Results of Bootstraps

In [ ]:
```
library(dplyr)
library(xtable)
```

In [ ]:
```
setwd("/home/leoKraushaar/Documents/School/Year 3/Semester 2/STAT 413/Project/protests/")
```

In [ ]:
```
resamp_results <- read.csv("data/results/resamp_boot_results.csv")[, -1]
param_results <- read.csv("data/results/param_boot_results.csv")[, -1]
smooth_results <- read.csv("data/results/smooth_boot_results.csv")[, -1]
error_results <- read.csv("data/results/error_boot_results.csv")[, -1]
```

In [ ]:
```
head(smooth_results)
```

	X.Intercept.	monthSpring	monthSummer	monthWinter	provBritish.Columbia	provManitoba	provNew.Brunswick	provNewfoundland.and.l
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	2.934920	-0.04245997	-0.5492539	-0.1948508	0.6515551	-0.5477383	-1.069497	-1
2	2.904923	-0.04074207	-0.5486760	-0.1950555	0.6414165	-0.4868302	-1.001618	-
3	2.984455	-0.04330338	-0.5468556	-0.1995221	0.6539932	-0.6336642	-1.164881	-
4	3.236990	-0.04804336	-0.5515612	-0.2117240	0.7190518	-1.1129440	-1.707100	-2
5	3.141159	-0.04155130	-0.5467633	-0.1992153	0.6917732	-0.9199258	-1.500135	-
6	3.278271	-0.02782923	-0.5458436	-0.1909998	0.7553655	-1.1985879	-1.813752	-

In [ ]:
```
head(error_results)
```

	X.Intercept.	seasonSpring	seasonSummer	seasonWinter	provBritish.Columbia	provManitoba	provNew.Brunswick	provNewfoundland.and
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	3.699515	-0.04939341	-0.5644029	-0.2540712	0.7970201	-2.006120	-2.671376	-
2	3.625176	-0.05852822	-0.5528195	-0.2371354	0.8058068	-1.841590	-2.525398	-
3	3.709236	-0.07304630	-0.5725859	-0.2355005	0.8071427	-1.939634	-2.735903	
4	3.616712	-0.05696503	-0.5356797	-0.2357946	0.8259597	-1.809953	-2.501839	
5	3.627947	-0.06911946	-0.5413708	-0.2521120	0.8041241	-1.844577	-2.525781	-
6	3.652942	-0.05582750	-0.5327621	-0.2334410	0.8185922	-1.896705	-2.601671	

In [ ]:
```
newnames <- colnames(error_results)[2:4]
colnames(resamp_results)[2:4] <- newnames
colnames(param_results)[2:4] <- newnames
colnames(smooth_results)[2:4] <- newnames
```

Confidence intervals

In [ ]:
```
interval <- function(X, alpha=0.05) {
    quantiles <- quantile(X, probs = c(alpha/2, 1/2, 1-(alpha/2)))
    return(quantiles)
}
```

In [ ]:
```
mean_and_sd <- function(X) {
    return(c(mean(X), sd(X)))
}
```

In [ ]:
```
params <- rbind(resamp_results, param_results, smooth_results, error_results)
type <- factor(rep(c("R", "P", "S", "E"), each=10000))

results <- data.frame(type, params)
head(results)
```

	type	X.Intercept.	seasonSpring	seasonSummer	seasonWinter	provBritish.Columbia	provManitoba	provNew.Brunswick	provNewfoundla
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	R	3.244585	0.006926766	-0.3788602	-0.1090520	0.9351093	-1.4236981	-2.010733	
2	R	3.260520	-0.138461074	-0.6429856	-0.2252399	0.6844205	-1.0172718	-1.743031	
3	R	4.766945	-0.147152623	-0.7518559	-0.2362456	1.0398042	-3.6080141	-4.505630	
4	R	3.709305	-0.118238225	-0.6555250	-0.3405262	0.7107021	-1.9523412	-2.606208	
5	R	3.552543	-0.197672329	-0.6612784	-0.1292685	0.9047185	-1.4811943	-2.291576	
6	R	3.012924	-0.121514940	-0.6325036	-0.1695788	0.8867711	-0.8422924	-1.191386	

In [ ]:
```
month_indices <- grepl("month", colnames(results))
month_labels <- colnames(results)[month_indices]
month_labels <- sapply(month_labels, function(x) substr(x, 6, 8))

prov_indices <- grepl("prov", colnames(results))
prov_labels <- colnames(results)[prov_indices]
matches <- gregexpr("[A-Z]", prov_labels)
capital_letters <- regmatches(prov_labels, matches)
combined_letters <- sapply(capital_letters, function(x) paste(x, collapse = ""))
```

In [ ]:
```
new_names <- colnames(results)[c(1, 2, 3:ncol(results))]

new_names[month_indices] <- month_labels
new_names[prov_indices] <- combined_letters
new_names[2] <- "intercept"

colnames(results) <- new_names
```

In [ ]:
```
split_data <- split(results, type)
```

Resampling Results

In [ ]:
```
R_data <- split_data$R
head(R_data)
```

A data.frame: 6 × 18

	type	intercept	seasonSpring	seasonSummer	seasonWinter	BC	M	NB	NL	NT	NS	
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	R	3.244585	0.006926766	-0.3788602	-0.1090520	0.9351093	-1.4236981	-2.010733	-2.692642	-4.857194	-2.057455	-4.3
2	R	3.260520	-0.138461074	-0.6429856	-0.2252399	0.6844205	-1.0172718	-1.743031	-2.039554	-3.623698	-1.370148	-3.5
3	R	4.766945	-0.147152623	-0.7518559	-0.2362456	1.0398042	-3.6080141	-4.505630	-5.352689	-8.452046	-4.269014	-7.46
4	R	3.709305	-0.118238225	-0.6555250	-0.3405262	0.7107021	-1.9523412	-2.606208	-2.614093	-5.208495	-2.246478	-5.08
5	R	3.552543	-0.197672329	-0.6612784	-0.1292685	0.9047185	-1.4811943	-2.291576	-2.839374	-5.344607	-2.297504	-4.45
6	R	3.012924	-0.121514940	-0.6325036	-0.1695788	0.8867711	-0.8422924	-1.191386	-1.660419	-3.773542	-1.144260	-3.01

In [ ]:
```
alpha = 0.05

R_mean_and_sd <- apply(X=R_data[2:ncol(R_data)], MARGIN = 2, FUN = function(x) mean_and_sd(x))
R_intervals <- apply(X=R_data[2:ncol(R_data)], MARGIN = 2, FUN = function(x) interval(x, alpha))

R_results <- rbind(R_mean_and_sd,
                   R_intervals)

rownames(R_results)[1:2] <- c("mean", "sd")

R_results <- t(R_results)

contains <- sapply(1:nrow(R_results), function(i) {
    (R_results[i, "2.5%"] <= 0) & (R_results[i, "97.5%"] >= 0)
})

R_results <- as.data.frame(R_results)
R_results$contains <- contains
R_results$sig <- !contains

R_results$col <- ifelse(R_results$contains == TRUE, 2, 1)
R_results$abs_Z <- R_results$mean / R_results$sd

print(xtable(R_results[, -c(6,8)]), type = "latex", file = "resamp.tex")
```

In [ ]:

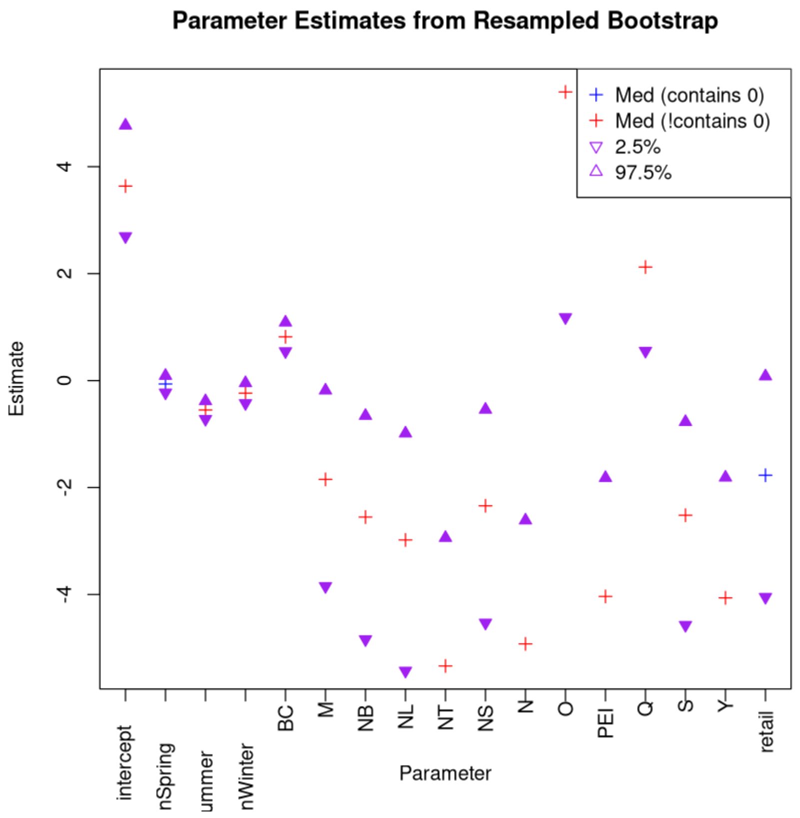
In [ ]:
```
# print(xtable(R_results[, 1:ncol(R_results)-1], type = "latex"), file = "filename2.tex")
```

In [ ]:
```
# Plot the median
palette(c("red", "blue"))
plot(R_results$`50%`, pch=3, type = "p", col = R_results$col, xlab = "Parameter", ylab = "Estimate", main = "Parame

# Add custom x-axis with row names as labels
axis(1, at = 1:nrow(R_results), labels = FALSE)
text(1:nrow(R_results), par("usr")[3] - 0.1, srt = 90, adj = 1.4, labels = rownames(R_results), xpd = TRUE)

# Add 2.5% and 97.5% rows with triangles
points(R_results$`2.5%`, type = "p", pch = 25, bg="purple", col="purple") # Downward triangle for 2.5%
points(R_results$`97.5%`, type = "p", pch = 24, bg="purple", col="purple") # Upward triangle for 97.5%

# Add a legend
legend("topright", legend = c("Med (contains 0)", "Med (!contains 0)", "2.5%", "97.5%"),
    col = c("blue", "red", "purple", "purple"), pch = c(3, 3, 25, 24), lty = 0)
```



Parametric Bootstrap

In [ ]:
```
P_data <- split_data$P
P_data
```

A data.frame: 10000 × 18

	type	intercept	seasonSpring	seasonSummer	seasonWinter	BC	M	NB	NL	NT	
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
10001	P	3.963172	-0.025441331	-0.5394282	-0.20774660	0.9894374	-2.3770425	-3.1694738	-3.8691247	-5.811581	-3.
10002	P	3.625495	-0.123532045	-0.6569072	-0.25612222	0.9308248	-1.8412242	-2.5448170	-2.8677713	-4.813196	-2.2
10003	P	3.231448	-0.083204960	-0.4454896	-0.18996214	0.5850643	-1.0456440	-1.5067522	-1.8791780	-4.369095	-1.6
10004	P	3.714770	0.051120142	-0.5743253	-0.22819306	1.0010481	-1.8812487	-2.6931498	-3.2693532	-5.779182	-2.4
10005	P	2.739805	-0.131038830	-0.5862504	-0.15223095	0.5540425	-0.1736415	-0.5632706	-0.8435661	-2.646468	-0.4
10006	P	4.086451	0.007154692	-0.5574182	-0.35073579	0.9401045	-2.6449460	-3.4885470	-4.2069282	-6.972127	-3.1
10007	P	3.098026	-0.115015400	-0.5463144	-0.23493110	0.6256107	-0.9446242	-1.2499443	-1.8048703	-3.840425	-1.1
10008	P	4.465287	-0.160730742	-0.5351699	-0.28129346	1.1585054	-3.4625239	-4.2877986	-4.8856748	-7.959978	-4.
10009	P	3.210447	-0.051048602	-0.5299081	-0.25652571	0.8444832	-1.1314215	-1.8254969	-2.1770043	-4.059834	-1.6
10010	P	3.874576	0.011841656	-0.4679920	-0.16240654	0.9975924	-2.3959363	-3.2482877	-3.6051183	-6.388035	-2.8
10011	P	3.465099	-0.091779333	-0.4780929	-0.18849862	0.8304001	-1.5251792	-2.3537467	-2.9414951	-5.245449	-2.0
10012	P	3.214803	-0.022455374	-0.7503301	-0.21901614	0.8236053	-1.2153860	-1.7401061	-2.1523595	-4.227023	-1.5
10013	P	3.350600	-0.063029102	-0.4350472	-0.18065724	0.6088835	-1.3151317	-1.8569667	-2.5173977	-5.123753	-1.6
10014	P	3.997284	-0.032332115	-0.4185573	-0.11640934	0.9177966	-2.7321259	-3.1319498	-4.2433394	-6.180524	-3.2
10015	P	3.464363	0.085018120	-0.5057278	-0.17434516	0.6419397	-1.5041108	-2.2778902	-2.7813705	-4.603006	-2.1
10016	P	2.480641	0.068053556	-0.4079774	-0.05672417	0.6745696	0.2051766	-0.2964067	-0.5852987	-2.636503	-0.
10017	P	4.081737	-0.073743213	-0.5096786	-0.28360183	0.7927525	-2.5554205	-3.2886457	-3.7070401	-6.787050	-3.0
10018	P	3.074808	0.016422557	-0.6217928	-0.22771574	0.5532824	-0.7405074	-1.4310618	-1.7686945	-3.634016	-1.3
10019	P	2.822826	0.128140283	-0.4637101	-0.08458616	0.8825775	-0.4490553	-1.4066468	-1.3777587	-3.976857	-0.8
10020	P	3.580907	-0.101492965	-0.5163413	-0.32832448	0.6512331	-1.5394147	-2.2098349	-2.5267655	-5.176555	-2.1
10021	P	3.554041	0.039512762	-0.6006763	-0.12185511	0.6969355	-1.6816014	-2.1236982	-2.8224022	-4.580680	-2.1
10022	P	3.734686	-0.243340179	-0.4919737	-0.30857982	0.8333082	-2.1414847	-2.6416565	-2.9203183	-5.541027	-2.4
10023	P	3.416295	-0.020134384	-0.4908029	-0.24575434	0.8245288	-1.4496369	-2.0343432	-2.6419570	-4.485506	-2.0
10024	P	4.303531	-0.243768035	-0.6107861	-0.37269990	1.0964836	-2.7966138	-3.7442123	-4.2730327	-6.469610	-3.2
10025	P	3.736701	-0.066190159	-0.4644469	-0.24622755	0.9272975	-1.9158812	-2.7751770	-3.3500015	-5.668535	-2.4
10026	P	3.713427	-0.019277039	-0.6258030	-0.19258427	1.0049982	-2.0339806	-2.8823319	-3.3215294	-5.420008	-2.5
10027	P	3.077516	-0.056443581	-0.5388238	-0.16477671	0.5820978	-0.8896446	-1.5321824	-1.8459377	-3.920946	-1.3
10028	P	3.877133	-0.086251515	-0.4855447	-0.29627024	0.7950820	-2.0349971	-3.0133531	-3.5466246	-5.820884	-2.8
10029	P	4.160267	-0.084103842	-0.6731616	-0.35453847	0.8858476	-2.6368667	-3.5295868	-4.0067744	-6.531960	-3.0
10030	P	4.179225	-0.100057320	-0.5821935	-0.23211039	1.1851016	-2.9199360	-3.8998196	-4.3092338	-7.190682	-3.4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
19971	P	3.922307	-0.1840713830	-0.5976597	-0.30679364	0.8360477	-2.26247570	-3.1138402	-3.4712444	-5.841009	-2.6
19972	P	3.723739	-0.1291270320	-0.6541084	-0.41021250	0.8307505	-1.99727599	-2.6141837	-3.0884166	-4.934490	-2.3
19973	P	3.375701	-0.1372822103	-0.8851829	-0.25477208	0.6693939	-1.08777853	-1.9076157	-2.1772981	-4.125170	-1.5
19974	P	2.986010	-0.1664248211	-0.6251844	-0.24848266	0.6437157	-0.59173115	-1.0723728	-1.6212631	-3.525911	-0.9
19975	P	4.040909	-0.0255836492	-0.7303429	-0.24805919	1.0090963	-2.58330605	-3.2366404	-3.9641620	-5.732589	-3.
19976	P	4.285958	-0.1469267106	-0.5846654	-0.29528298	0.9052870	-2.80125216	-3.4606444	-4.1797200	-6.978336	-3.3
19977	P	2.426414	0.1412956025	-0.5229547	-0.19905388	0.6945921	0.17205402	-0.2795703	-0.5889814	-2.708896	-0.
19978	P	4.173659	-0.0056754087	-0.4486016	-0.23342589	0.8009610	-2.89730055	-3.6081510	-4.3172414	-7.089221	-3.3
19979	P	3.422301	-0.1010708105	-0.7462384	-0.22348730	0.7617241	-1.35411133	-2.0551465	-2.5586230	-4.493835	-1.7
19980	P	3.045413	-0.0628007393	-0.4719591	-0.15129364	0.6185508	-0.82919512	-1.4438170	-1.8897486	-4.228409	-1.1
19981	P	4.203584	-0.1027525235	-0.6460967	-0.24637741	0.6948458	-2.69942610	-3.4515189	-3.7509864	-6.743648	-3.0
19982	P	3.068661	-0.1550414651	-0.6325369	-0.22913731	0.8027402	-0.66501263	-1.3550398	-1.7072885	-3.687121	-1.0
19983	P	3.765064	-0.0889594029	-0.6563636	-0.43526599	0.8745679	-1.84014492	-2.6404258	-3.2003053	-5.532267	-2.3
19984	P	3.395730	-0.1289337369	-0.6502018	-0.32651716	0.7169369	-1.09193780	-2.0491493	-2.3206213	-5.183448	-1.6
19985	P	4.494615	0.0019120966	-0.4781124	-0.29469558	0.6819522	-3.35442836	-4.0623495	-4.8376818	-6.857696	-3.7
19986	P	2.728471	-0.0004633587	-0.5887597	-0.30577960	0.5129422	-0.09145382	-0.4028928	-0.7412871	-3.002060	-0.

	type	intercept	seasonSpring	seasonSummer	seasonWinter	BC	M	NB	NL	NT	
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
19987	P	4.204175	-0.1311045610	-0.5531199	-0.50762413	0.7924015	-2.55410542	-3.4425254	-3.9786600	-6.343277	-3.0
19988	P	4.307881	-0.1283574811	-0.5202520	-0.30077020	0.8471243	-3.03786994	-3.8347666	-4.3349951	-6.890604	-3.5
19989	P	3.536774	-0.0333670858	-0.6036621	-0.17102324	0.5590245	-1.41099963	-2.3166370	-2.7498841	-5.250662	-2.0
19990	P	3.819316	-0.1598460071	-0.5065319	-0.23804587	0.8967563	-2.02844545	-2.9496618	-3.1779625	-6.114514	-2.6
19991	P	3.443132	-0.1553963547	-0.5938737	-0.30614209	0.6666251	-1.23339752	-1.8744896	-2.4444806	-4.470226	-1.7
19992	P	2.684542	0.0832175347	-0.5349234	-0.11569904	0.6782929	-0.29357528	-0.5271464	-1.2496118	-3.246023	-0.4
19993	P	3.800504	-0.0463417688	-0.5069850	-0.21198392	0.9834718	-2.22743403	-3.0081614	-3.4046789	-6.487022	-2.7
19994	P	3.922196	-0.0748455440	-0.5937210	-0.13620395	0.8608902	-2.40350865	-3.2408915	-3.6223381	-5.565976	-2.7
19995	P	4.119777	-0.0285014886	-0.4489953	-0.21513497	0.9187696	-2.70694915	-3.5098377	-4.0085237	-6.361484	-3.4
19996	P	3.717265	-0.0901500542	-0.5763614	-0.22747959	0.8798356	-2.03004047	-2.9766043	-3.2141708	-5.345047	-2.5
19997	P	3.433786	-0.0303653463	-0.4021653	-0.18523779	0.7369193	-1.43552455	-2.6044686	-2.8671004	-4.999158	-2.1
19998	P	3.098646	0.0514582622	-0.4535218	-0.09281154	0.7800067	-0.97911648	-1.7288948	-2.0269420	-4.097409	-1.3
19999	P	3.910570	-0.0983818092	-0.5786447	-0.24443163	0.8892209	-2.29227209	-2.9736451	-3.2964327	-5.665786	-2.7
20000	P	3.049759	-0.0752211524	-0.5496037	-0.29652879	0.9230332	-0.81876030	-1.4509908	-1.9565901	-4.111581	-1.3

In [ ]:

```r
alpha = 0.05

P_mean_and_sd <- apply(X=P_data[2:ncol(P_data)], MARGIN = 2, FUN = function(x) mean_and_sd(x))
P_intervals <- apply(X=P_data[2:ncol(P_data)], MARGIN = 2, FUN = function(x) interval(x, alpha))

P_results <- rbind(P_mean_and_sd,
                   P_intervals)

rownames(P_results)[1:2] <- c("mean", "sd")

P_results <- t(P_results)

contains <- sapply(1:nrow(P_results), function(i) {
    (P_results[i, "2.5%"] <= 0) & (P_results[i, "97.5%"] >= 0)
})

P_results <- as.data.frame(P_results)
P_results$contains <- contains
P_results$sig <- !contains

P_results$col <- ifelse(P_results$contains == TRUE, 2, 1)

P_results[, 1:ncol(P_results)-1]
P_results$abs_Z <- P_results$mean / P_results$sd
```

A data.frame: 17 × 7

	mean	sd	2.5%	50%	97.5%	contains	sig
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<lgl>
intercept	3.67940258	0.51126548	2.6773091	3.67511313	4.69077470	FALSE	TRUE
seasonSpring	-0.06532076	0.08374888	-0.2295456	-0.06519182	0.10043350	TRUE	FALSE
seasonSummer	-0.55194157	0.08688979	-0.7236796	-0.55074506	-0.38213582	FALSE	TRUE
seasonWinter	-0.22869157	0.08965910	-0.4065933	-0.22995397	-0.05078027	FALSE	TRUE
BC	0.81822871	0.16416765	0.5051385	0.81748162	1.14329057	FALSE	TRUE
M	-1.91769183	0.92325906	-3.7485942	-1.90911200	-0.11100067	FALSE	TRUE
NB	-2.63122463	1.04759939	-4.7203129	-2.62641109	-0.56624051	FALSE	TRUE
NL	-3.07894069	1.11194691	-5.3064835	-3.06323557	-0.88791718	FALSE	TRUE
NT	-5.40678126	1.27490062	-7.9180310	-5.39421497	-2.91237127	FALSE	TRUE
NS	-2.41685188	0.99602853	-4.3900516	-2.40565950	-0.46400224	FALSE	TRUE
N	-5.01375817	1.27121516	-7.5238939	-5.00626317	-2.53171733	FALSE	TRUE
O	5.61433589	2.45433256	0.8046414	5.58701315	10.50540385	FALSE	TRUE
PEI	-4.13001808	1.21857541	-6.5354052	-4.11568268	-1.72776230	FALSE	TRUE
Q	2.19820238	0.93431833	0.3649575	2.18910987	4.06524340	FALSE	TRUE
S	-2.58652779	0.94259830	-4.4586517	-2.57098597	-0.74586577	FALSE	TRUE
Y	-4.15124719	1.24599908	-6.6127238	-4.14105063	-1.70864540	FALSE	TRUE
retail	-1.85982174	1.06640198	-3.9830471	-1.84893190	0.23150104	TRUE	FALSE

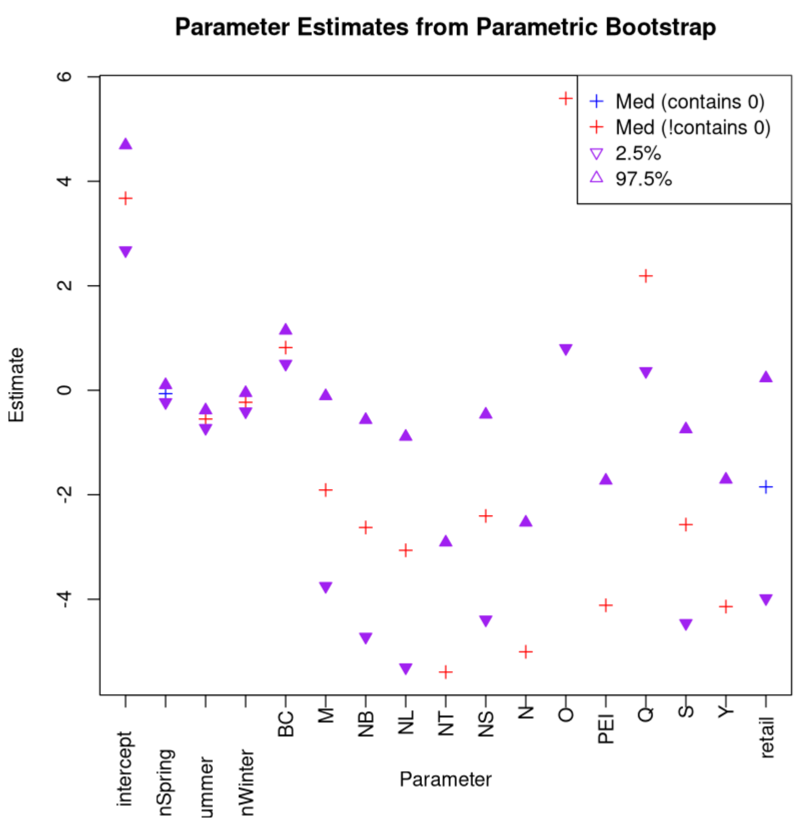
In [ ]:
```
print(xtable(P_results[, -c(6,8)]), type = "latex", file = "param.tex")
```

In [ ]:
```
# Plot the median
palette(c("red", "blue"))
plot(P_results$`50%`, pch=3, type = "p", col = P_results$col, xlab = "Parameter", ylab = "Estimate", main = "Parame

# Add custom x-axis with row names as labels
axis(1, at = 1:nrow(P_results), labels = FALSE)
text(1:nrow(P_results), par("usr")[3] - 0.1, srt = 90, adj = 1.4, labels = rownames(P_results), xpd = TRUE)

# Add 2.5% and 97.5% rows with triangles
points(P_results$`2.5%`, type = "p", pch = 25, bg="purple", col="purple") # Downward triangle for 2.5%
points(P_results$`97.5%`, type = "p", pch = 24, bg="purple", col="purple") # Upward triangle for 97.5%

# Add a legend
legend("topright", legend = c("Med (contains 0)", "Med (!contains 0)", "2.5%", "97.5%"),
    col = c("blue", "red", "purple", "purple"), pch = c(3, 3, 25, 24), lty = 0)
```



Smooth Bootstrap

In [ ]:
```
S_data <- split_data$S
```

In [ ]:
```
alpha = 0.05
```

```r
S_mean_and_sd <- apply(X=S_data[2:ncol(S_data)], MARGIN = 2, FUN = function(x) mean_and_sd(x))
S_intervals <- apply(X=S_data[2:ncol(S_data)], MARGIN = 2, FUN = function(x) interval(x, alpha))

S_results <- rbind(S_mean_and_sd,
                   S_intervals)

rownames(S_results)[1:2] <- c("mean", "sd")

S_results <- t(S_results)

contains <- sapply(1:nrow(S_results), function(i) {
    (S_results[i, "2.5%"] <= 0) & (S_results[i, "97.5%"] >= 0)
})

S_results <- as.data.frame(S_results)
S_results$contains <- contains
S_results$sig <- !contains


S_results$col <- ifelse(S_results$contains == TRUE, 2, 1)
S_results$abs_Z <- S_results$mean / S_results$sd

S_results[, 1:ncol(S_results)-1]
```

A data.frame: 17 × 8

	mean	sd	2.5%	50%	97.5%	contains	sig	col
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<lgl>	<dbl>
intercept	3.0032167	0.211839080	2.5785509	3.00613192	3.41745348	FALSE	TRUE	1
seasonSpring	-0.0432973	0.009423530	-0.0657243	-0.04139669	-0.02813229	FALSE	TRUE	1
seasonSummer	-0.5488623	0.006840469	-0.5649198	-0.54816059	-0.53537315	FALSE	TRUE	1
seasonWinter	-0.1984346	0.011935837	-0.2268721	-0.19627501	-0.17925248	FALSE	TRUE	1
BC	0.6665134	0.047726253	0.5721851	0.66546257	0.76338312	FALSE	TRUE	1
M	-0.6717034	0.387893635	-1.4323908	-0.67648953	0.10359043	TRUE	FALSE	2
NB	-1.2128494	0.440162662	-2.0806681	-1.21694261	-0.32875168	FALSE	TRUE	1
NL	-1.5647129	0.468847487	-2.4853613	-1.57066552	-0.62292596	FALSE	TRUE	1
NT	-3.6764259	0.524415788	-4.7039603	-3.68370460	-2.62777800	FALSE	TRUE	1
NS	-1.0724642	0.417827800	-1.8940387	-1.07733113	-0.23237239	FALSE	TRUE	1
N	-3.2969437	0.526169874	-4.3305003	-3.30283072	-2.24253079	FALSE	TRUE	1
O	2.2357463	1.047170530	0.1371099	2.24835685	4.28994263	FALSE	TRUE	1
PEI	-2.4742033	0.510562465	-3.4806158	-2.48037349	-1.45084004	FALSE	TRUE	1
Q	0.9184694	0.396391911	0.1251399	0.92385777	1.69348049	FALSE	TRUE	1
S	-1.3130474	0.395640963	-2.0942263	-1.31717742	-0.51636677	FALSE	TRUE	1
Y	-2.4598483	0.523237150	-3.4876671	-2.46581953	-1.41114819	FALSE	TRUE	1
retail	-0.3948373	0.454406930	-1.2904557	-0.40028423	0.51938701	TRUE	FALSE	2

In [ ]:

```r
print(xtable(S_results[, -c(6,8)]), type = "latex", file = "smooth.tex")
```

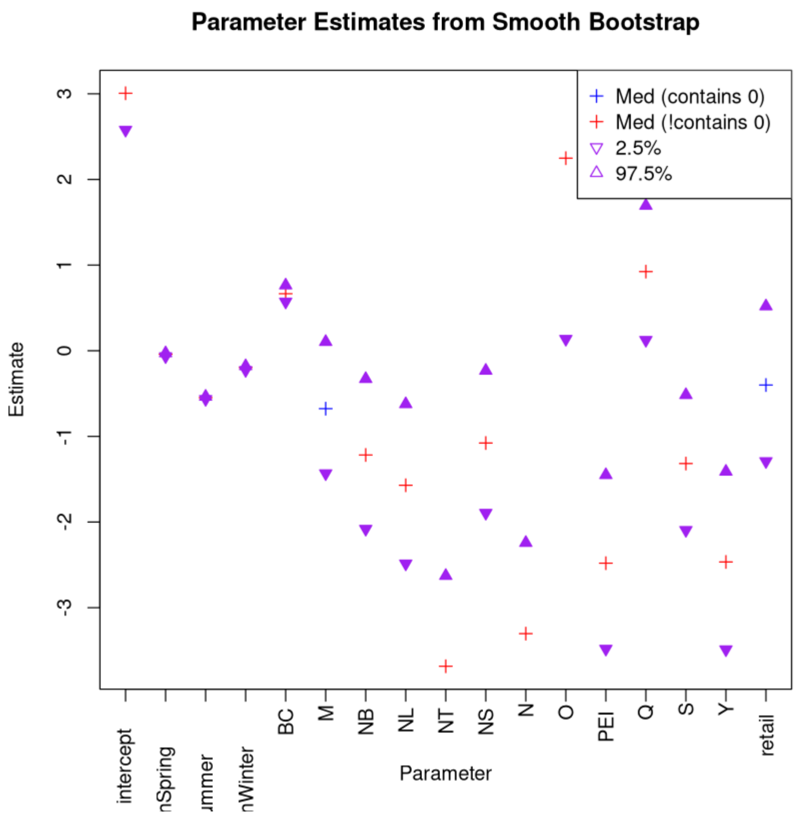
In [ ]:

```r
# Plot the median
palette(c("red", "blue"))
plot(S_results$`50%`, pch=3, type = "p", col = S_results$col, xlab = "Parameter", ylab = "Estimate", main = "Parame

# Add custom x-axis with row names as labels
axis(1, at = 1:nrow(S_results), labels = FALSE)
text(1:nrow(S_results), par("usr")[3] - 0.1, srt = 90, adj = 1.4, labels = rownames(S_results), xpd = TRUE)

# Add 2.5% and 97.5% rows with triangles
points(S_results$`2.5%`, type = "p", pch = 25, bg="purple", col="purple") # Downward triangle for 2.5%
points(S_results$`97.5%`, type = "p", pch = 24, bg="purple", col="purple") # Upward triangle for 97.5%

# Add a legend
legend("topright", legend = c("Med (contains 0)", "Med (!contains 0)", "2.5%", "97.5%"),
    col = c("blue", "red", "purple", "purple"), pch = c(3, 3, 25, 24), lty = 0)
```



Error-Resampling Bootstrap

In [ ]:
```
E_data <- split_data$E
head(E_data)
```

A data.frame: 6 × 18

	type	intercept	seasonSpring	seasonSummer	seasonWinter	BC	M	NB	NL	NT	NS
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
30001	E	3.699515	-0.04939341	-0.5644029	-0.2540712	0.7970201	-2.006120	-2.671376	-3.064948	-5.488898	-2.425398
30002	E	3.625176	-0.05852822	-0.5528195	-0.2371354	0.8058068	-1.841590	-2.525398	-2.934343	-4.962600	-2.384595
30003	E	3.709236	-0.07304630	-0.5725859	-0.2355005	0.8071427	-1.939634	-2.735903	-3.228107	-5.071177	-2.461993
30004	E	3.616712	-0.05696503	-0.5356797	-0.2357946	0.8259597	-1.809953	-2.501839	-3.113147	-5.306339	-2.372605
30005	E	3.627947	-0.06911946	-0.5413708	-0.2521120	0.8041241	-1.844577	-2.525781	-3.042427	-4.808079	-2.368481
30006	E	3.652942	-0.05582750	-0.5327621	-0.2334410	0.8185922	-1.896705	-2.601671	-3.046191	-5.058875	-2.463401

In [ ]:
```
alpha = 0.05

E_mean_and_sd <- apply(X=E_data[2:ncol(E_data)], MARGIN = 2, FUN = function(x) mean_and_sd(x))
E_intervals <- apply(X=E_data[2:ncol(E_data)], MARGIN = 2, FUN = function(x) interval(x, alpha))

E_results <- rbind(E_mean_and_sd,
                   E_intervals)

rownames(E_results)[1:2] <- c("mean", "sd")

E_results <- t(E_results)

contains <- sapply(1:nrow(E_results), function(i) {
    (E_results[i, "2.5%"] <= 0) & (E_results[i, "97.5%"] >= 0)
})

E_results <- as.data.frame(E_results)
E_results$contains <- contains
E_results$sig <- !contains

E_results$col <- ifelse(E_results$contains == TRUE, 2, 1)

E_results[, 1:ncol(E_results)-1]

E_results$abs_Z <- E_results$mean / E_results$sd
```

A data.frame: 17 × 7

	mean	sd	2.5%	50%	97.5%	contains	sig
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<lgl>
intercept	3.65907931	0.05232609	3.55597981	3.65889011	3.76098758	FALSE	TRUE
seasonSpring	-0.06686249	0.01206230	-0.09069678	-0.06694102	-0.04367337	FALSE	TRUE
seasonSummer	-0.56164712	0.01631474	-0.59371407	-0.56176815	-0.52947010	FALSE	TRUE
seasonWinter	-0.24188696	0.01411872	-0.26954937	-0.24184874	-0.21472836	FALSE	TRUE
BC	0.81978632	0.02120434	0.77836319	0.81988192	0.86149735	FALSE	TRUE
M	-1.89277538	0.09310281	-2.07349089	-1.89171302	-1.70922080	FALSE	TRUE
NB	-2.61137323	0.10887839	-2.82858549	-2.61084232	-2.39847612	FALSE	TRUE
NL	-3.06541203	0.12135411	-3.29789921	-3.06543911	-2.82500134	FALSE	TRUE
NT	-5.19838895	0.28099973	-5.79723068	-5.17886681	-4.70356729	FALSE	TRUE
NS	-2.39774194	0.10319199	-2.60090312	-2.39562149	-2.19665406	FALSE	TRUE
N	-4.96495805	0.25442124	-5.49613999	-4.94864522	-4.51231450	FALSE	TRUE
O	5.54232967	0.24100089	5.06586437	5.54061719	6.01579804	FALSE	TRUE
PEI	-4.15854853	0.17505125	-4.51275822	-4.15337366	-3.82661703	FALSE	TRUE
Q	2.17329766	0.09249328	1.99000881	2.17234989	2.35457061	FALSE	TRUE
S	-2.57567031	0.10171068	-2.77510764	-2.57459895	-2.37636350	FALSE	TRUE
Y	-4.18062240	0.17282377	-4.53605214	-4.17558619	-3.84641487	FALSE	TRUE
retail	-1.82297973	0.10469870	-2.02791516	-1.82179096	-1.61589834	FALSE	TRUE

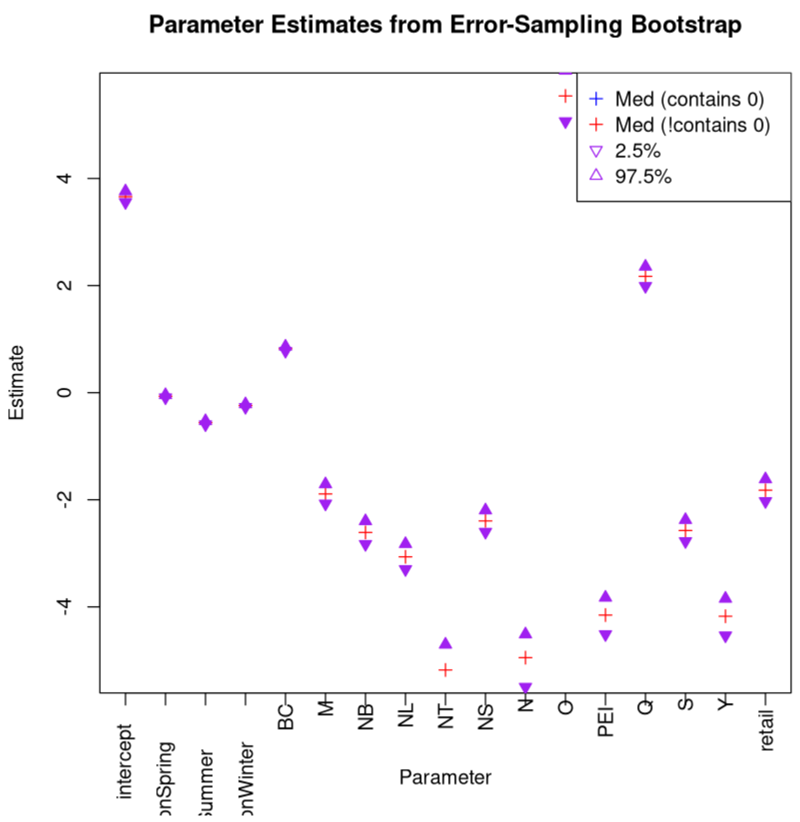
In [ ]:
```
print(xtable(E_results[, -c(6,8)]), type = "latex", file = "error.tex")
```

In [ ]:
```
# Plot the median
palette(c("red", "blue"))
plot(E_results$`50%`, pch=3, type = "p", col = E_results$col, xlab = "Parameter", ylab = "Estimate", main = "Parame

# Add custom x-axis with row names as labels
axis(1, at = 1:nrow(E_results), labels = FALSE)
text(1:nrow(E_results), par("usr")[3], srt = 90, adj = 1.4, labels = rownames(E_results), xpd = TRUE)

# Add 2.5% and 97.5% rows with triangles
points(E_results$`2.5%`, type = "p", pch = 25, bg="purple", col="purple") # Downward triangle for 2.5%
points(E_results$`97.5%`, type = "p", pch = 24, bg="purple", col="purple") # Upward triangle for 97.5%

# Add a legend
legend("topright", legend = c("Med (contains 0)", "Med (!contains 0)", "2.5%", "97.5%"),
    col = c("blue", "red", "purple", "purple"), pch = c(3, 3, 25, 24), lty = 0)
```



Concatenate Results

In [ ]:
```
all_results <- rbind(R_results, P_results, S_results, E_results)
params <- rownames(R_results)

labels <- c(rep("R", nrow(R_results)), rep("P", nrow(P_results)), rep("S", nrow(S_results)), rep("E", nrow(E_result
```

```
all_results$type <- labels
all_results[, c("contains", "col")] <- NULL
all_results$param <- rep(params, 4)
all_results$type <- as.factor(all_results$type)
rownames(all_results) <- NULL
head(all_results)
```

A data.frame: 6 × 9

	mean	sd	2.5%	50%	97.5%	sig	abs_Z	type	param
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<dbl>	<fct>	<chr>
1	3.66228302	0.52607625	2.6975978	3.63694632	4.77025453	TRUE	6.961506	R	intercept
2	-0.06441936	0.08027795	-0.2257916	-0.06331621	0.09047168	FALSE	-0.802454	R	seasonSpring
3	-0.55129740	0.08485389	-0.7191962	-0.54987054	-0.38350957	TRUE	-6.497020	R	seasonSummer
4	-0.23505052	0.09606068	-0.4236374	-0.23491057	-0.04595924	TRUE	-2.446896	R	seasonWinter
5	0.81861051	0.13647920	0.5470062	0.81865706	1.08664175	TRUE	5.998061	R	BC
6	-1.89181259	0.92979598	-3.8450863	-1.84755375	-0.18400619	TRUE	-2.034653	R	M

In [ ]:
```
all_results$Z <- all_results$mean / all_results$sd

head(all_results)
all_results <- all_results[, c(-3:-1)]
res <- all_results[order(-abs(all_results$Z)), ]
split(res, f=type)
```

A data.frame: 6 × 10

	mean	sd	2.5%	50%	97.5%	sig	abs_Z	type	param	Z
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<dbl>	<fct>	<chr>	<dbl>
1	3.66228302	0.52607625	2.6975978	3.63694632	4.77025453	TRUE	6.961506	R	intercept	6.961506
2	-0.06441936	0.08027795	-0.2257916	-0.06331621	0.09047168	FALSE	-0.802454	R	seasonSpring	-0.802454
3	-0.55129740	0.08485389	-0.7191962	-0.54987054	-0.38350957	TRUE	-6.497020	R	seasonSummer	-6.497020
4	-0.23505052	0.09606068	-0.4236374	-0.23491057	-0.04595924	TRUE	-2.446896	R	seasonWinter	-2.446896
5	0.81861051	0.13647920	0.5470062	0.81865706	1.08664175	TRUE	5.998061	R	BC	5.998061
6	-1.89181259	0.92979598	-3.8450863	-1.84755375	-0.18400619	TRUE	-2.034653	R	M	-2.034653

```
Warning message in split.default(x = seq_len(nrow(x)), f = f, drop = drop, ...):
“data length is not a multiple of split variable”
```

$E

A data.frame: 0 × 7

50%	97.5%	sig	abs_Z	type	param	Z
<dbl>	<dbl>	<lgl>	<dbl>	<fct>	<chr>	<dbl>

$P

A data.frame: 0 × 7

50%	97.5%	sig	abs_Z	type	param	Z
<dbl>	<dbl>	<lgl>	<dbl>	<fct>	<chr>	<dbl>

$R

A data.frame: 68 × 7

	50%	97.5%	sig	abs_Z	type	param	Z
	<dbl>	<dbl>	<lgl>	<dbl>	<fct>	<chr>	<dbl>
37	-0.54816059	-0.53537315	TRUE	-80.237528	S	seasonSummer	-80.237528
52	3.65889011	3.76098758	TRUE	69.928386	E	intercept	69.928386
56	0.81988192	0.86149735	TRUE	38.661247	E	BC	38.661247
54	-0.56176815	-0.52947010	TRUE	-34.425737	E	seasonSummer	-34.425737
66	-2.57459895	-2.37636350	TRUE	-25.323500	E	S	-25.323500
59	-3.06543911	-2.82500134	TRUE	-25.260060	E	NL	-25.260060
67	-4.17558619	-3.84641487	TRUE	-24.190089	E	Y	-24.190089
58	-2.61084232	-2.39847612	TRUE	-23.984313	E	NB	-23.984313
64	-4.15337366	-3.82661703	TRUE	-23.756178	E	PEI	-23.756178
65	2.17234989	2.35457061	TRUE	23.496816	E	Q	23.496816
61	-2.39562149	-2.19665406	TRUE	-23.235737	E	NS	-23.235737
63	5.54061719	6.01579804	TRUE	22.997134	E	O	22.997134
57	-1.89171302	-1.70922080	TRUE	-20.329949	E	M	-20.329949
62	-4.94864522	-4.51231450	TRUE	-19.514715	E	N	-19.514715
60	-5.17886681	-4.70356729	TRUE	-18.499623	E	NT	-18.499623
68	-1.82179096	-1.61589834	TRUE	-17.411675	E	retail	-17.411675
55	-0.24184874	-0.21472836	TRUE	-17.132352	E	seasonWinter	-17.132352
38	-0.19627501	-0.17925248	TRUE	-16.625106	S	seasonWinter	-16.625106