Quantile regression models for abalone shells

Claudia Czado, Technical University Munich

07 August, 2018

1.The abalone data set

Source and data description

The `abalone` dataset is available from the University of California Irvine (UCI) machine learning repository. Metadata can be obtained from http://archive.ics.uci.edu/ml/datasets/Abalone

It is also available in the library PivotalR.

- Sex / nominal / – / M, F, and I (infant)
- Length / continuous / mm / Longest shell measurement
- Diameter / continuous / mm / perpendicular to length
- Height / continuous / mm / with meat in shell
- Whole weight / continuous / grams / whole abalone
- Shucked weight / continuous / grams / weight of meat
- Viscera weight / continuous / grams / gut weight (after bleeding)
- Shell weight / continuous / grams / after being dried
- Rings / integer / – / +1.5 gives the age in years

2. Setup

Load packages

```
library(VineCopula)
library(PivotalR)
library(rafalib)
#library(kdevine)
library(vinereg)
library(ggplot2)
```

Load data and name columns

The dataset contains 10 variables and 4177 observations. Most of the variables are numeric. The only exception is the `sex` variable. The `rings` variable is slightly different from the other numeric variables because it assumes discrete, integer values.

```
data("abalone")
abalone.cols = c( "sex", "len", "dia", "h", "whole",
                  "shuck", "vis", "shell", "rings")
abalone1=abalone[,-1]
colnames(abalone1)=abalone.cols
sex1=abalone1[,1]
sex.num=rep(0,4177)
sex.num[sex1=="M"]=1
sex.num[sex1=="F"]=0
sex.num[sex1=="I"]=2
abalone1[,1]=sex.num
```

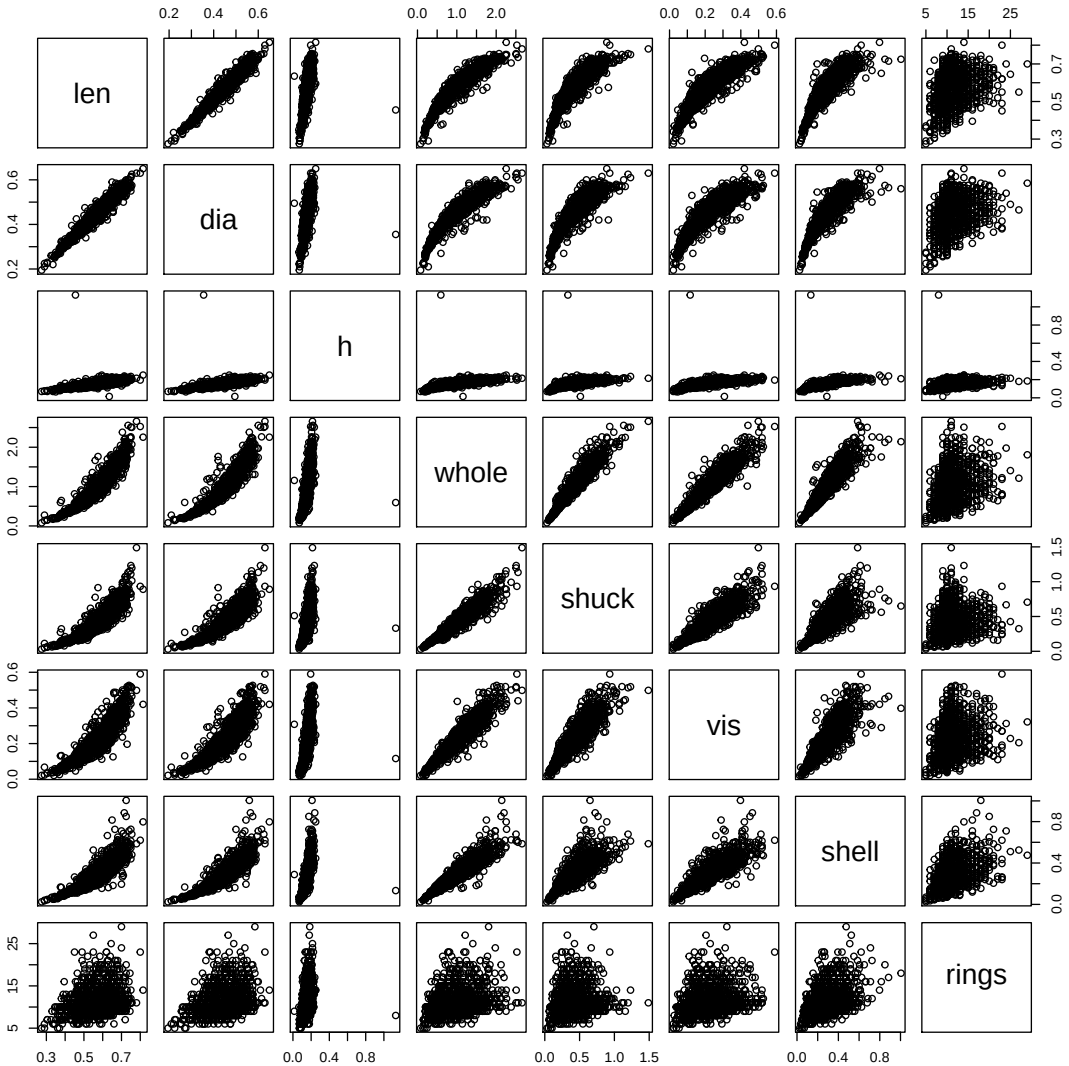
Create datasets for male, female and junvenile separately

```
attach(abalone)
abalone.f<-abalone1[sex=="F",-1]
abalone.m<-abalone1[sex=="M",-1]
abalone.i<-abalone1[sex=="I",-1]
detach(abalone)
```

3. Female abalone data set

Raw data of female abalone shells

```
pairs(abalone.f)
```



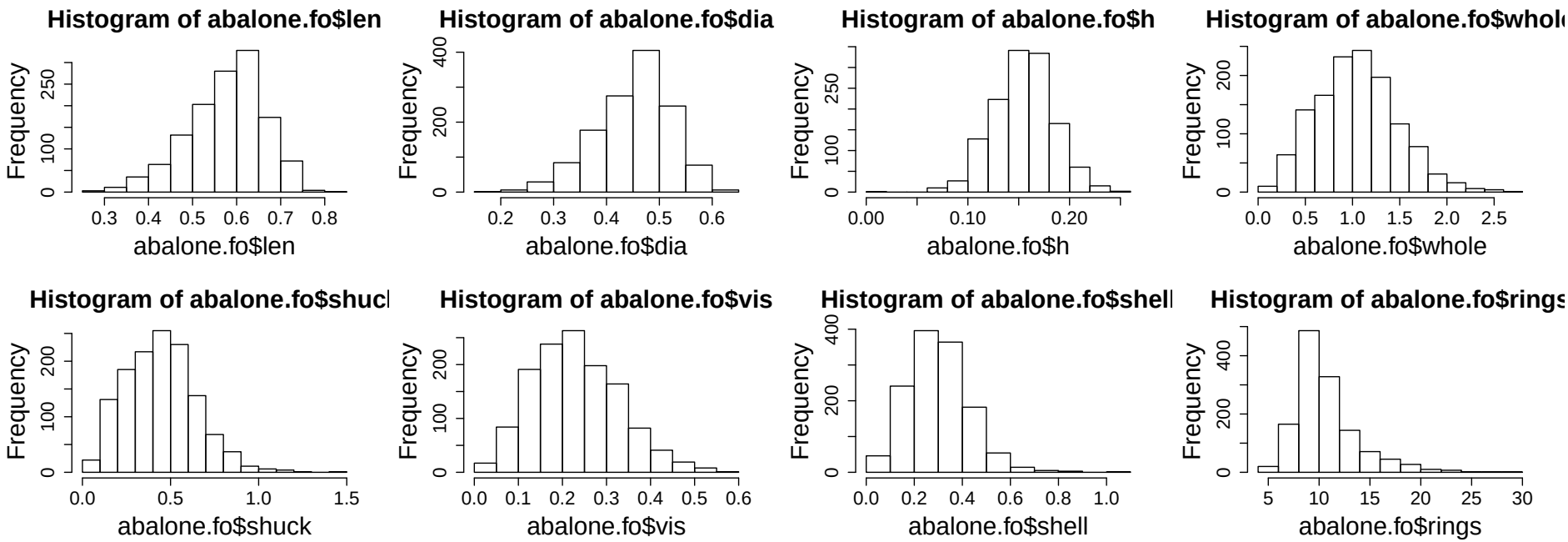
Remove outlier in height

```
temp<-max(abalone.f$h)
ind<-(1:length(abalone.f$h))[abalone.f$h==temp]
abalone.fo<-abalone.f[-ind,]
summary(abalone.fo)
```

```
##       len               dia                h               whole
##  Min.   :0.2750   Min.   :0.1950   Min.   :0.0150   Min.   :0.0800
##  1st Qu.:0.5250   1st Qu.:0.4100   1st Qu.:0.1400   1st Qu.:0.7315
##  Median :0.5900   Median :0.4650   Median :0.1600   Median :1.0385
##  Mean   :0.5792   Mean   :0.4548   Mean   :0.1573   Mean   :1.0469
##  3rd Qu.:0.6400   3rd Qu.:0.5050   3rd Qu.:0.1750   3rd Qu.:1.3204
##  Max.   :0.8150   Max.   :0.6500   Max.   :0.2500   Max.   :2.6570
##      shuck              vis              shell             rings
##  Min.   :0.0310   Min.   :0.0210   Min.   :0.0250   Min.   : 5.00
##  1st Qu.:0.2950   1st Qu.:0.1590   1st Qu.:0.2142   1st Qu.: 9.00
##  Median :0.4405   Median :0.2240   Median :0.2950   Median :10.00
##  Mean   :0.4463   Mean   :0.2308   Mean   :0.3021   Mean   :11.13
##  3rd Qu.:0.5734   3rd Qu.:0.2974   3rd Qu.:0.3750   3rd Qu.:12.00
##  Max.   :1.4880   Max.   :0.5900   Max.   :1.0050   Max.   :29.00
```

Marginal histograms

```
bigpar(2,4)
hist(abalone.fo$len)
hist(abalone.fo$dia)
hist(abalone.fo$h)
hist(abalone.fo$whole)
hist(abalone.fo$shuck)
hist(abalone.fo$vis)
hist(abalone.fo$shell)
hist(abalone.fo$rings)
```



Check for discreteness

```
out.unique<-c(length(unique(abalone.fo$len)),
length(unique(abalone.fo$dia)),
length(unique(abalone.fo$h)),
length(unique(abalone.fo$whole)),
length(unique(abalone.fo$shuck)),
length(unique(abalone.fo$vis)),
length(unique(abalone.fo$shell)),
length(unique(abalone.fo$rings)))
names(out.unique)<-c("len","dia","h","whole","shuck","vis","shell","rings")
out.unique
```

```
##   len   dia     h whole shuck   vis shell rings
##    91    81    38  1072   854   627   482    23
```

The variables h and rings are very discrete, therefore we also consider models where h and rings are considered as ordered

Include ordered versions of rings and h to the data set

```
rings.ord<-ordered(abalone.fo$rings,levels=sort(unique(abalone.fo$rings)))
h.ord<-ordered(abalone.fo$h,levels=sort(unique(abalone.fo$h)))
abalone.fo1<-data.frame(abalone.fo,rings.ord,h.ord)
summary(abalone.fo1[,c("rings.ord","h.ord")])
```

```
##    rings.ord       h.ord
##  10     :248   0.15   :113
##  9      :238   0.175  : 96
##  11     :200   0.16   : 91
##  12     :128   0.165  : 90
##  8      :121   0.17   : 86
##  13     : 88   0.155  : 82
##  (Other):283   (Other):748
```

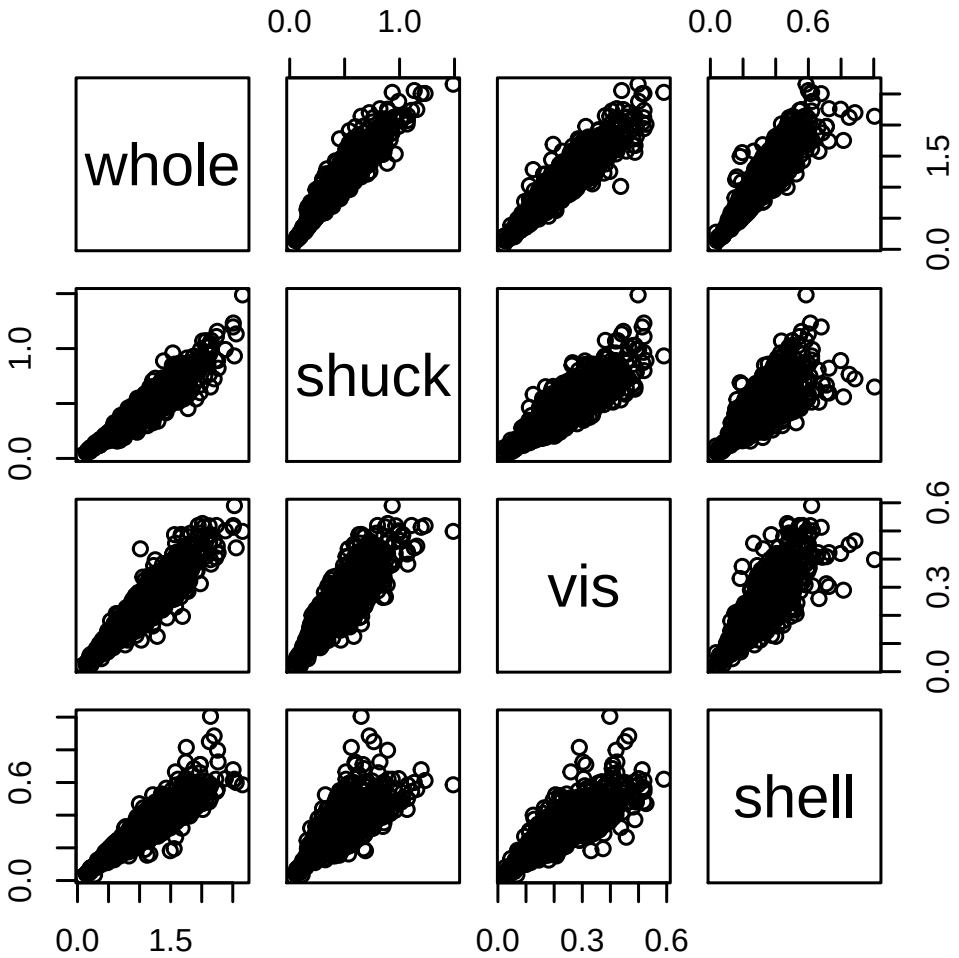
```
rm(rings.ord)
rm(h.ord)
```

Use emprical cdfs to transform to copula data

```
n<-nrow(abalone.fo)
fak<-n/(n+1)
temp<-ecdf(abalone.fo$len)
u1<-temp(abalone.fo$len)*fak
temp<-ecdf(abalone.fo$dia)
u2<-temp(abalone.fo$dia)*fak
temp<-ecdf(abalone.fo$h)
u3<-temp(abalone.fo$h)*fak
temp<-ecdf(abalone.fo$whole)
u4<-temp(abalone.fo$whole)*fak
temp<-ecdf(abalone.fo$shuck)
u5<-temp(abalone.fo$shuck)*fak
temp<-ecdf(abalone.fo$vis)
u6<-temp(abalone.fo$vis)*fak
temp<-ecdf(abalone.fo$shell)
u7<-temp(abalone.fo$shell)*fak
temp<-ecdf(abalone.fo$rings)
u8<-temp(abalone.fo$rings)*fak
udata.fo<-cbind(u1,u2,u3,u4,u5,u6,u7,u8)
colnames(udata.fo)<-c("len","dia","h","whole","shuck","vis","shell","rings")
udata.fo<-as.copuladata(udata.fo)
```

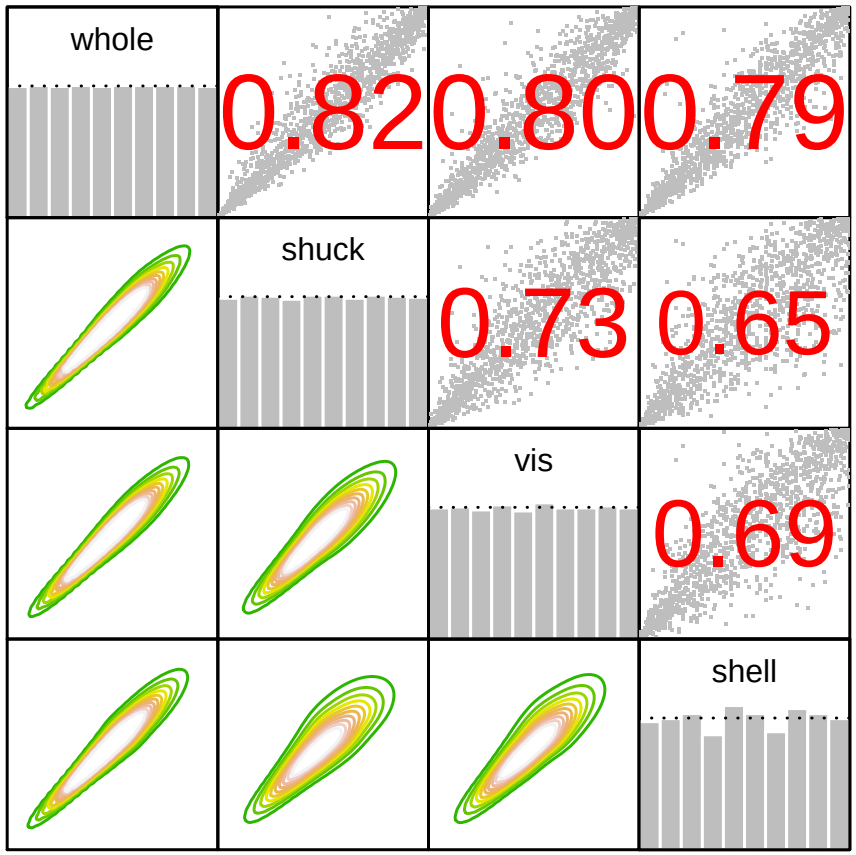
EDA for whole, shuck, vis and shell (x-level)

```
abalone.f4<-abalone.fo[,4:7]
pairs(abalone.f4)
```



Empirical normalized contour plots (z-level)

```
udata.f4<-udata.fo[,4:7]
pairs(udata.f4)
```



Pairwise Kendalls'tau

```
round(cor(udata.f4,method="kendall"),digits=2)
```

```
##        whole shuck  vis shell
## whole   1.00  0.82 0.80  0.79
## shuck   0.82  1.00 0.73  0.65
## vis     0.80  0.73 1.00  0.69
## shell   0.79  0.65 0.69  1.00
```

4. D-vine quantile regression of whole against shuck, vis, shell

Parametric D-vine quantile regression of whole against shuck, vis and shell with forward selection of variables

```
whole_dvqr <- vinereg(whole ~ shuck+vis+shell, dat=abalone.f4)
summary(whole_dvqr)
```

```
##     var      edf       cll      caic       cbic       p_value
## 1 whole 11.73151 -715.2116  1453.886  1514.5936           NA
## 2 shuck  1.00000 1807.4510 -3612.902 -3607.7272  0.000000e+00
## 3 shell  2.00000  638.0342 -1272.068 -1261.7189 8.040271e-278
## 4   vis  4.00000  132.2680  -256.536  -235.8371  4.802544e-56
```

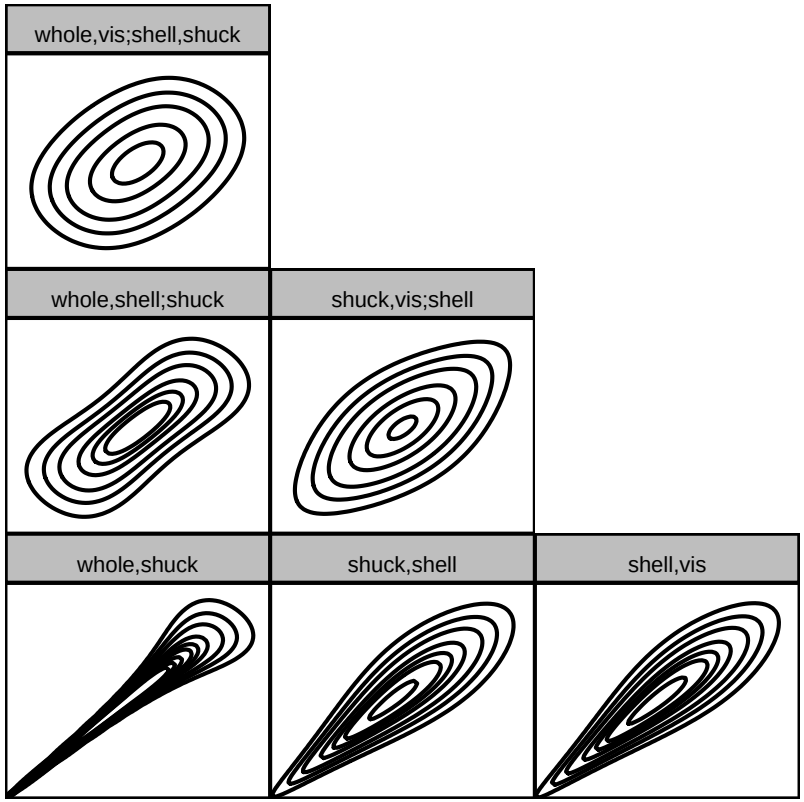
Fitted vine

```
summary(whole_dvqr$vine)
```

```
## # A data.frame: 6 x 9
##   tree edge conditioned conditioning family rotation  parameters df  tau
## 1    1    1        1, 2                joe      180          16  1 0.88
## 2    1    2        2, 4              gumbel      180         2.9  1 0.66
## 3    1    3        4, 3              gumbel      180         3.1  1 0.68
## 4    2    1        1, 4           2  frank        0         6.6  1 0.54
## 5    2    2        2, 3           4      t        0 0.61, 11.09  2 0.42
## 6    3    1        1, 3        4, 2  frank        0         2.3  1 0.25
```

Fitted normalized contour plots

```
contour(whole_dvqr$vine)
```



Fitted values based on parametric D-vine quantile regression

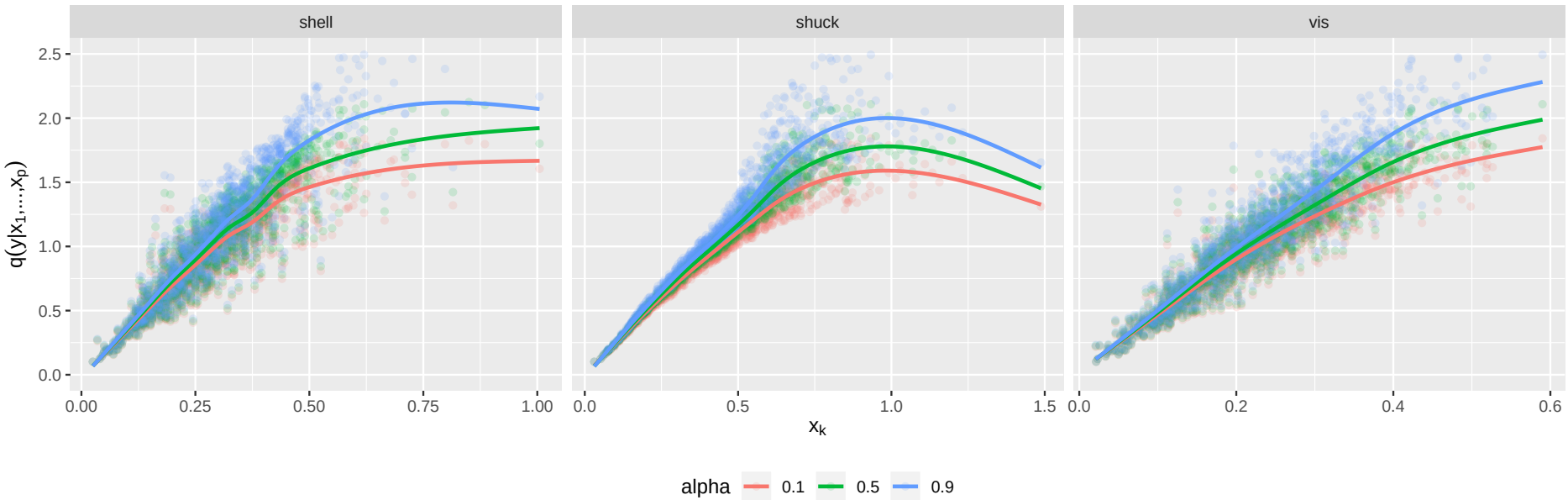
```
alpha_vec <- c(0.1, 0.5, 0.9)
pred_whole_dvqr <- fitted(whole_dvqr,alpha = alpha_vec)
```

Marginal effect plotting function

```
plot_marginal_effects <- function(covs, preds) {
    cbind(covs, preds) %>%
        tidyr::gather(alpha, prediction, -seq_len(NCOL(covs))) %>%
        dplyr::mutate(prediction = as.numeric(prediction)) %>%
        tidyr::gather(variable, value, -(alpha:prediction)) %>%
        ggplot(aes(value, prediction, color = alpha)) +
            geom_point(alpha = 0.15) +
            geom_smooth(se = FALSE) +
            facet_wrap(~ variable, scale = "free_x") +
            ylab(quote(q(y* "|" * x[1] * ",...," * x[p]))) +
            xlab(quote(x[k])) +
            theme(legend.position = "bottom")+
            ylim(0, 2.5)
}
```

Marginal effects of shuck, vis and shell on whole

```
plot_marginal_effects(abalone.f4[, c("shuck","vis","shell")],
                      pred_whole_dvqr)
```



5. Non parametric D-vine quantile regression of whole against shuck, vis and shell

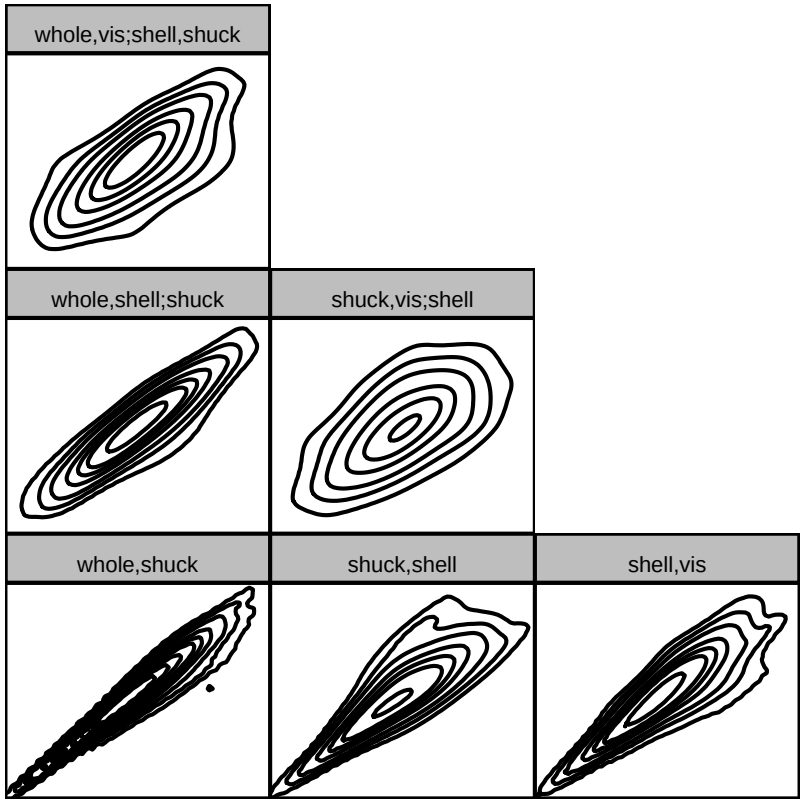
Non parametric D-vine quantile regression of whole against shuck, vis and shell with forward selection

```
whole_dvqr_np <- vinereg(whole ~ shuck+vis+shell,
    dat=abalone.f4, family_set = "nonparametric")
summary(whole_dvqr_np)
```

```
##     var       edf       cll       caic          cbic      p_value
## 1 whole  11.73151 -715.2116  1453.8862  1514.5935553           NA
## 2 shuck  47.51917 1787.8678 -3480.6972 -3234.7986199 0.000000e+00
## 3 shell  70.28715  943.8365 -1747.0987 -1383.3821133 0.000000e+00
## 4   vis 104.94725  375.9959  -542.0974     0.9756906 2.101021e-98
```

Fitted non parametric D-vine quantile values

```
contour(whole_dvqr_np$vine)
```

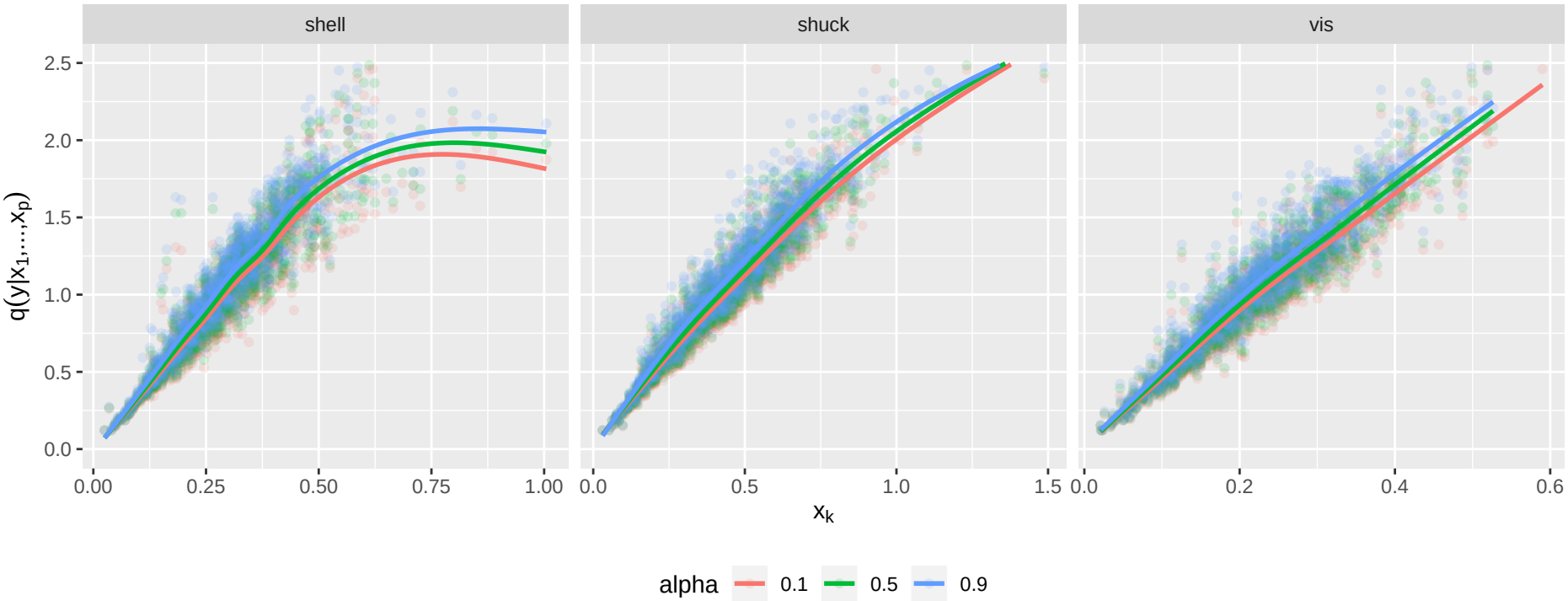


Fitted values based on non parametric D-vine quantile regression

```
alpha_vec <- c(0.1, 0.5, 0.9)
pred_whole_dvqr_np <- fitted(whole_dvqr_np,alpha = alpha_vec)
```

Marginal effects of shuck, vis and shell on whole

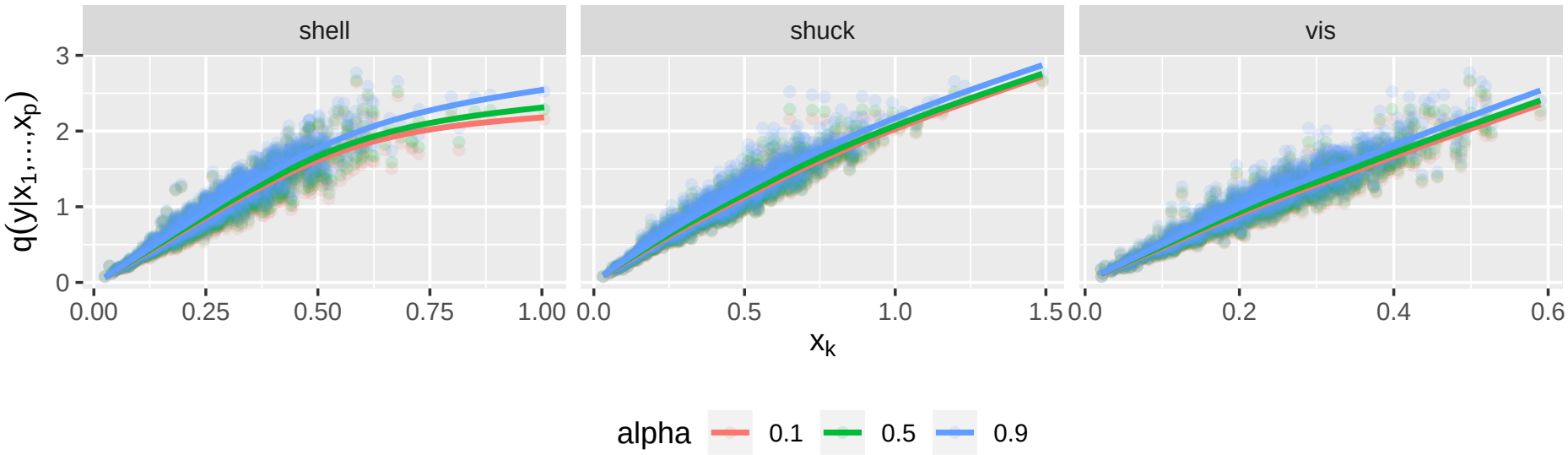
```
plot_marginal_effects(abalone.f4[, c("shuck","vis","shell")], pred_whole_dvqr_np)
```



6. Linear quantile regression for whole against shuck, vis and shell

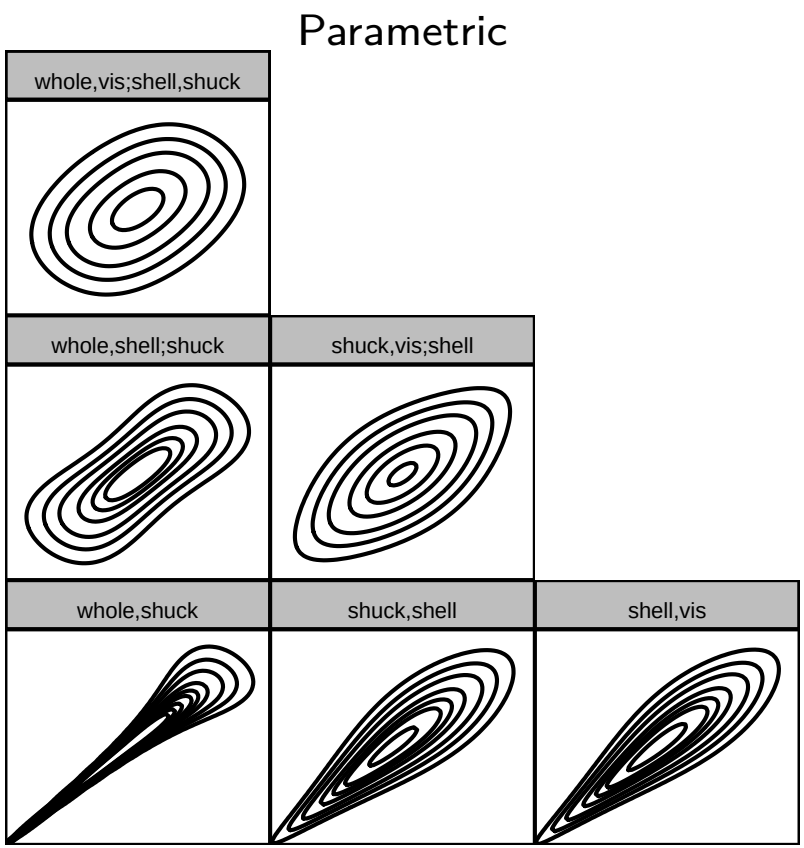
Linear quantile regression for whole against shuck, vis and shell

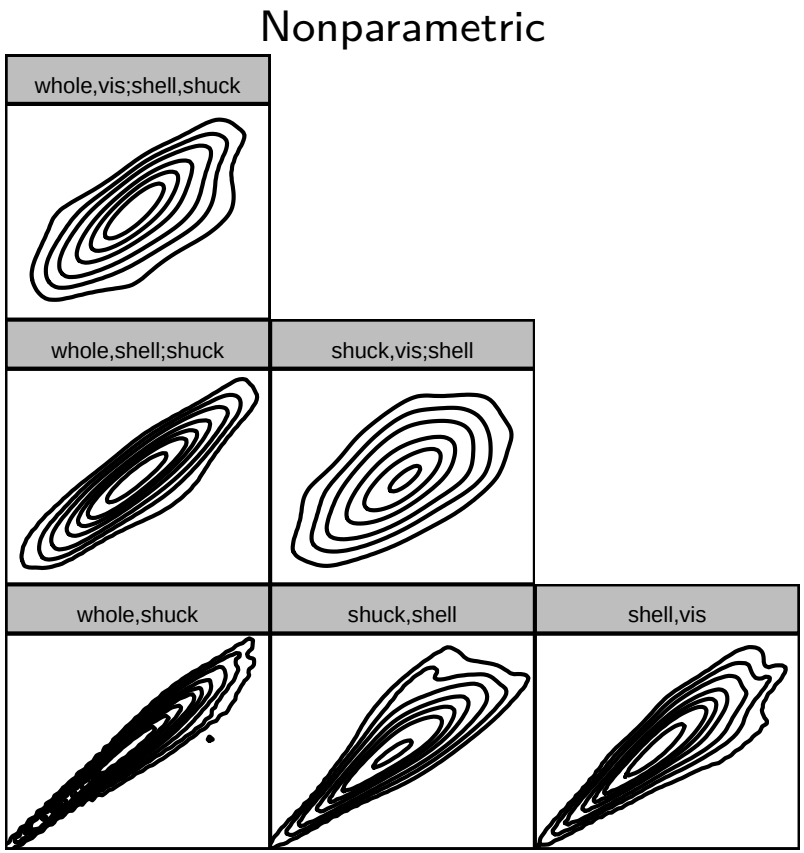
```
pred_lqr_whole <- pred_whole_dvqr_np
for (a in seq_along(alpha_vec)) {
    my.rq <- quantreg::rq(
        whole  ~ shuck+vis+shell ,
        tau = alpha_vec[a],
        data = abalone.f4
    )
    pred_lqr_whole[, a] <- quantreg::predict.rq(my.rq)
}
plot_marginal_effects(abalone.f4[,c("shuck","vis","shell")], pred_lqr_whole)
```



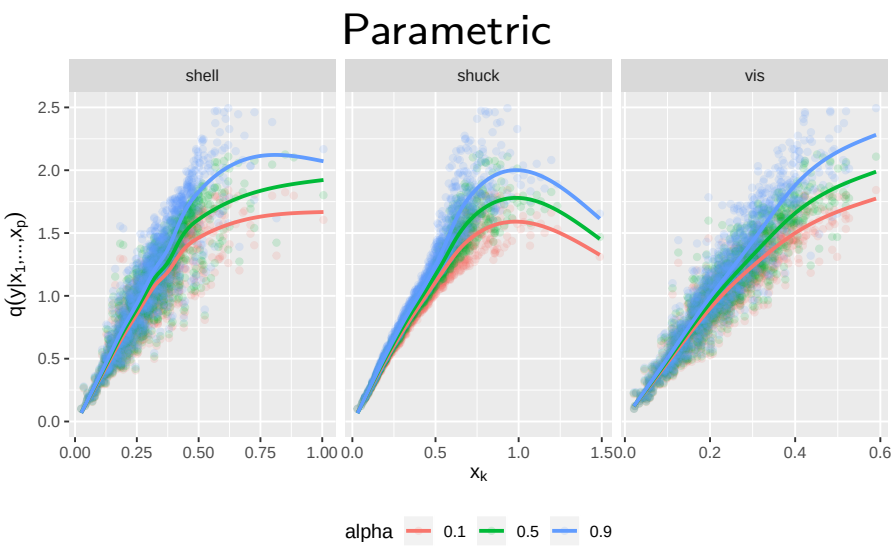
7. Comparison between non parametric/parametric/linear quantile regression model

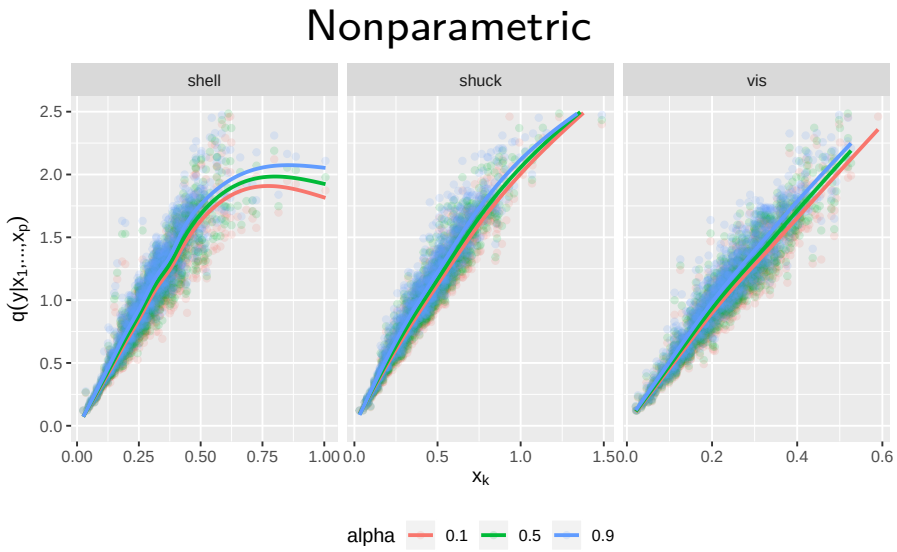
Fitted normalized contour plots used in parametric and non parametric quantile regression models



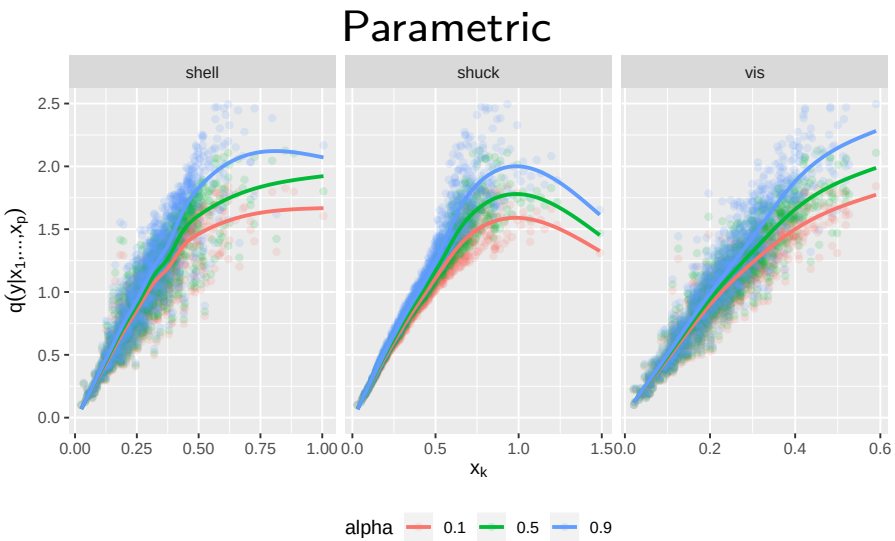


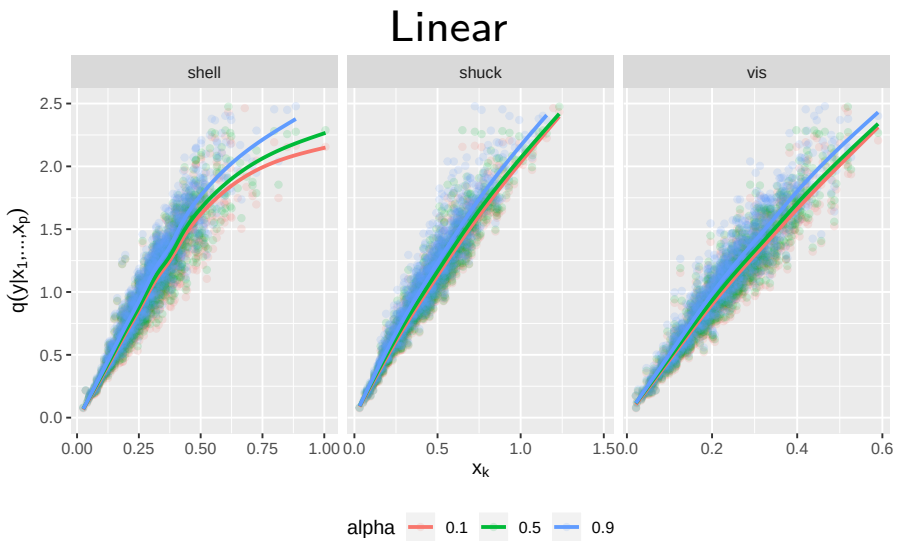
Marginal effects for parametric and non parametric quantile fitted values





Marginal effects for parametric and linear quantile fitted values





Marginal effects for non parametric and linear quantile fitted values