color="black", size=0.05),
   # legend.box.margin = margin(6, 6, 6, 6),
    legend.background = element_rect(fill = "white", color = "black"))
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Question 2

```
df<-read_csv("http://www-stat.stanford.edu/~tibs/ElemStatLearn/datasets/SAheart.data")

View(df)
```

Spline basis

B-spline basis

```
b_spline<-bs(df$tobacco,df=6, intercept = FALSE)
```

Natural Spline Basis

```
n_spline<- ns(df$tobacco, df = 4, intercept = FALSE)
```

Truncated polynomial spline bases

```
t_spline<- bs(df$tobacco,df=6, intercept = TRUE)
```

Logistic Regressions

```
b_logistic<-glm(df$chd ~ b_spline,family = binomial(link = "logit"))
b_logistic
```

```
Call:  glm(formula = df$chd ~ b_spline, family = binomial(link = "logit"))

Coefficients:
(Intercept)    b_spline1    b_spline2    b_spline3    b_spline4    b_spline5
    -1.7800      -0.6808       1.7603       1.0070       3.0172       1.5492
  b_spline6
     5.6237

Degrees of Freedom: 461 Total (i.e. Null);  455 Residual
Null Deviance:      596.1
Residual Deviance: 538.4    AIC: 552.4
```

```
n_logistic<-glm(df$chd ~ n_spline,family = binomial(link = "logit"))
n_logistic
```

```
Call:  glm(formula = df$chd ~ n_spline, family = binomial(link = "logit"))

Coefficients:
(Intercept)    n_spline1    n_spline2    n_spline3    n_spline4
     -1.896        1.323        1.813        4.558        3.328

Degrees of Freedom: 461 Total (i.e. Null);  457 Residual
Null Deviance:      596.1
Residual Deviance: 539.8    AIC: 549.8
```

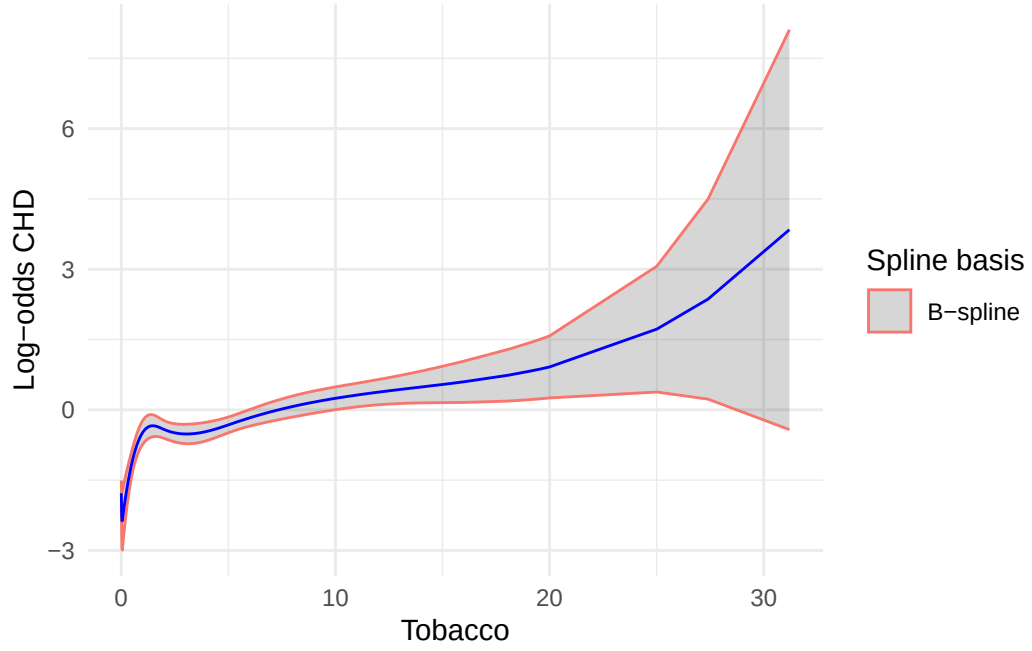
```
t_logistic<-glm(df$chd ~ t_spline,family = binomial(link = "logit"))
t_logistic
```

```
Call:  glm(formula = df$chd ~ t_spline, family = binomial(link = "logit"))

Coefficients:
(Intercept)    t_spline1    t_spline2    t_spline3    t_spline4    t_spline5
      2.610       -4.487       -3.749       -3.001       -2.631       -1.115
  t_spline6
         NA

Degrees of Freedom: 461 Total (i.e. Null);  456 Residual
Null Deviance:      596.1
Residual Deviance: 541.1    AIC: 553.1
```

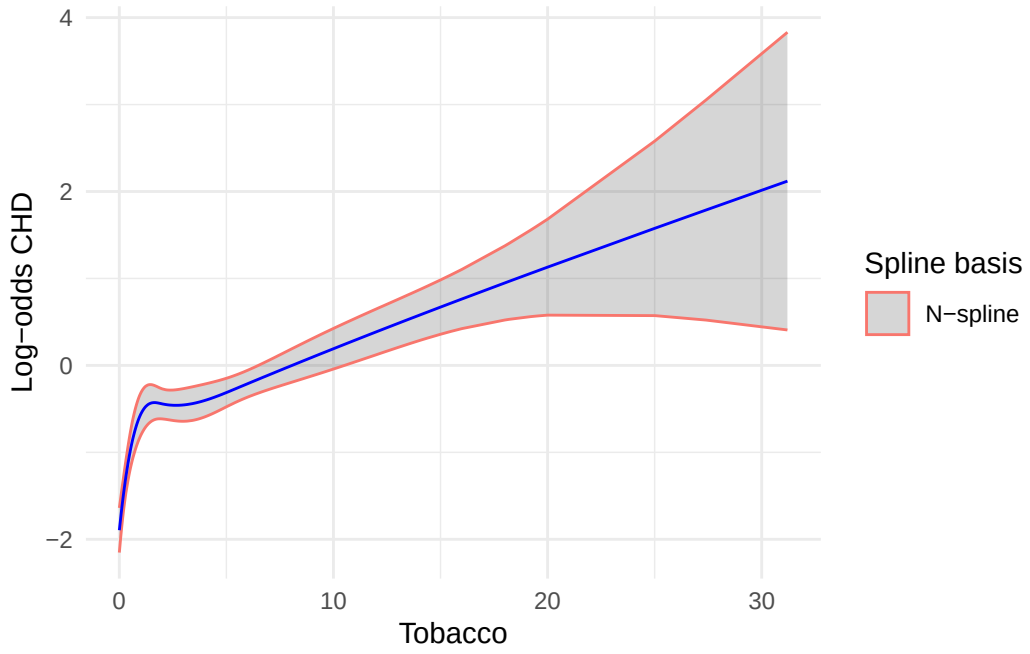
Predicted Model and Variance

```
fit_bspline <- glm(chd ~ b_spline, data = df, family = binomial())
pred_bspline <- predict(fit_bspline, type = "link")
var_bspline <- predict(fit_bspline, type = "link", se.fit = TRUE)$se.fit^2
plot_data_bspline <- data.frame(tobacco = df$tobacco, pred = pred_bspline,
                                upper = pred_bspline + sqrt(var_bspline),
                                lower = pred_bspline - sqrt(var_bspline), spline = "B-spli
```

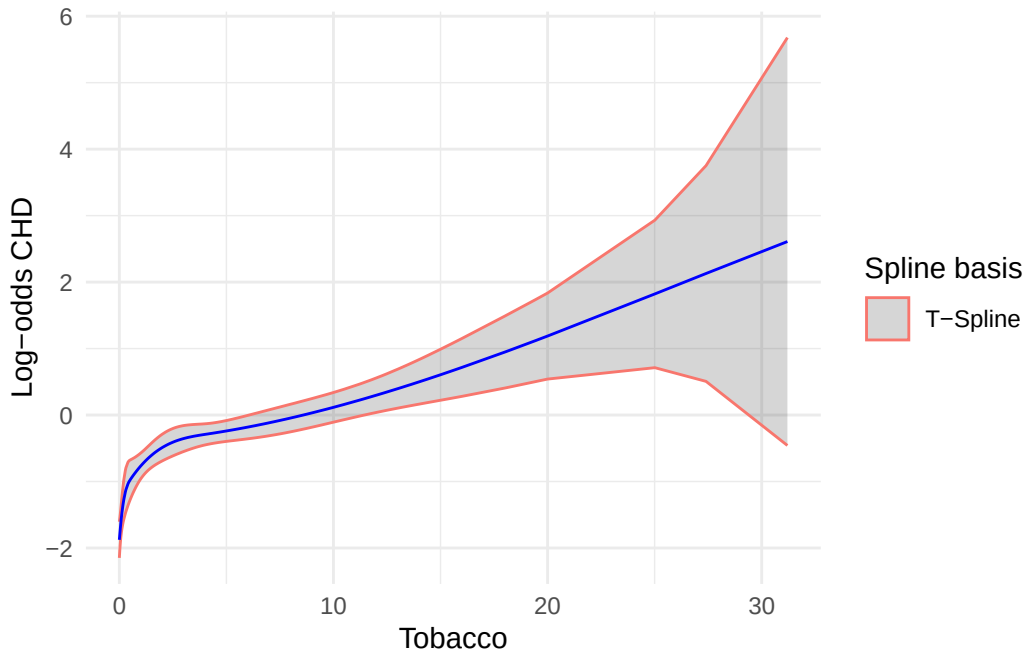
```r
ggplot(plot_data_bspline, aes(x = tobacco, y = pred, colour = spline)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  geom_line(color="blue")+
  labs(x = "Tobacco", y = "Log-odds CHD", colour = "Spline basis") +
  theme_minimal()
```




```r
fit_nspline <- glm(chd ~ n_spline, data = df, family = binomial())
pred_nspline <- predict(fit_nspline, type = "link")
var_nspline <- predict(fit_nspline, type = "link", se.fit = TRUE)$se.fit^2
plot_data_nspline <- data.frame(tobacco = df$tobacco, pred = pred_nspline,
                                upper = pred_nspline + sqrt(var_nspline),
                                lower = pred_nspline - sqrt(var_nspline), spline = "N-spli

ggplot(plot_data_nspline, aes(x = tobacco, y = pred, colour = spline)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  geom_line(color="blue")+
  labs(x = "Tobacco", y = "Log-odds CHD", colour = "Spline basis") +
  theme_minimal()
```



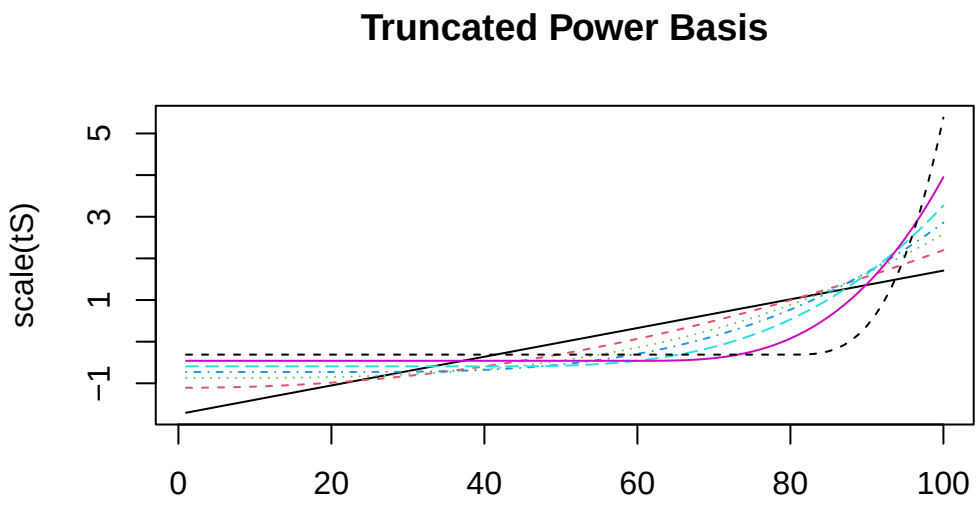

```
fit_tspline <- glm(chd ~ t_spline, data = df, family = binomial())
pred_tspline <- predict(fit_tspline, type = "link")
var_tspline <- predict(fit_tspline, type = "link", se.fit = TRUE)$se.fit^2
plot_data_tspline <- data.frame(tobacco = df$tobacco, pred = pred_tspline,
                                upper = pred_tspline + sqrt(var_tspline),
                                lower = pred_tspline - sqrt(var_tspline), spline = "T-Spli

ggplot(plot_data_tspline, aes(x = tobacco, y = pred, colour = spline)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2) +
  geom_line(color="blue")+
  labs(x = "Tobacco", y = "Log-odds CHD", colour = "Spline basis") +
  theme_minimal()
```



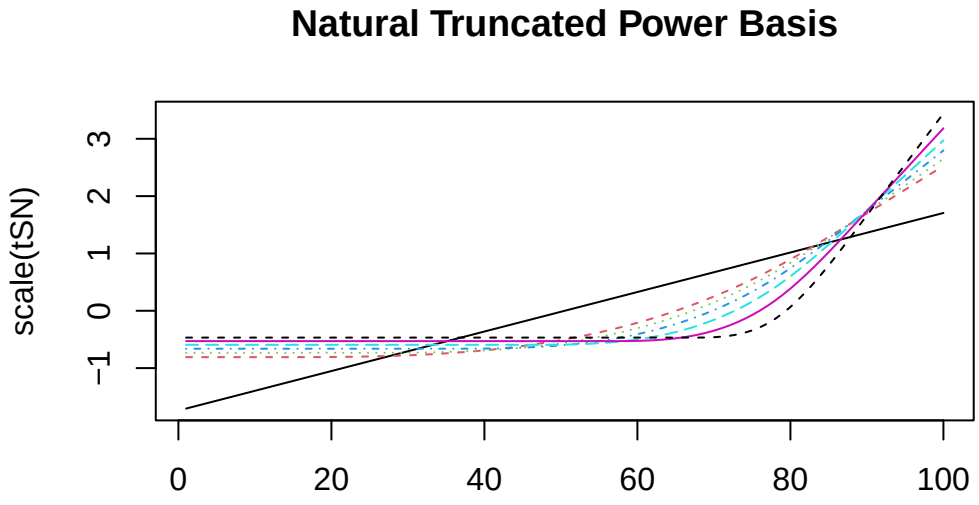

```
truncpolyspline <- function(x, df, natural = FALSE) {
  if (!require("Matrix")) stop("need Matrix package")

  if (natural) {
    knots <- quantile(x, seq(0, 1, length = df + 3))
    xi_K = knots[df+2]; xi_K_1 = knots[df+1]
    d_K_1 = ((x>=xi_K_1)*(x-xi_K_1)^3 - (x>=xi_K)*(x-xi_K)^3) / (xi_K-xi_K_1)
    S <- sapply(knots[2:(df)], function(k) ((x>=k)*(x-k)^3 - (x>=xi_K)*(x-xi_K)^3) / (xi_K
    S <- as(S, "CsparseMatrix");
    S <- cbind(x, S)
  } else {
    knots <- quantile(x, seq(0, 1, length = df - 1))
    trunc_fun <- function(k) (x>=k)*(x-k)^3
    S <- sapply(knots[1:(df-2)], function(k) (x>=k)*(x-k)^3)
    S <- as(S, "CsparseMatrix");
    S <- cbind(x, x^2, S)
  }
  return(S)
}

xvec <- seq(0, 5, length = 100)
tS <- truncpolyspline(xvec, df = 7, natural = FALSE)
```

```
Loading required package: Matrix
```

```
tSN <- truncpolyspline(xvec, df = 7, natural = TRUE)

matplot(scale(tS), type = "l",
main = "Truncated Power Basis")
```




```
matplot(scale(tSN), type = "l",
main = "Natural Truncated Power Basis")
```



Please accept my sincere apologies. I wil work on Question 4 as soon as possible.