Data analysis of simulated soil erosion data

Xiaodan Lyu, Emily Berg and Heike Hofmann

2020-05-13

```
rm(list = ls(all = T))
library(dplyr)
library(ggplot2)
library(patchwork)
## load model package
if(!require(saezero)) devtools::install_github("XiaodanLyu/saezero")
library(saezero)
## load pseudo data due to confidentiality
data(erosion)
glimpse(erosion)
```

```
## Observations: 646
## Variables: 10
## $ ctylab <chr> "Aurora", "Aurora", "Aurora", "Aurora", "Aurora", "Aurora", ...
## $ cty    <chr> "003", "003", "003", "003", "003", "003", "003", "003", "005...
## $ mukey  <dbl> 354836, 354837, 354841, 354844, 354847, 354851, 354866, 3548...
## $ crop   <chr> "others", "soybean", "others", "others", "others", "others",...
## $ logR   <dbl> 4.553877, 4.553877, 4.553877, 4.553877, 4.553877, 4.553877, ...
## $ logK   <dbl> -1.4271164, -1.4271164, -1.4271164, -0.9942523, -1.1394343, ...
## $ logS   <dbl> -0.3549698, -2.1465136, -1.0483327, -2.1465136, -1.0483327, ...
## $ crop2  <dbl> 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
## $ crop3  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, ...
## $ RUSLE2 <dbl> 0.091, 0.009, 0.007, 0.034, 0.087, 0.000, 0.006, 0.001, 0.00...
```

Figure 1

```
g1 <- erosion %>% ggplot(aes(x = RUSLE2)) +
  geom_histogram(aes(y = ..density..), binwidth = 0.01,
                 color = "black", fill = NA, size = rel(1)) +
  geom_rug(sides = "b") +
  xlab("RUSLE2") + theme_bw(base_size = 15)
erosion2 <- erosion %>% group_by(cty) %>%
  summarise(
    ybar = mean(log(RUSLE2)[RUSLE2>0]),
    pbar = mean(RUSLE2>0),
    count = n()) %>%
  filter(!is.na(ybar))
g2 <- erosion2 %>% ggplot() +
  geom_point(aes(x = ybar, y = pbar, size = count), color = "black") +
```

```
  labs(x = "Average of positive RUSLE2s in log scale",
       y = "Proportion of positive RUSLE2s",
       size = "Sample size", title = "Counties") +
  theme_bw(base_size = 15) + ylim(0, 1) +
  theme(plot.title = element_text(hjust = 0.5),
        plot.margin = margin(0, 0, 0, 1, "cm"))
g1 + g2 + plot_layout(widths = c(6, 4))
```

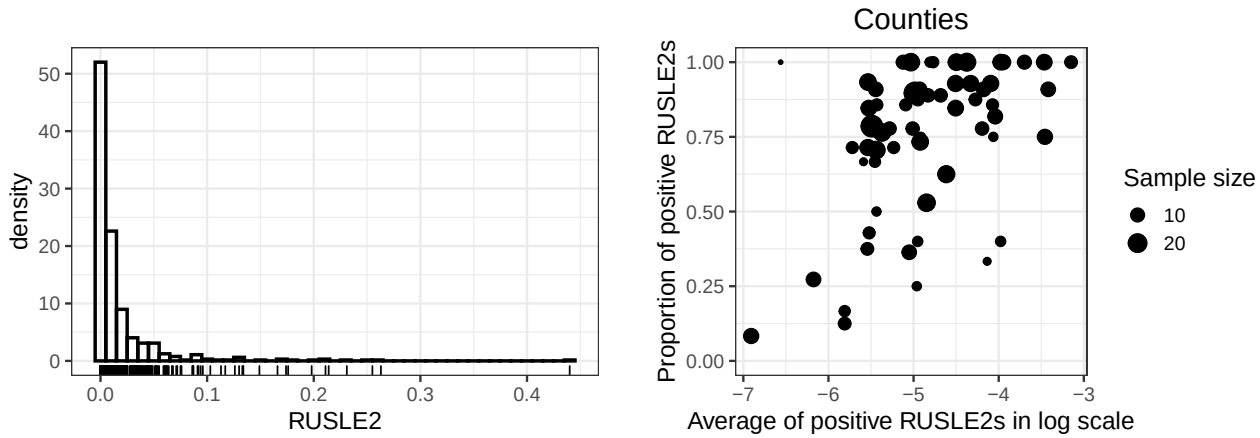


Table 4

```
## data pre-processing
erosion_2p <- as.2pdata(f_pos = RUSLE2~logR+logK+logS,
                        f_zero = ~logR+logS+crop2+crop3,
                        f_area = ~cty, data = erosion)
## Maximum Likelihood Estimates
fit <- mleLBH(erosion_2p)
```

```
## parametric bootstrap
## 2-3 hrs
system.time({
  set.seed(2015)
  est.store <- do.call(
    "rbind.data.frame",
    lapply(1:500, function(b){
      ys <- simLBH(fit, erosion, f_pos = ~logR+logK+logS,
                   f_zero = ~logR+logS+crop2+crop3, f_area = ~cty)
      erosion_boot <- erosion %>% mutate(RUSLE2 = ys)
      erosion_boot_2p <- as.2pdata(f_pos = RUSLE2~logR+logK+logS,
                                   f_zero = ~logR+logS+crop2+crop3,
                                   f_area = ~cty, data = erosion_boot)
      fit_boot <- mleLBH(erosion_boot_2p)
      c(unlist(fit_boot$fixed), fit_boot$refcor)
    }))
})
colnames(est.store) <- c(paste0("beta", 0:3), paste0("alpha", 0:4), "rho")
save(est.store, file = "data/theta_boot.RData")
```

```r
load("data/theta_boot.RData")
## fixed effect coefficients and bootstrap standard errors
est <- c(beta = fit$fixed$p1[-1], alpha = fit$fixed$p0[-1])
se <- apply(t(est.store[,c(2:4, 6:9)]) - est, 1, sd)
coef <- sprintf("%.2f (%.2f)", est, se)
names(coef) <- names(est)
coef
```

```
##         beta1         beta2         beta3        alpha1        alpha2
## "1.83 (0.32)" "0.38 (0.15)" "0.52 (0.06)" "4.82 (0.64)" "0.17 (0.15)"
##        alpha3        alpha4
## "0.62 (0.32)" "1.42 (0.38)"
```

```r
## variance components
c(sig2lu = fit$refvar1, sig2b = fit$refvar0,
  sig2le = fit$errorvar, rho = fit$refcor) %>%
  print(digit = 2)
```

```
## sig2lu  sig2b sig2le    rho
##   0.22   0.47   1.20   0.64
```

```r
## 95% confidence interval of rho
round(quantile(est.store[,10], c(0.025, 0.975)), 2)
```

```
##  2.5% 97.5%
##  0.01  0.99
```

Shapiro-Wilk test of normality

```r
## standardized marginal residuals
stderr.mar <- na.omit(fit$residuals$mar)/sqrt(fit$refvar1 + fit$errorvar)
## standardized conditional residuals
stderr.con <- na.omit(fit$residuals$con)/sqrt(fit$errorvar)
shapiro.test(stderr.mar)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stderr.mar
## W = 0.99546, p-value = 0.1549
```

```r
shapiro.test(stderr.con)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  stderr.con
## W = 0.99502, p-value = 0.1078
```

Figure 2

```
predictions <- ebLBH(Xaux, f_q = ~cnt, erosion_2p, fit, fullpop = TRUE)
predictions <- predictions %>% mutate(cv = sqrt(mse)/eb)
predictions %>% glimpse()
```

```
## Observations: 64
## Variables: 4
## $ area <fct> 003, 005, 007, 009, 011, 013, 015, 017, 021, 023, 025, 027, 02...
## $ eb   <dbl> 0.017522184, 0.021843915, 0.002520479, 0.020928505, 0.02743210...
## $ mse  <dbl> 3.508430e-05, 2.707652e-05, 2.700753e-06, 4.347515e-05, 5.3349...
## $ cv   <dbl> 0.3380401, 0.2382133, 0.6520176, 0.3150520, 0.2662602, 0.21297...
```

```
## parametric bootstrap estimates of M2
set.seed(2020)
b <- 0
eb_boot.store <- mmse_boot.store <- c()
## take several hours
system.time(repeat{
  b <- b + 1
  ys <- simLBH(fit, erosion, f_pos = ~logR+logK+logS,
               f_zero = ~logR+logS+crop2+crop3, f_area = ~cty)
  erosion_boot <- erosion %>% mutate(RUSLE2 = ys)
  erosion_boot_2p <- as.2pdata(f_pos = RUSLE2~logR+logK+logS,
                               f_zero = ~logR+logS+crop2+crop3,
                               f_area = ~cty, data = erosion_boot)
  fit_boot <- mleLBH(erosion_boot_2p)
  eb_boot <- ebLBH(Xaux, f_q = ~cnt, erosion_boot_2p, fit_boot, fullpop = TRUE)$eb
  mmse_boot <- ebLBH(Xaux, f_q = ~cnt, erosion_boot_2p, fit, fullpop = TRUE)$eb
  eb_boot.store <- rbind(eb_boot.store, eb_boot)
  mmse_boot.store <- rbind(mmse_boot.store, mmse_boot)
  if(b>=100) break
})
m2_boot <- colMeans((eb_boot.store - mmse_boot.store)^2)
save(eb_boot.store, mmse_boot.store, m2_boot, file = "data/eb_boot.RData")
```

```
load("data/eb_boot.RData")
## pooled standard error of direct estimator
direct <- erosion %>% group_by(cty) %>%
  summarise(nis = n(),
            ctymean = mean(RUSLE2),
            pool = sqrt(var(erosion$RUSLE2)/nis)) %>%
  mutate(group = cut(nis, c(0, 5, 10, 20, 30)))
se_all <- direct %>% mutate(
  ID = 1:n(),
  semiboot = sqrt(predictions$mse+m2_boot),
  onestep = sqrt(predictions$mse))
se_long <- se_all %>%
  tidyr::gather(method, SE, pool, semiboot, onestep) %>%
  mutate(method = forcats::fct_recode(
    method, "Direct" = "pool",
    "EB (Semi-Boot)" = "semiboot",
```

```
    "EB (One-step)" = "onestep") %>%
      forcats::fct_relevel("Direct", "EB (One-step)"))
ggplot(se_long, aes(x = ID, y = SE)) +
  geom_line(aes(group = method, color = method, linetype = method)) +
  scale_linetype_manual(values = c(1,1,0)) +
  geom_point(aes(shape = method, color = method),
             size = rel(2), stroke = rel(1)) +
  facet_grid(.~group, scales = "free_x") +
  scale_color_manual(values = c("darkgrey", "black", "black")) +
  scale_shape_manual(values = c(8, 1, 4)) +
  labs(x = "County", y = "Standard Error",
       color = "", shape = "", linetype = "") +
  theme_bw(base_size = 18) +
  theme(legend.position = "bottom",
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        legend.background = element_rect(colour = "black", size = 0.5))
```

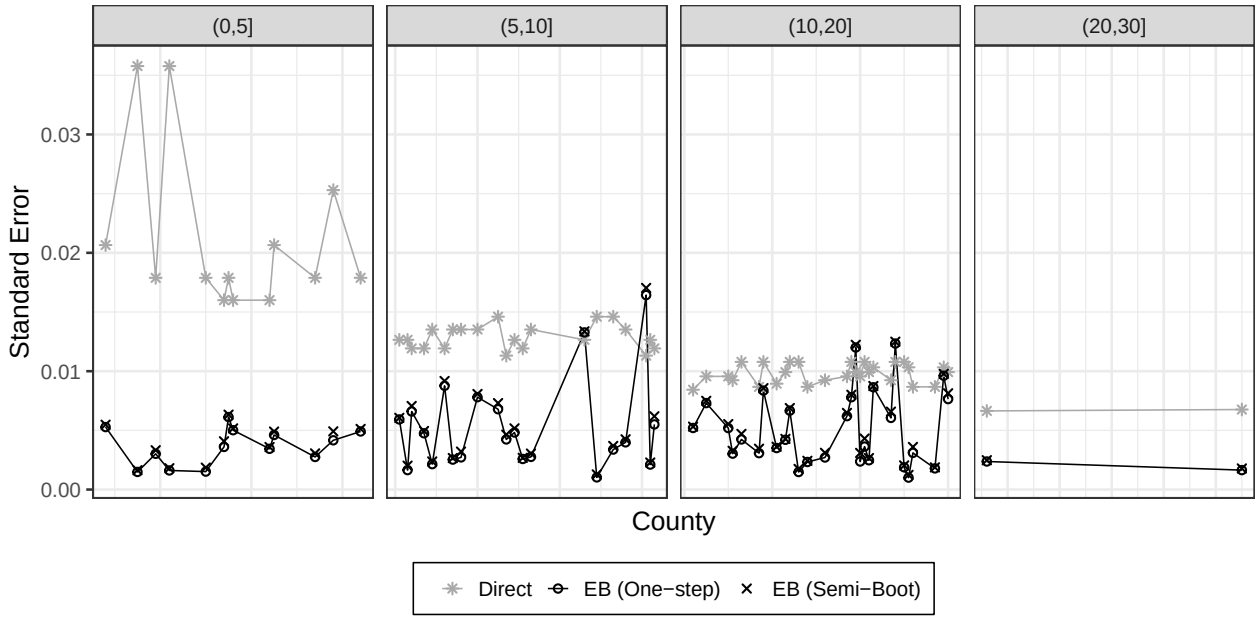
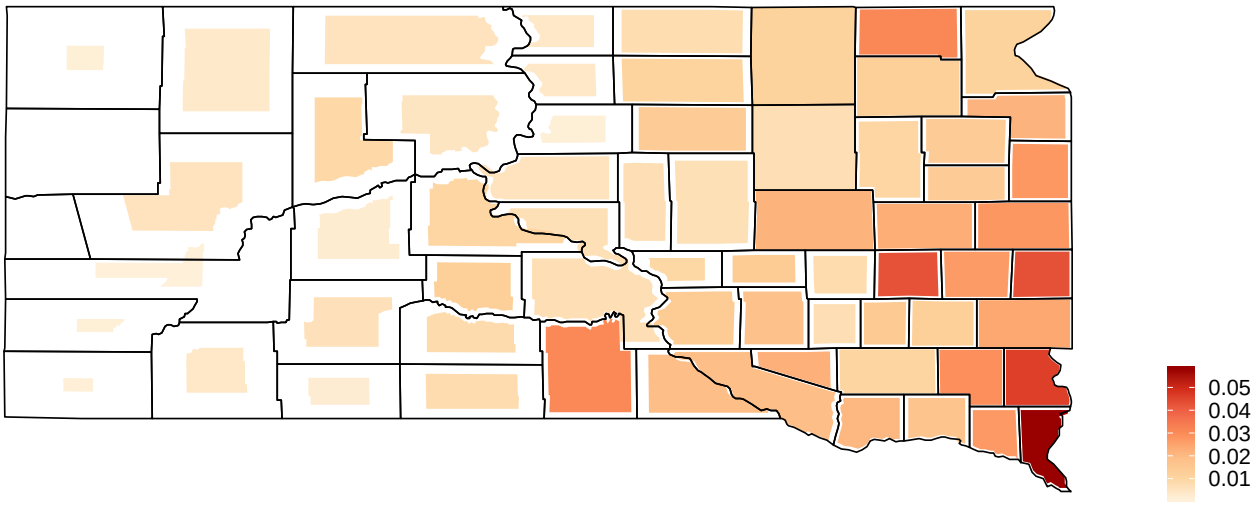


Figure 3

```
if(!require(ggmapr)) devtools::install_github("heike/ggmapr")
## EB predictions and one-step MSE estimates
map.eb <- map_sd %>% left_join(predictions, by = c("COUNTY" = "area"))
## scale by CV (coffecient of variance)
b <- (0.2-1)/diff(range(map.eb$cv, na.rm = T))
map.eb.df <- map.eb %>% mutate(s = b*(cv-min(cv, na.rm = T))+1)
scale_cty <- function(cty) {
  map.eb.df %>% filter(COUNTY == cty) %>% ggmapr::scale(scale = unique(.$s))
}
map.eb.scale <- do.call("rbind", lapply(unique(map.eb.df$COUNTY), scale_cty))
map.eb.scale %>%
```

```
  ggplot(aes(x = long, y = lat, group = factor(group))) +
  geom_polygon(aes(fill = eb), colour = NA) +
  geom_path(data = map_sd) +
  scale_fill_distiller("", palette = "OrRd", direction = 1) +
  scale_alpha(trans = "reciprocal", range = c(0.3, 1)) +
  ggthemes::theme_map(base_size = 18) +
  theme(legend.position = "right") +
  coord_equal()
```



Appendix B. Link Function Analysis

```
lambdas <- seq(-0.5, 0.5, by = 0.01)
profile_lam <- function(lambda){
  mleLBH(as.2pdata(f_pos = RUSLE2~logR+logK+logS,
                   f_zero = ~logR+logS+crop2+crop3,
                   f_area = ~cty, data = erosion,
                   transform = "BoxCox", lambda = lambda))$loglik
}
## 30-40 minutes
system.time(loglike_lambda <- sapply(lambdas, profile_lam))
plot(x = lambdas, y = exp(loglike_lambda-max(loglike_lambda)), type = "l")
abline(h = exp(-0.5*qchisq(0.95, 1)), col = "red", lty = "dashed")
(lambdahat <- lambdas[which.max(loglike_lambda)])
## 2-3 minutes each
rend <- uniroot(function(lambda) exp(profile_lam(lambda)-max(loglike_lambda))-exp(-0.5*qchisq(0.95, 1))
lend <- uniroot(function(lambda) exp(profile_lam(lambda)-max(loglike_lambda))-exp(-0.5*qchisq(0.95, 1))
save(loglike_lambda, lend, rend, file = "data/link_analysis.RData")
```

```
load("data/link_analysis.RData")
sprintf("(%.3f, %.3f)", lend$root, rend$root)
```

```
## [1] "(-0.123, 0.000)"
```

Figure S2

```
cty <- erosion %>% group_by(ctylab = tolower(ctylab)) %>%
  summarise(nis = n(), qi = mean(RUSLE2==0)) %>%
  mutate(sizegroup = cut(nis, c(0, 5, 10, 20, 30)),
         qigroup = cut(qi, seq(0, 1, 0.25)))
levels(cty$qigroup) <- c(levels(cty$qigroup), "0")
levels(cty$sizegroup) <- c(levels(cty$sizegroup), "0")
cty <- cty %>% mutate(qigroup = replace(qigroup, is.na(qigroup), "0"),
                      qigroup = relevel(qigroup, "0"))
cty2 <- left_join(map_sd, cty, by = c("subregion" = "ctylab"))
cty2 <- cty2 %>% mutate(sizegroup = replace(sizegroup, is.na(sizegroup), "0"),
                        sizegroup = relevel(sizegroup, "0"))
p1 <- cty2 %>% ggplot() +
  geom_polygon(aes(x = long, y = lat, group = factor(group), fill = qigroup), alpha = 0.6) +
  scale_fill_brewer(name = "", palette = "Greens", direction = -1) +
  geom_path(aes(x = long, y = lat, group = factor(group)))
p2 <- cty2 %>% ggplot() +
  geom_polygon(aes(x = long, y = lat, group = factor(group),
                   fill = sizegroup), alpha = 0.6) +
  scale_fill_brewer(name = "", palette = "Blues") +
  geom_path(aes(x = long, y = lat, group = factor(group)))
(p1 / p2) & coord_equal() &
  ggthemes::theme_map(base_size = 18) &
  theme(legend.position = "right")
```

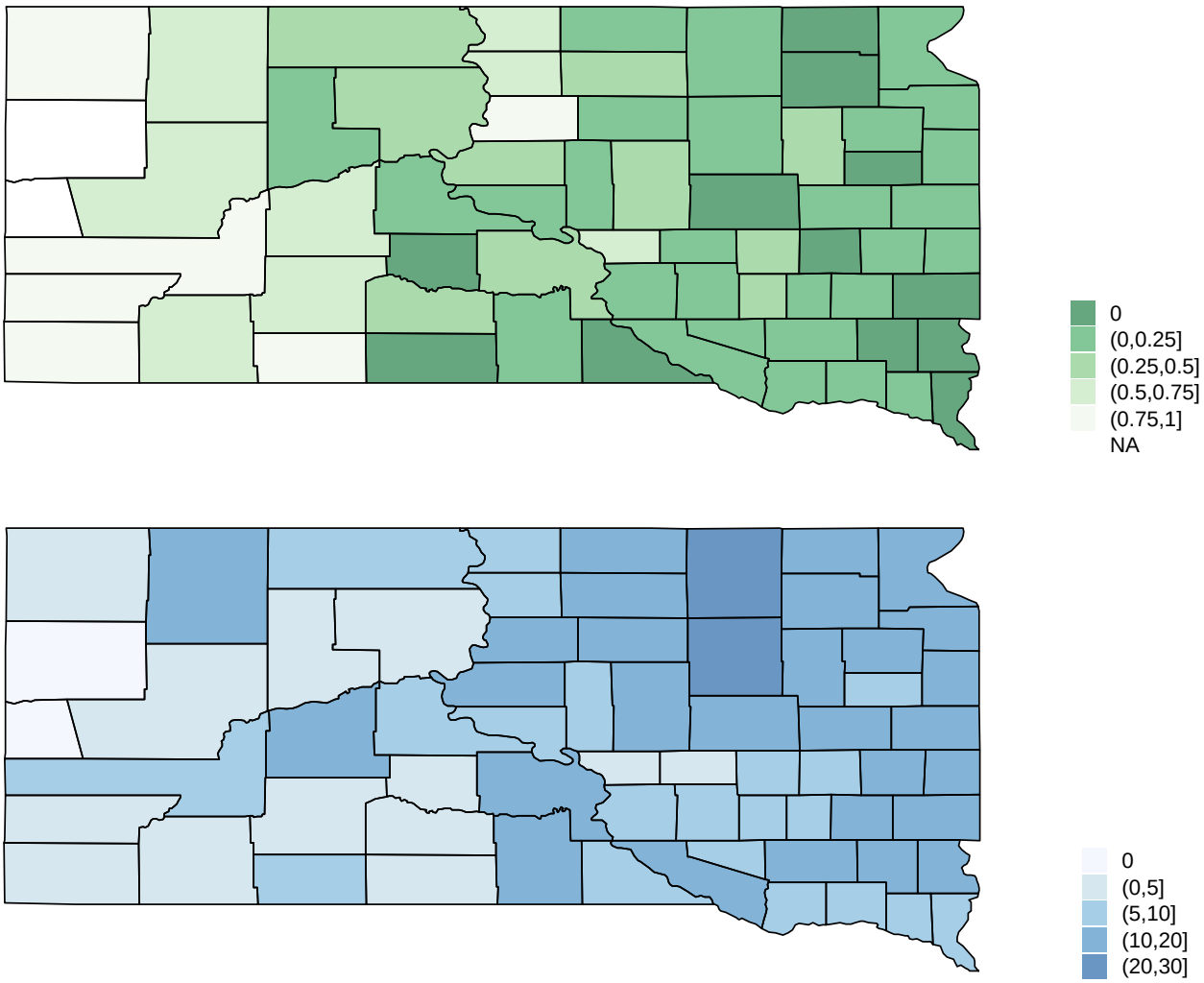


Figure S3

```
fitted.mar <- erosion_2p$lys - na.omit(fit$residuals$mar)
fitted.con <- erosion_2p$lys - na.omit(fit$residuals$con)
ggplot(data.frame(r = stderr.mar)) + stat_qq(aes(sample = r)) +
  geom_abline(slope = 1, intercept = 0) +
  labs(x = "theoretical quantile", y = "sample quantile") -> g1
ggplot(data.frame(x = fitted.mar, y = stderr.mar)) +
  geom_point(aes(x, y)) +
  labs(x = "Fitted Value", y = "Marginal Residual") +
  geom_hline(yintercept = 0) -> g2
ggplot(data.frame(r = stderr.con)) + stat_qq(aes(sample = r)) +
  geom_abline(slope = 1, intercept = 0) +
  labs(x = "theoretical quantile", y = "sample quantile") -> g3
ggplot(data.frame(x = fitted.con, y = stderr.con)) +
  geom_point(aes(x, y)) +
  labs(x = "Fitted Value", y = "Conditional Residual") +
  geom_hline(yintercept = 0) -> g4
(g1 | g2) / (g3 | g4) & theme_bw(base_size = 18)
```

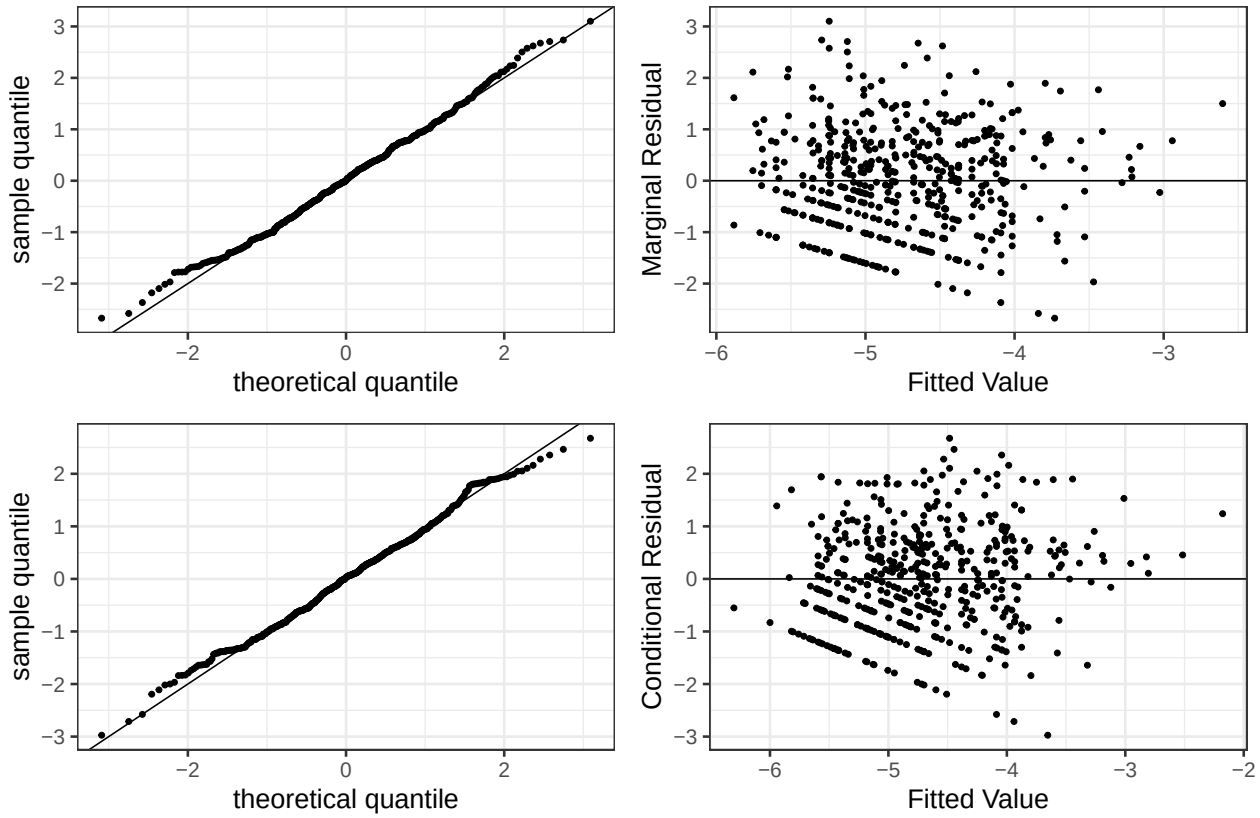


Figure S4

```
## Hosmer-Lemeshow Goodness of Fit Test
library(ResourceSelection)
etahat <- erosion_2p$Xs0 %*% fit$fixed$p0 +
  fit$random$p0[as.numeric(as.factor(erosion_2p$area))]
phat <- exp(etahat)/(1+exp(etahat))
pvalues <- sapply(5:15, function(i) hoslem.test(erosion_2p$deltas, phat, g = i)$p.value)
ggplot(data.frame(x = 5:15, y = pvalues), aes(x, y)) +
  geom_point(shape = 19, size = rel(2)) +
  geom_hline(yintercept = 0.1, linetype = "dashed")+
  ylim(0, 1) +
  labs(x = "number of groups", y = "p-value") +
  theme_bw(base_size = 18)
```

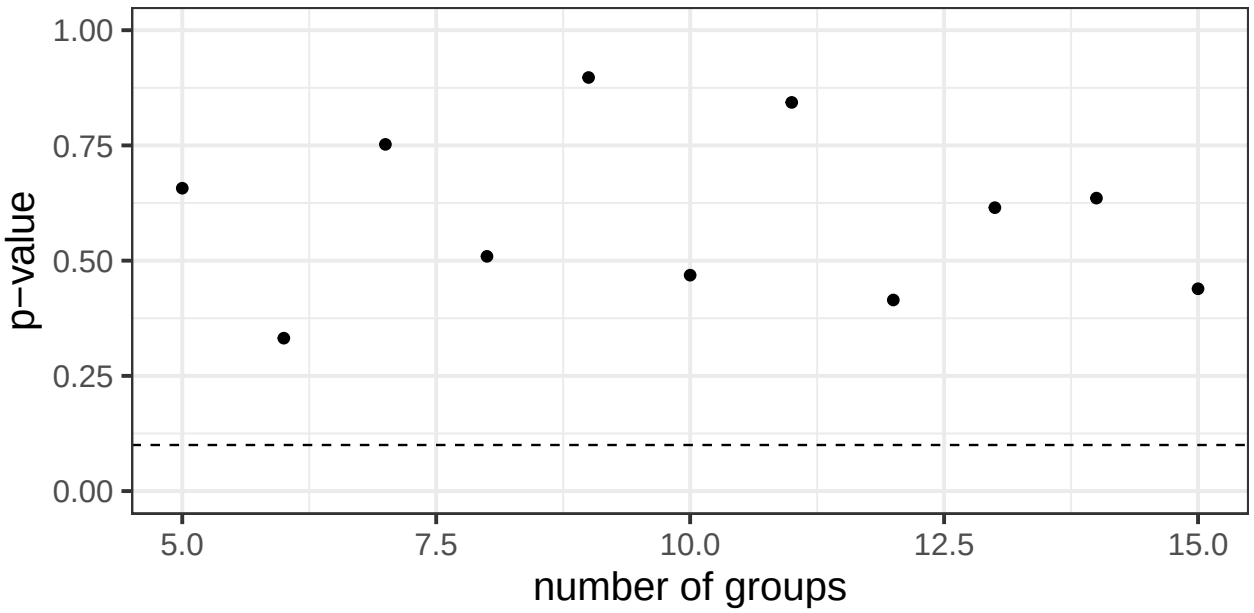
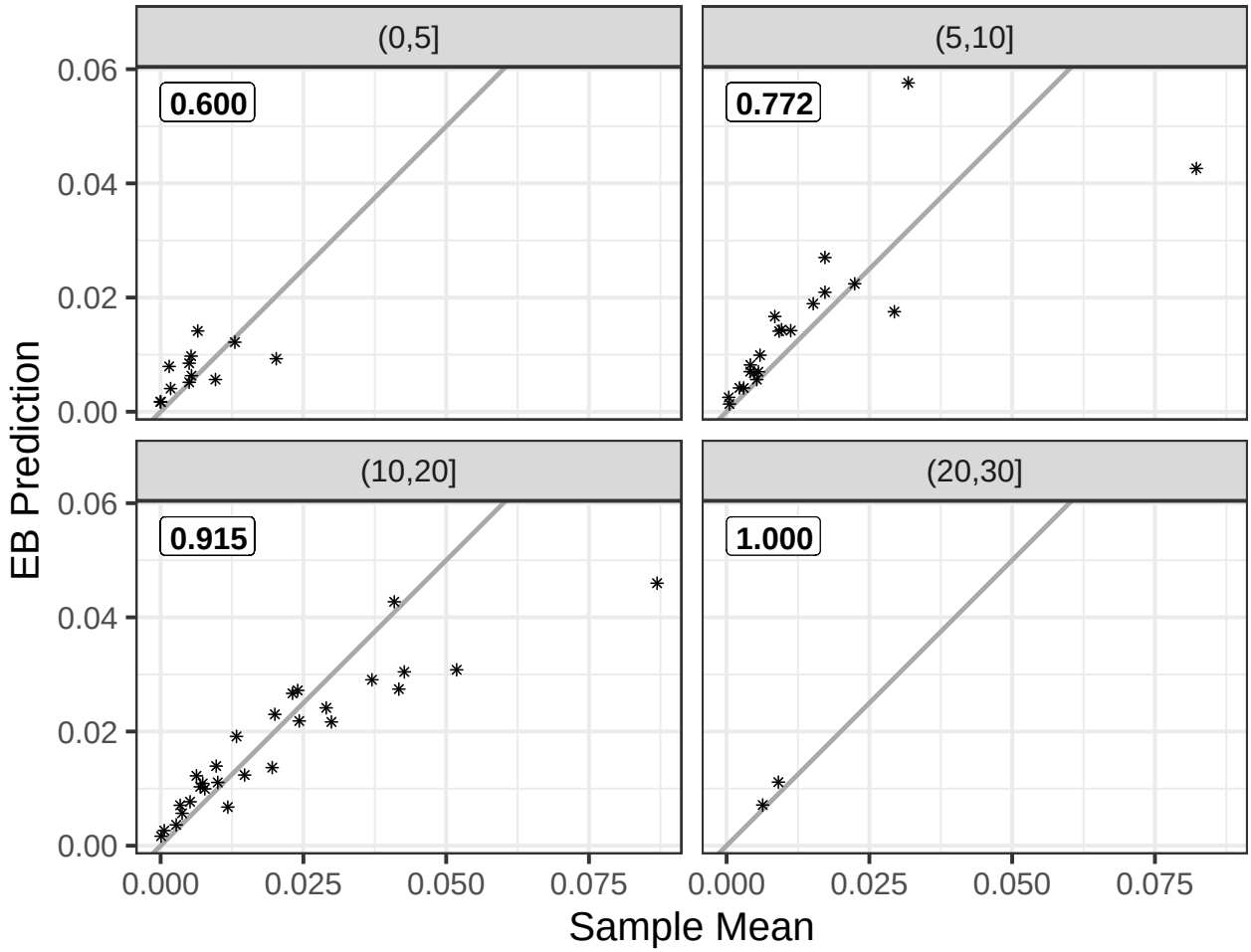


Figure S5

```
est_all <- direct %>% mutate(eb = predictions$eb) %>%
  group_by(group) %>% mutate(cor = cor(ctymean, eb))
ggplot(est_all, aes(x = ctymean, y = eb)) +
  geom_abline(slope = 1, size = rel(1), color = "darkgrey") +
  geom_point(shape = 8, size = rel(1.5)) +
  geom_label(aes(x = min(ctymean), y = max(eb),
                 label = sprintf("%.3f", cor)),
             hjust = 0, vjust = 1,
             size = 5, fontface = "bold") +
  theme_bw(base_size = 18) +
  labs(x = "Sample Mean", y = "EB Prediction") +
  coord_equal() + facet_wrap(~group, nrow = 2)
```



System Configurations

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.3
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ResourceSelection_0.3-5 ggmapr_0.1.0            saezero_0.1.0
## [4] patchwork_1.0.0.9000    ggplot2_3.2.1           dplyr_0.8.3
```

```
##
## loaded via a namespace (and not attached):
##  [1] statmod_1.4.34     tidyselect_1.0.0   xfun_0.11          reshape2_1.4.3
##  [5] purrr_0.3.3        ggthemes_4.2.0     splines_3.6.1      lattice_0.20-38
##  [9] colorspace_1.4-1   vctrs_0.2.2        htmltools_0.4.0    yaml_2.2.0
## [13] utf8_1.1.4         rlang_0.4.4        pillar_1.4.3       nloptr_1.2.1
## [17] foreign_0.8-71     glue_1.3.1         withr_2.1.2        RColorBrewer_1.1-2
## [21] sp_1.3-2           plyr_1.8.4         lifecycle_0.1.0    stringr_1.4.0
## [25] munsell_0.5.0      gtable_0.3.0       evaluate_0.14      labeling_0.3
## [29] knitr_1.28         forcats_0.4.0      maptools_0.9-9     sae_1.3
## [33] fansi_0.4.1        Rcpp_1.0.3         scales_1.1.0       farver_2.0.1
## [37] lme4_1.1-21        digest_0.6.23      stringi_1.4.5      grid_3.6.1
## [41] cli_2.0.1          tools_3.6.1        magrittr_1.5       lazyeval_0.2.2
## [45] tibble_2.1.3       crayon_1.3.4       tidyr_1.0.0        pkgconfig_2.0.3
## [49] MASS_7.3-51.4      ellipsis_0.3.0     Matrix_1.2-17      assertthat_0.2.1
## [53] minqa_1.2.4        rmarkdown_2.1      R6_2.4.1           boot_1.3-22
## [57] nlme_3.1-140       compiler_3.6.1
```