

EstComp-Tarea13

Ixchel G. Meza Chávez

3 de diciembre 2018

```
library(tidyverse)
library(readr)
library(here)

df <- read_csv(
  "https://archive.ics.uci.edu/ml/machine-learning-databases/00211/CommViolPredUnnormalizedData.txt",
  col_names = FALSE,
  na = "?"
)

names(df) <- read_table(
  here::here("nombres.txt"),
  col_names = FALSE
) %>%
  mutate(
    var_names = gsub(
      "(.*) (.*)",
      "\\1",
      X2
    )
  ) %>%
  pull(var_names) %>%
  make.names()

estados_regiones <- read_csv("estados_regiones") %>%
  select(State, Division)
print(estados_regiones %>% arrange(Division) ,n=51)
```

A tibble: 51 x 2

##	State	Division
##	<chr>	<chr>
## 1	Illinois	East North Central
## 2	Indiana	East North Central
## 3	Michigan	East North Central
## 4	Ohio	East North Central
## 5	Wisconsin	East North Central
## 6	Alabama	East South Central
## 7	Kentucky	East South Central
## 8	Mississippi	East South Central
## 9	Tennessee	East South Central
## 10	New Jersey	Middle Atlantic
## 11	New York	Middle Atlantic
## 12	Pennsylvania	Middle Atlantic
## 13	Arizona	Mountain
## 14	Colorado	Mountain
## 15	Idaho	Mountain
## 16	Montana	Mountain

## 17 New Mexico	Mountain
## 18 Nevada	Mountain
## 19 Utah	Mountain
## 20 Wyoming	Mountain
## 21 Connecticut	New England
## 22 Massachusetts	New England
## 23 Maine	New England
## 24 New Hampshire	New England
## 25 Rhode Island	New England
## 26 Vermont	New England
## 27 Alaska	Pacific
## 28 California	Pacific
## 29 Hawaii	Pacific
## 30 Oregon	Pacific
## 31 Washington	Pacific
## 32 District of Columbia	South Atlantic
## 33 Delaware	South Atlantic
## 34 Florida	South Atlantic
## 35 Georgia	South Atlantic
## 36 Maryland	South Atlantic
## 37 North Carolina	South Atlantic
## 38 South Carolina	South Atlantic
## 39 Virginia	South Atlantic
## 40 West Virginia	South Atlantic
## 41 Iowa	West North Central
## 42 Kansas	West North Central
## 43 Minnesota	West North Central
## 44 Missouri	West North Central
## 45 North Dakota	West North Central
## 46 Nebraska	West North Central
## 47 South Dakota	West North Central
## 48 Arkansas	West South Central
## 49 Louisiana	West South Central
## 50 Oklahoma	West South Central
## 51 Texas	West South Central

```
estados <- read_csv("estados_regiones") %>%
  select(`State Code`, Division) %>%
  rename(State = `State Code`)
```

```
x <- df %>%
  left_join(estados, by = "State") %>%
  mutate(
    State = State %>% as.factor,
    Division = Division %>% as.factor
  ) %>%
  select(
    State,
    murders,
    pop,
    Division,
    pctBlack,
    pctWhite,
```

```

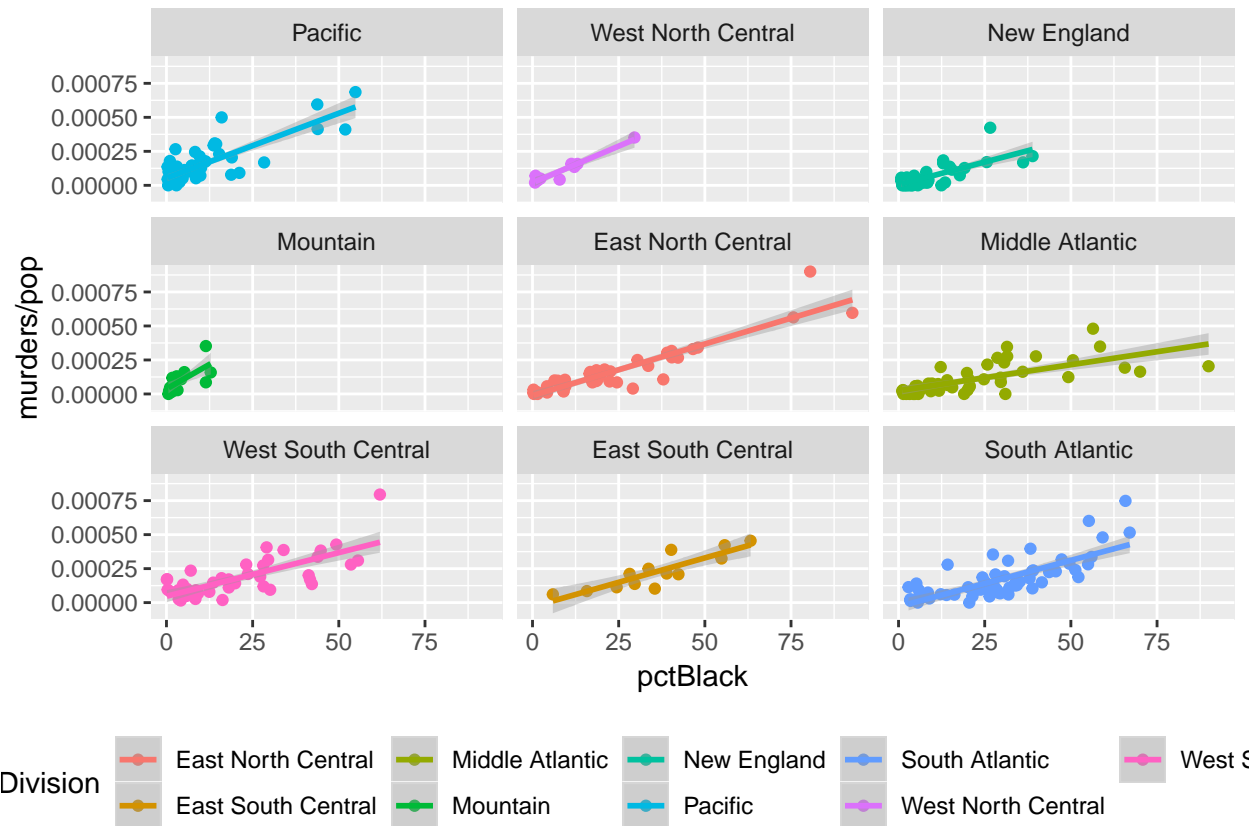
pctPoverty,
pct12.17w2Par,
pctNotSpeakEng,
pctBornStateResid,
# pctPolicWhite,
# pctPolicBlack,
# officDrugUnits
whitePerCap, #whitePerCap,
blackPerCap, #blackPerCap,
NAperCap, #indianPerCap,
asianPerCap, #AsianPerCap,
otherPerCap, #OtherPerCap,
hispPerCap, #HispPerCap,
pctNotHSgrad, #pctnotHSGrad,
pctLowEdu, #pctLess9thGrade,
pctWorkMom.6, #PctWorkMomYoungKids,
pctWorkMom.18, #PctWorkMom,
pctFgnImmig.3, #PctImmigRec10,
pctFgnImmig.5, #PctImmigRec10,
pctFgnImmig.8, #PctImmigRec10,
pctFgnImmig.10, #PctImmigRec10,
numDiffDrugsSeiz #NumKindsDrugsSeiz
) %>%
na.omit() %>%
arrange(match(Division, c("Pacific", "West North Central", "New England",
                          "Mountain", "East North Central", "Middle Atlantic",
                          "West South Central", "East South Central", "South Atlantic")) %>%
mutate(order = row_number())

x$Division %>% unique()

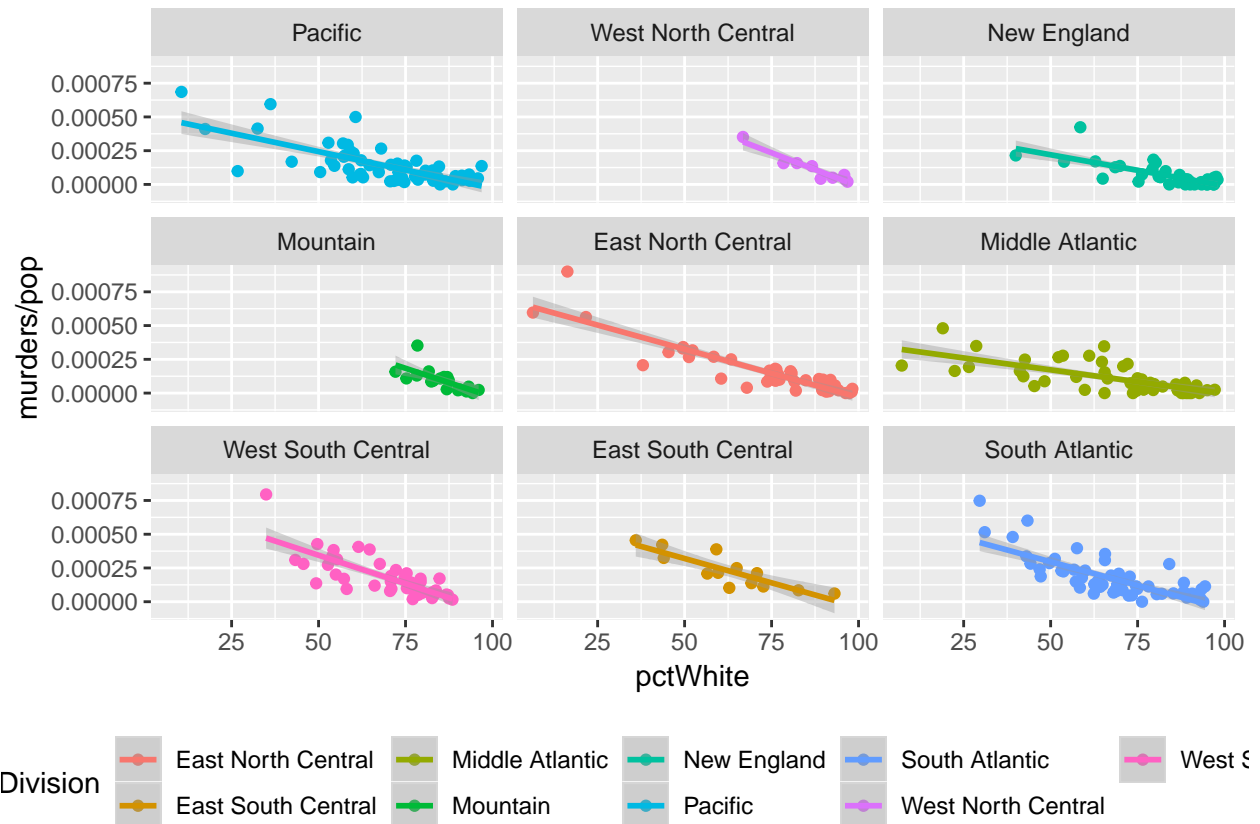
## [1] Pacific           West North Central New England
## [4] Mountain           East North Central Middle Atlantic
## [7] West South Central East South Central South Atlantic
## 9 Levels: East North Central East South Central ... West South Central

# racepctblack: percentage of population that is african american (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctBlack, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctBlack, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")

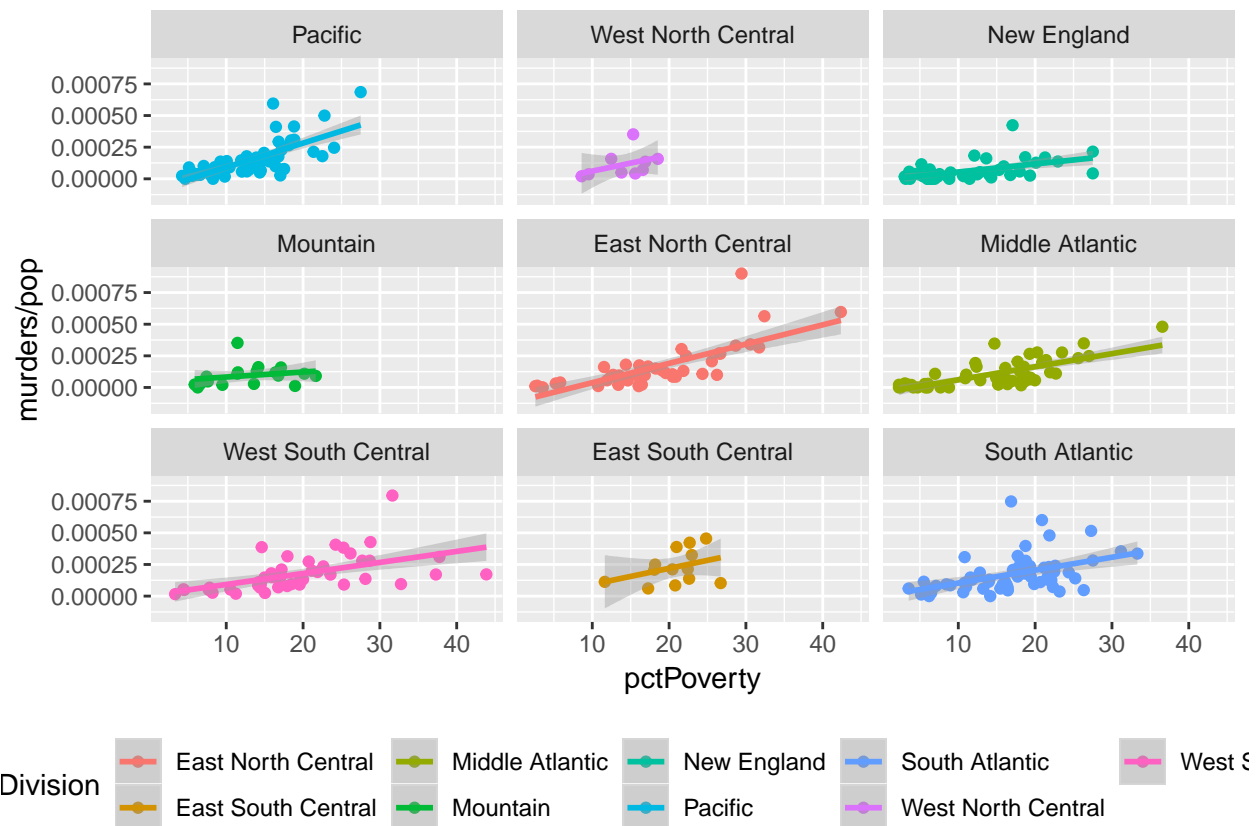
```



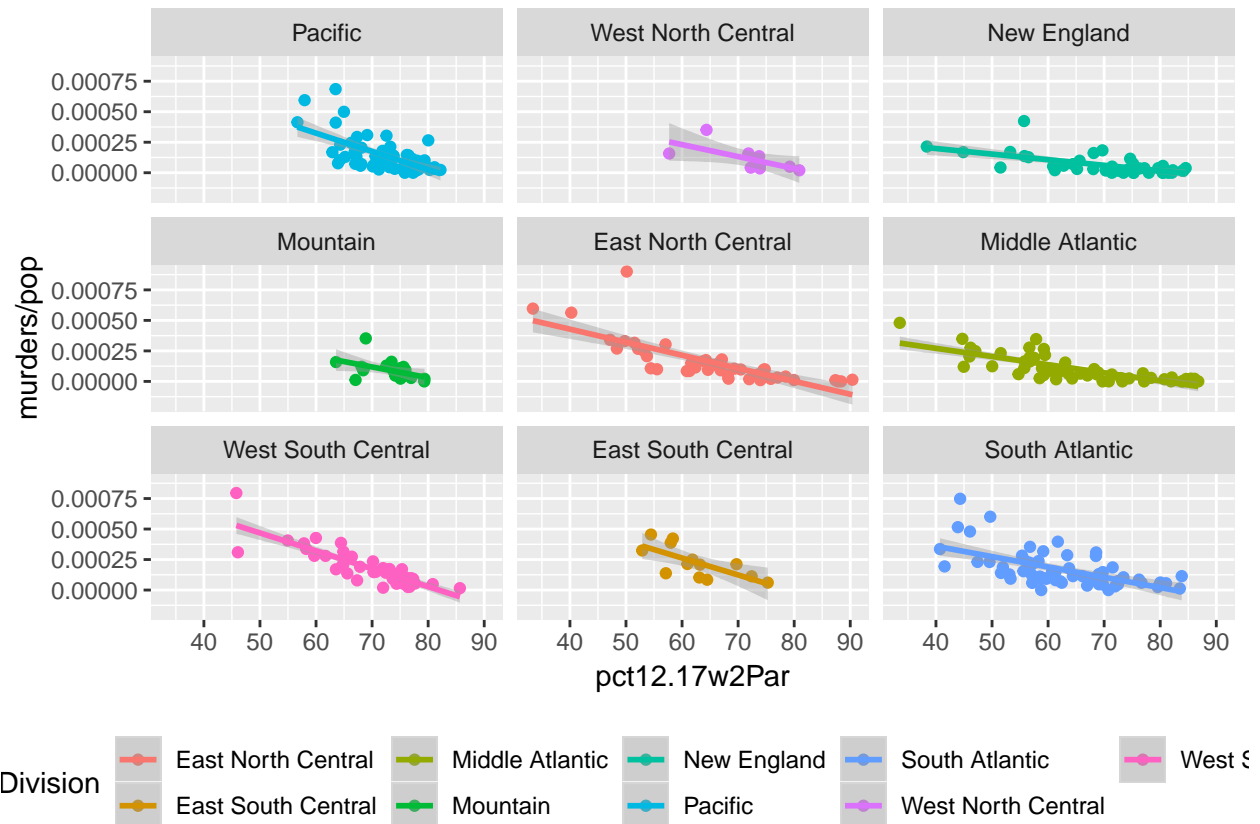
```
# racePctWhite: percentage of population that is caucasian (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctWhite, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctWhite, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```



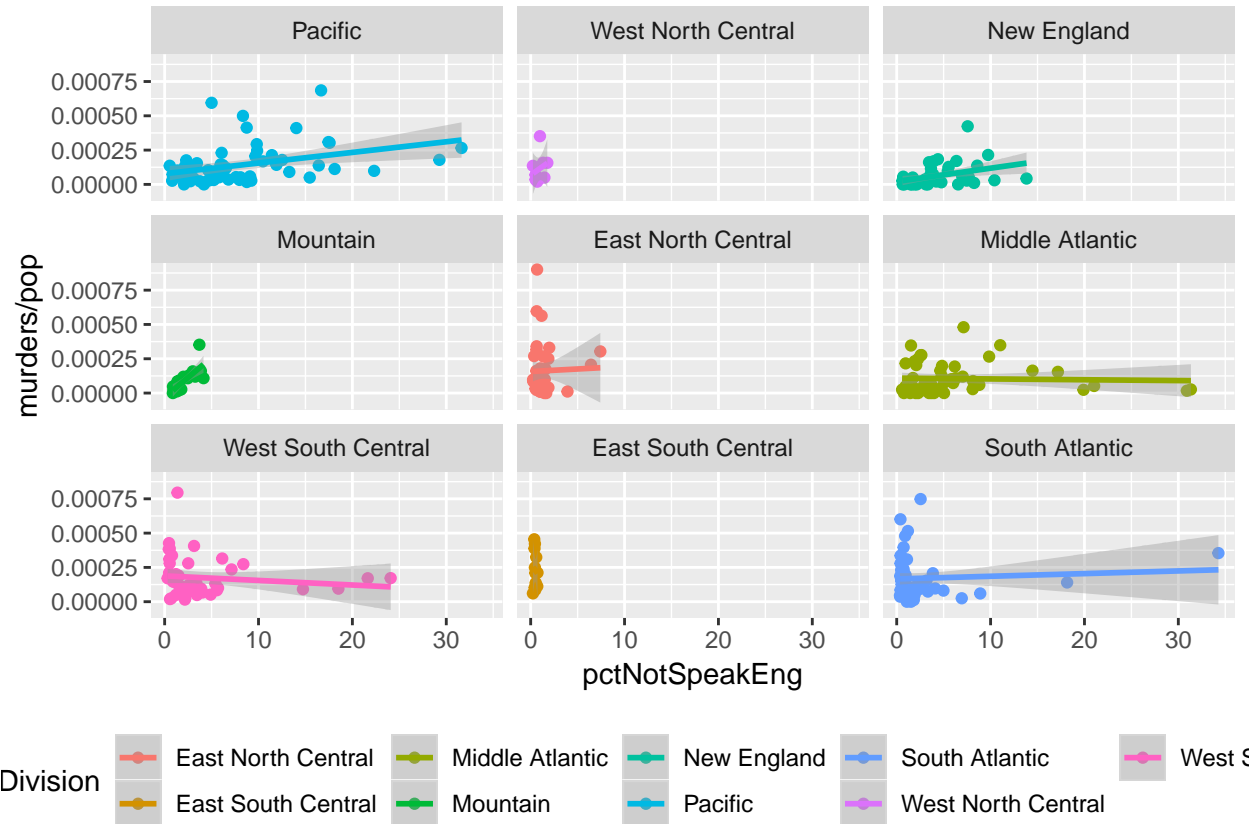
```
# PctPopUnderPov: percentage of people under the poverty level (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctPoverty, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctPoverty, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



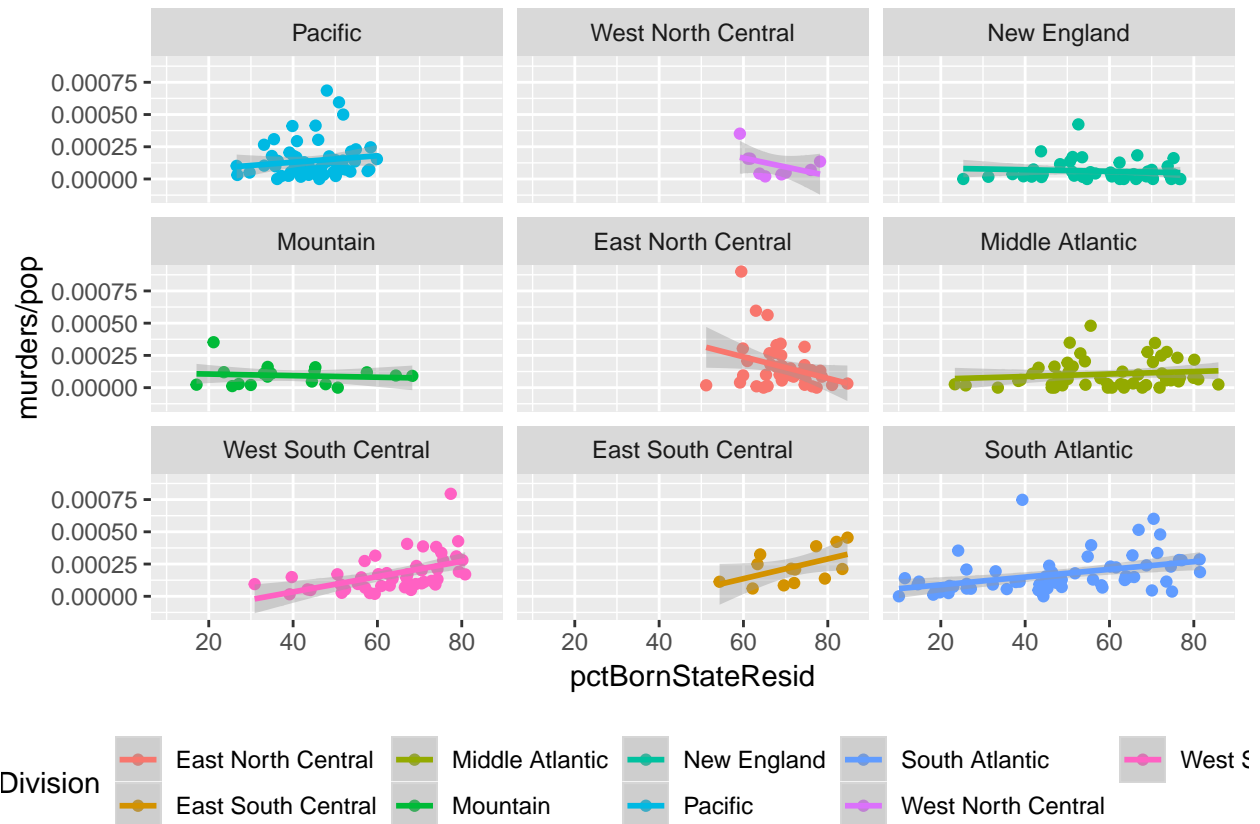
```
# PctTeen2Par: percent of kids age 12-17 in two parent households (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pct12.17w2Par, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pct12.17w2Par, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



```
# PctNotSpeakEnglWell: percent of people who do not speak English well (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctNotSpeakEng, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctNotSpeakEng, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```

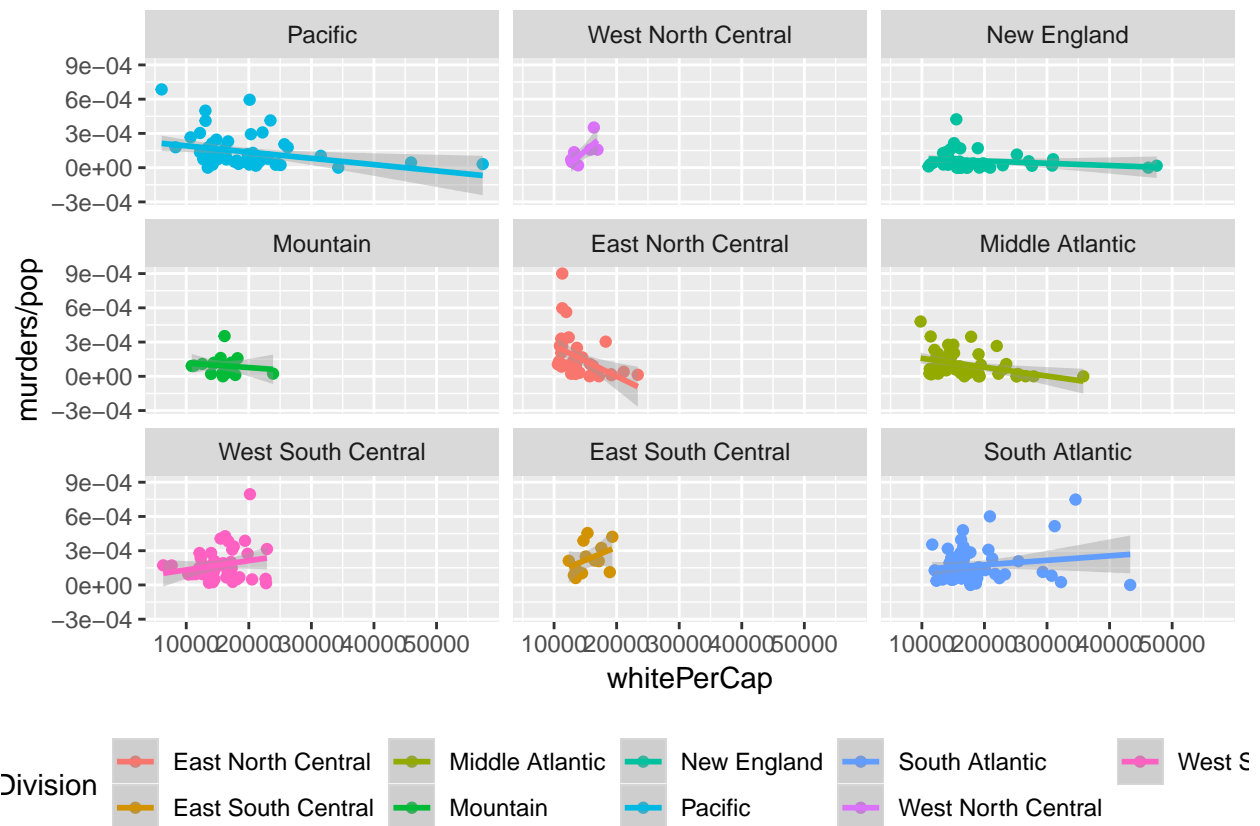


```
# PctBornSameState: percent of people born in the same state as currently living (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctBornStateResid, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctBornStateResid, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```

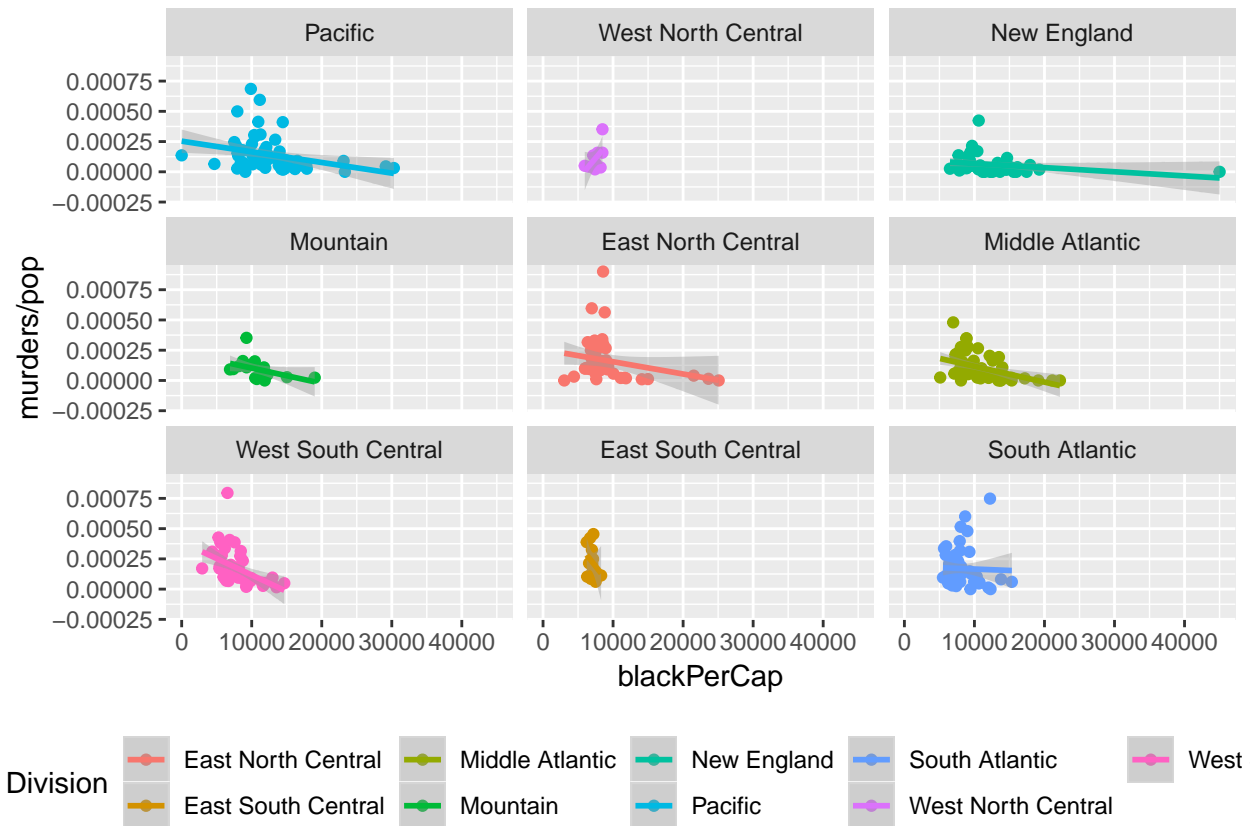



```
# Las que propuse

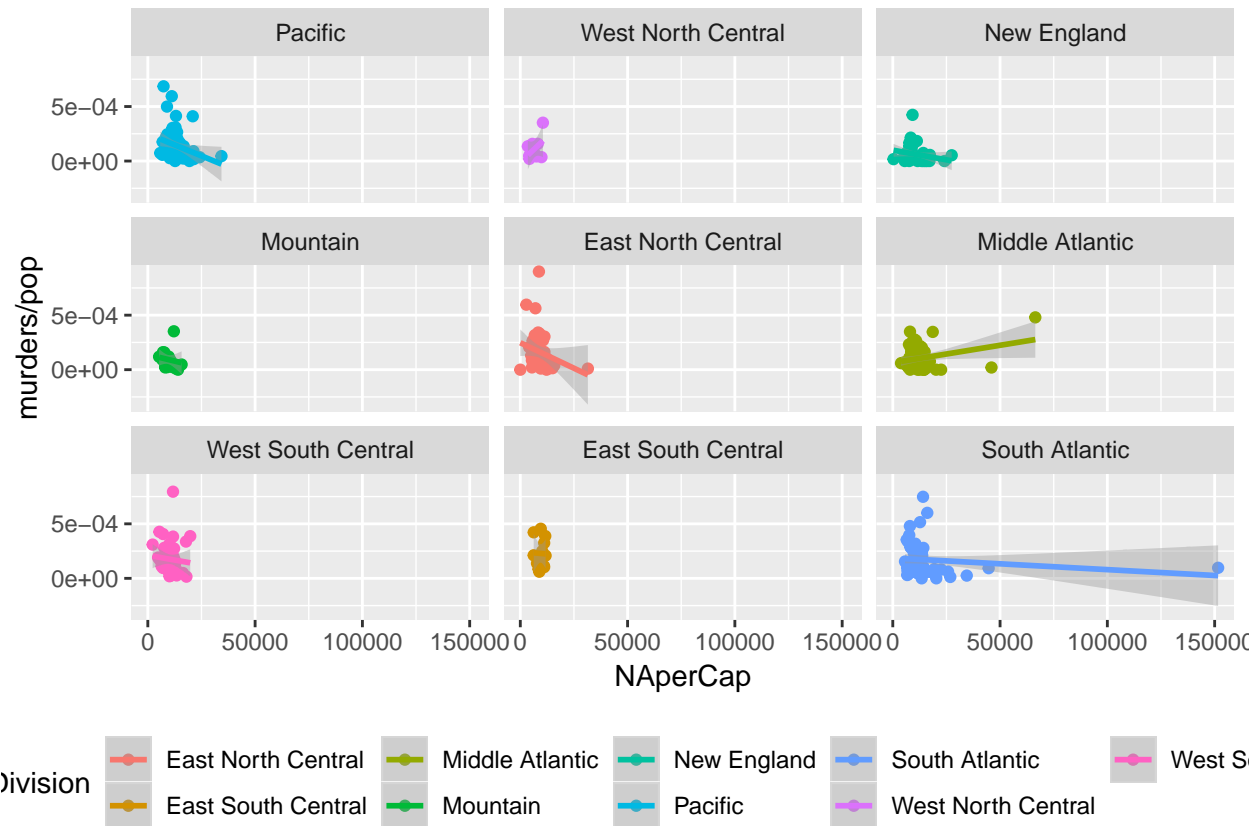
# whitePerCap: per capita income for caucasians (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=whitePerCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=whitePerCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



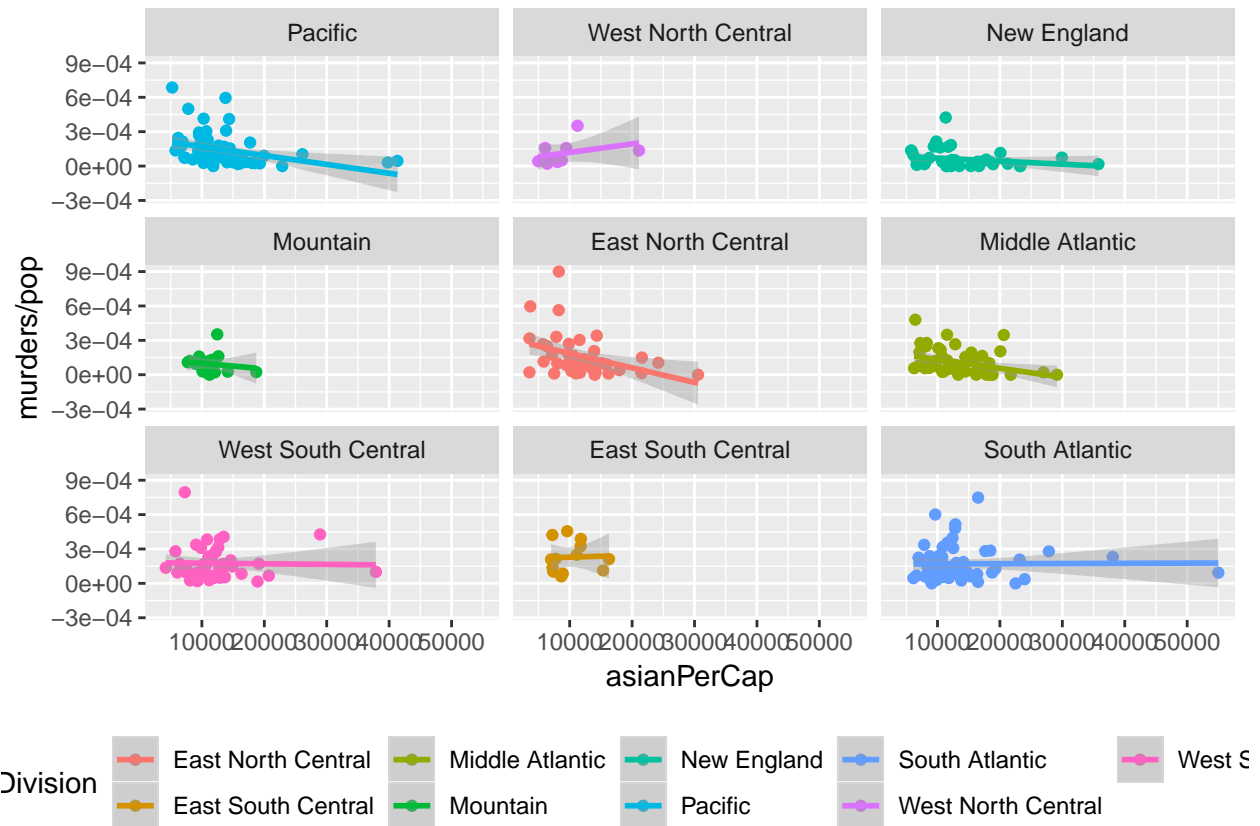
```
# blackPerCap: per capita income for african americans (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=blackPerCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=blackPerCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



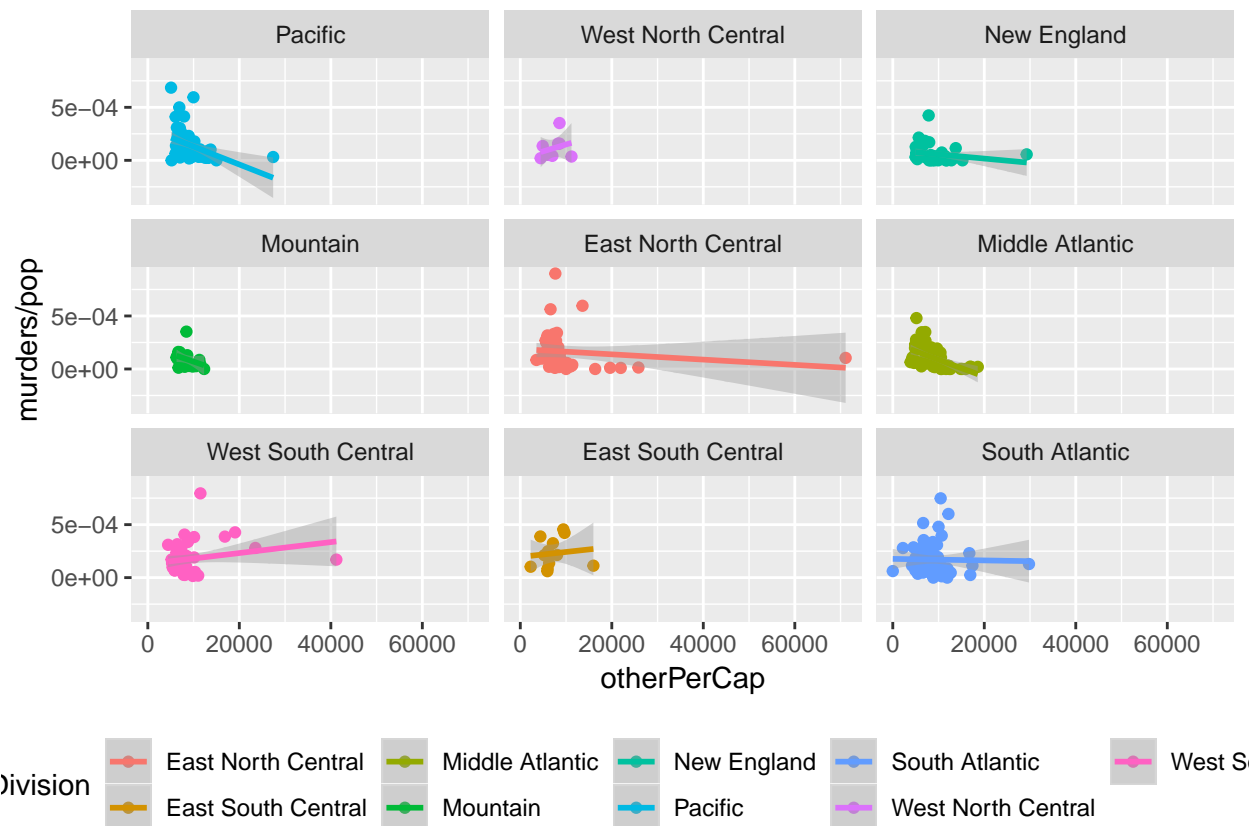
```
# indianPerCap: per capita income for native americans (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=NaperCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=NaperCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



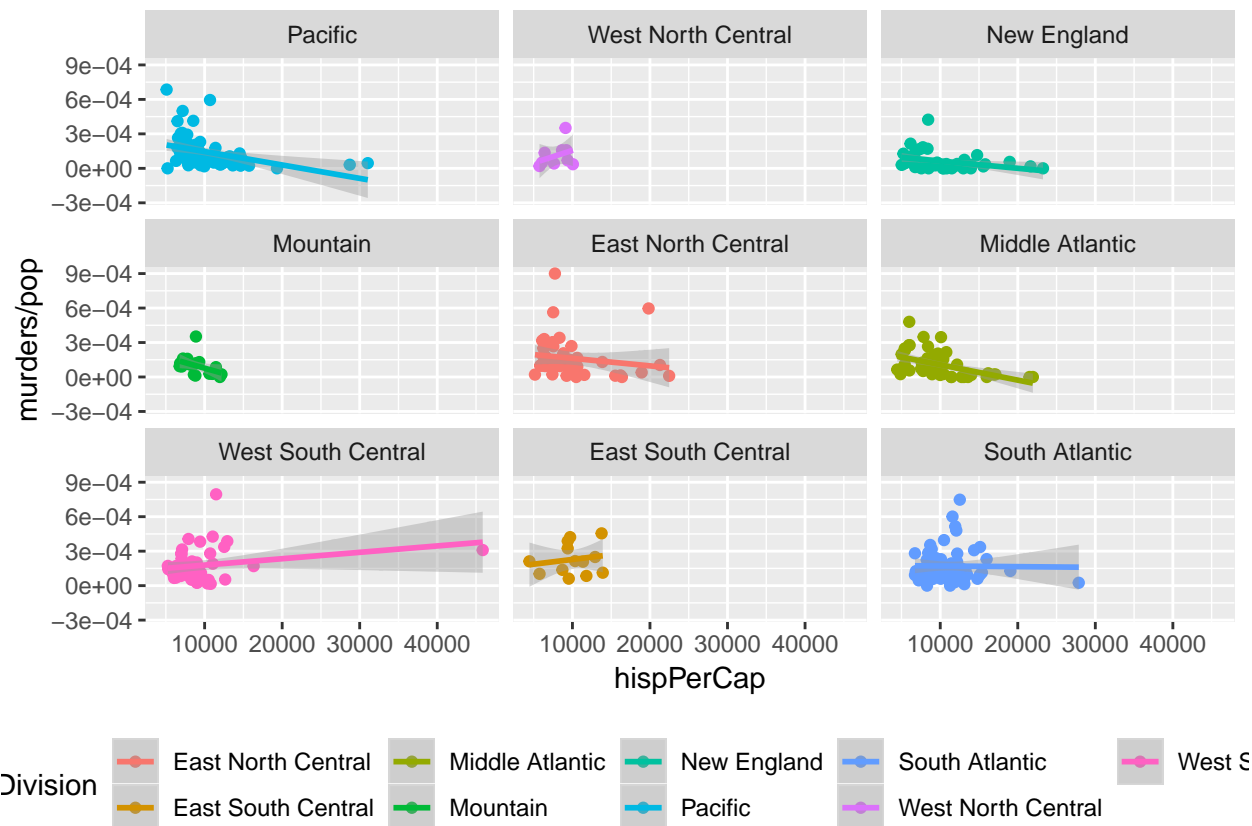
```
# AsianPerCap: per capita income for people with asian heritage (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=asianPerCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=asianPerCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



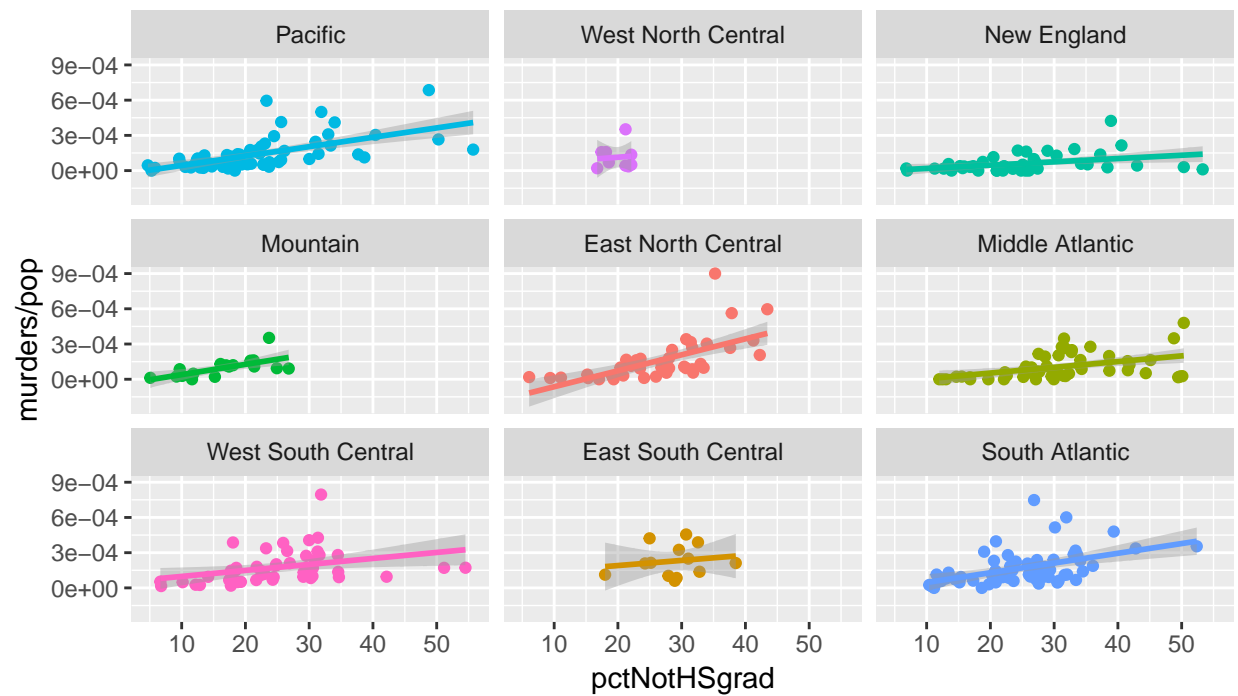
```
# OtherPerCap: per capita income for people with 'other' heritage (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=otherPerCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=otherPerCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```



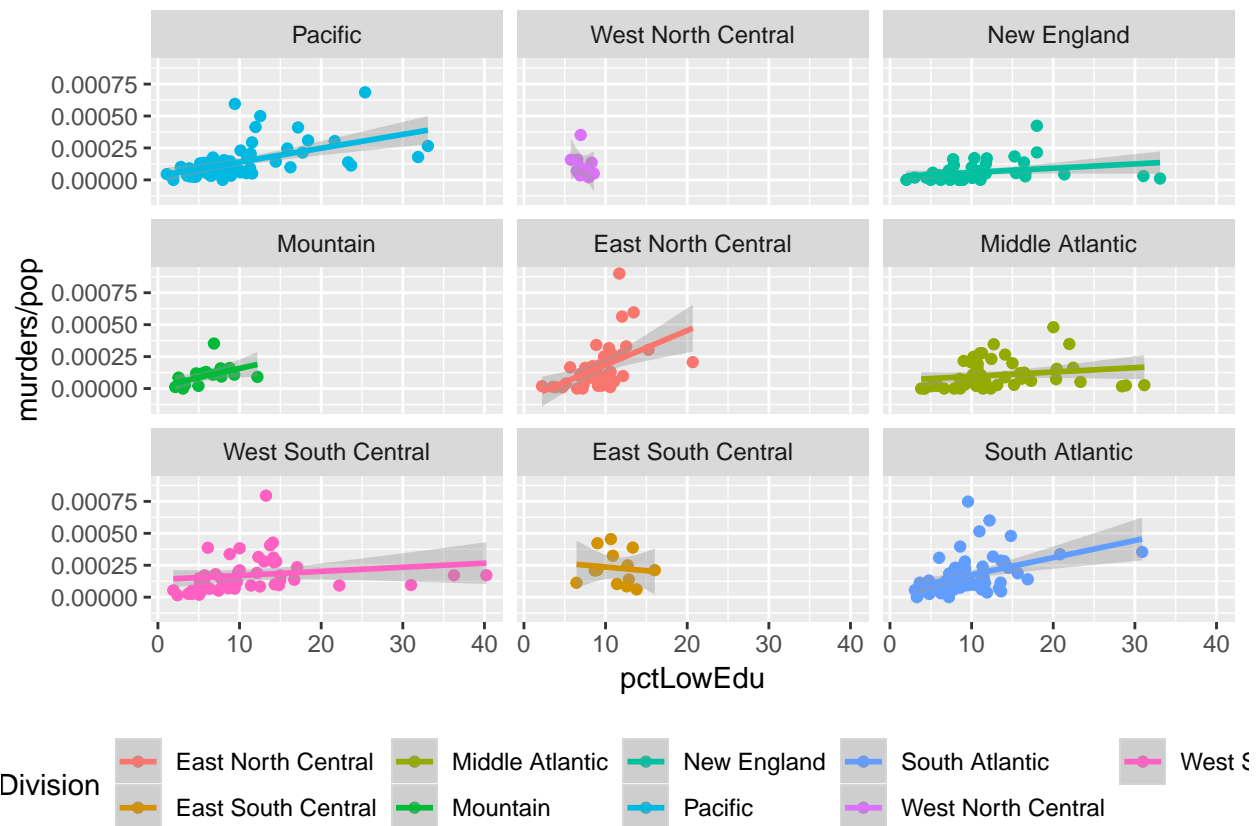
```
# HispPerCap: per capita income for people with hispanic heritage (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=hispPerCap, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=hispPerCap, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```



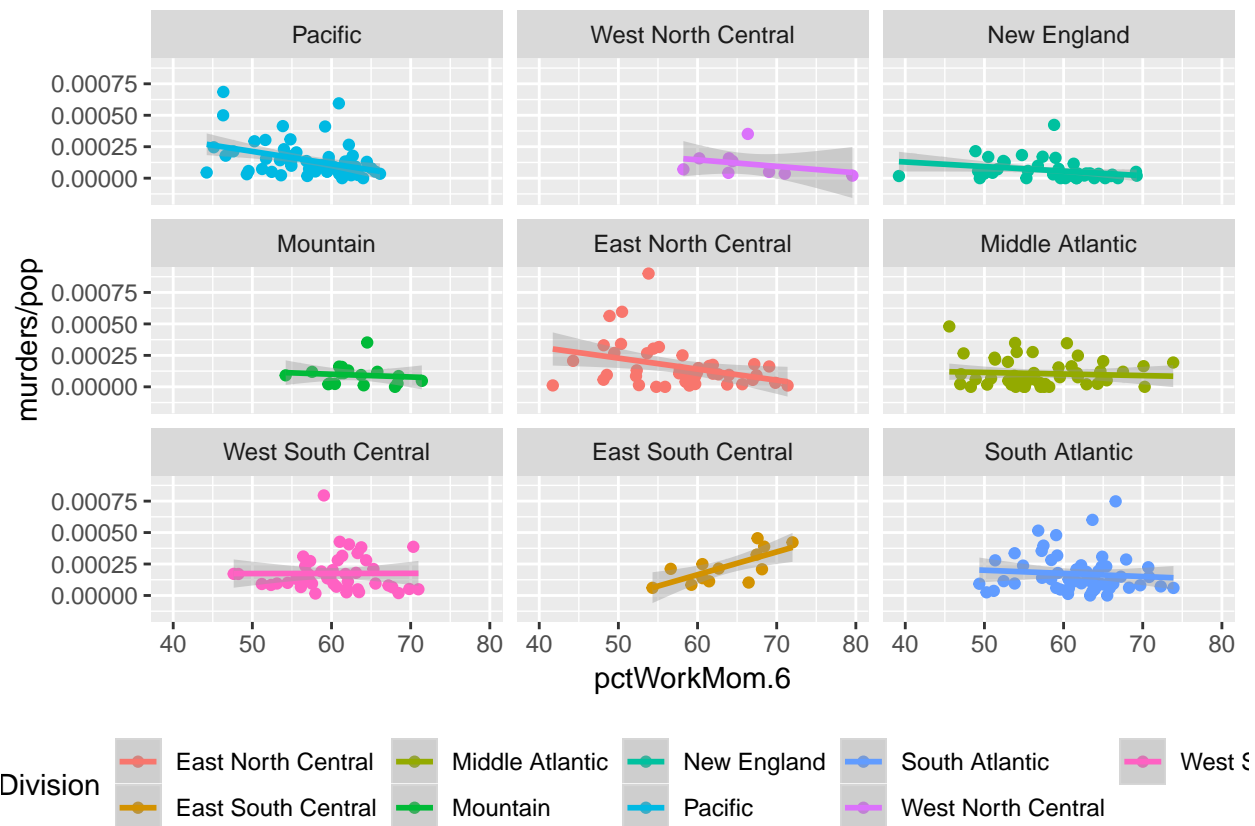
```
# PctNotHSGrad: percentage of people 25 and over that are not high school graduates (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctNotHSgrad, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctNotHSgrad, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```



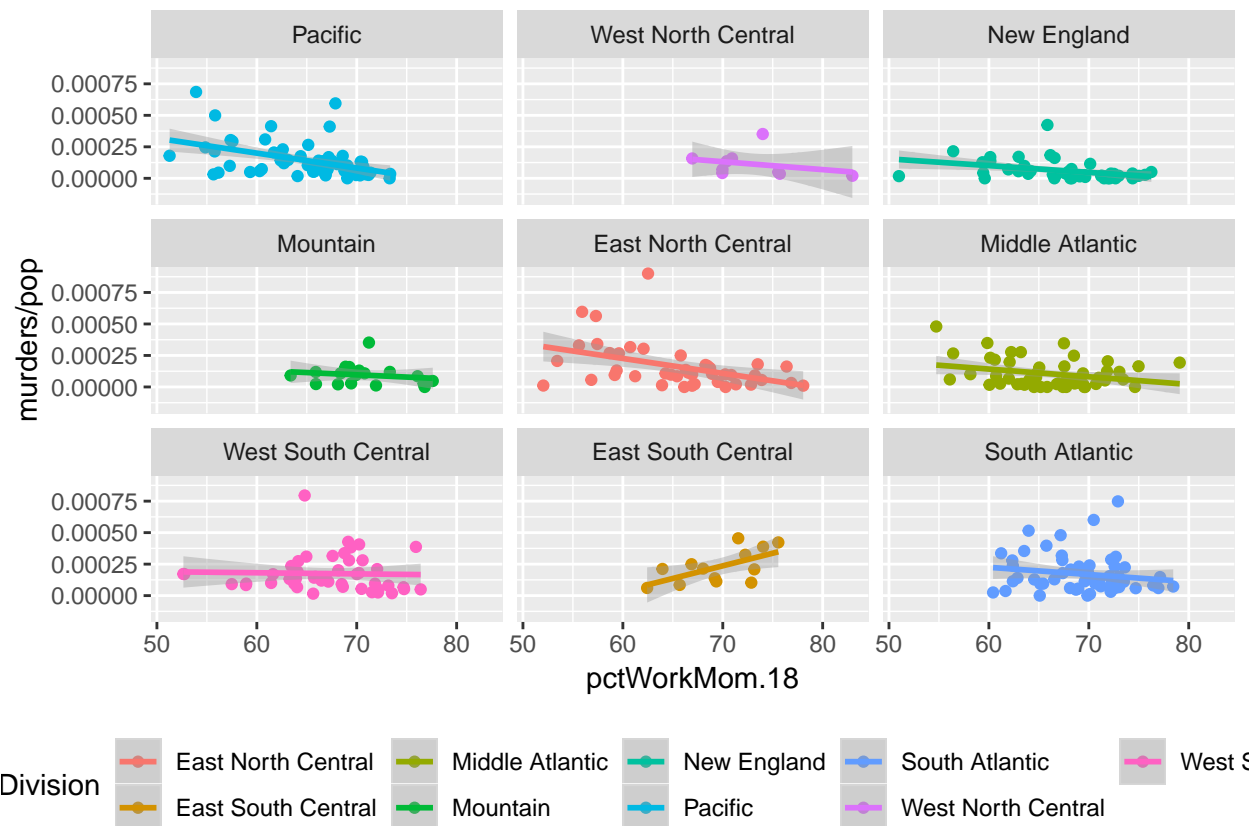
```
# PctLess9thGrade: percentage of people 25 and over with less than a 9th grade education (numeric - dec
ggplot(x) +
  geom_jitter(aes(x=pctLowEdu, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctLowEdu, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```

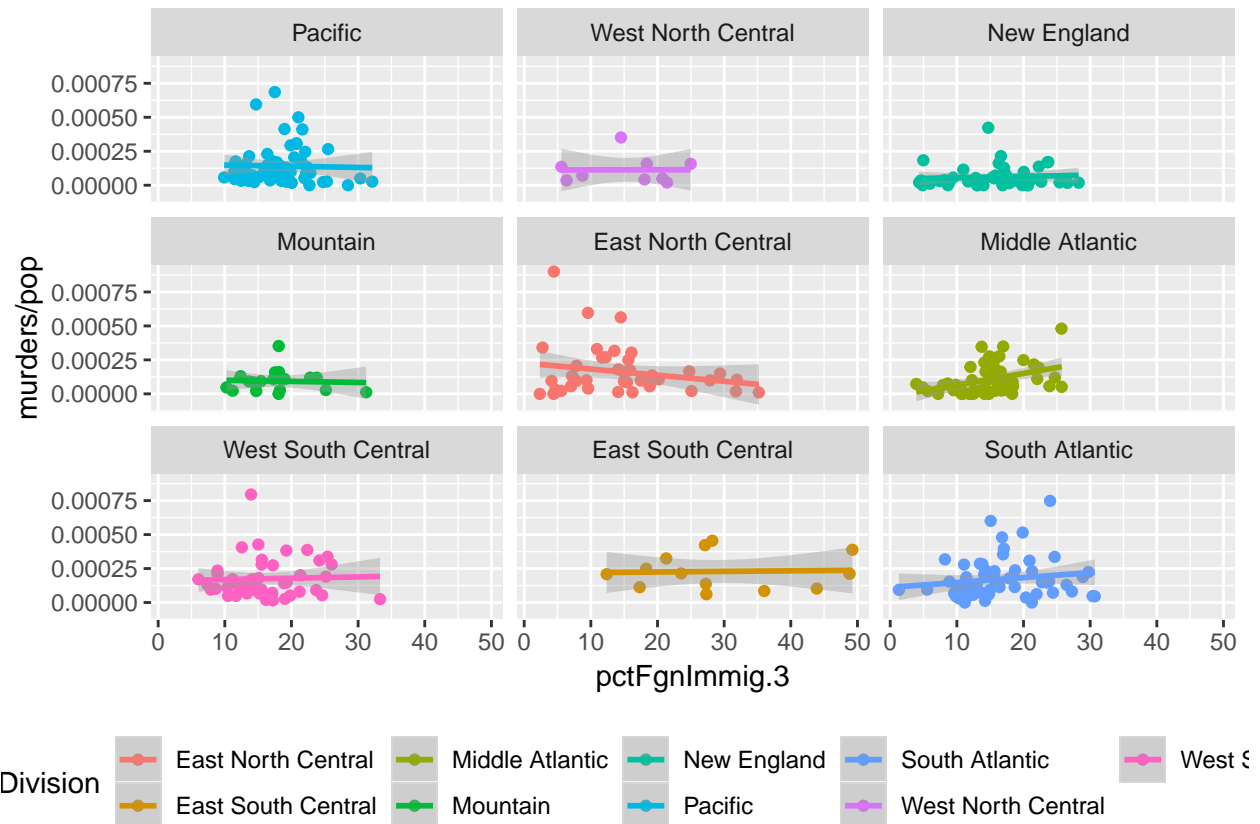
```
# PctWorkMomYoungKids: percentage of moms of kids 6 and under in labor force (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctWorkMom.6, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctWorkMom.6, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



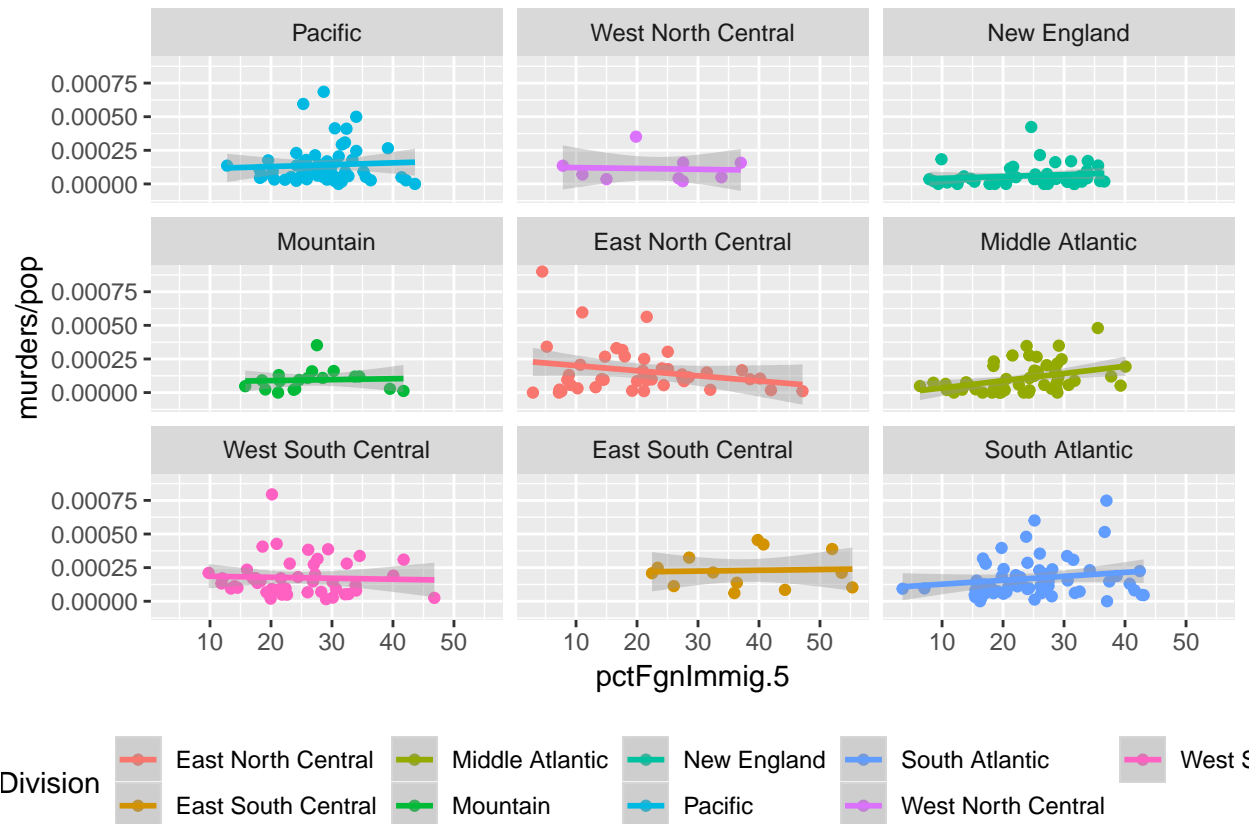
```
# PctWorkMom: percentage of moms of kids under 18 in labor force (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctWorkMom.18, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctWorkMom.18, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



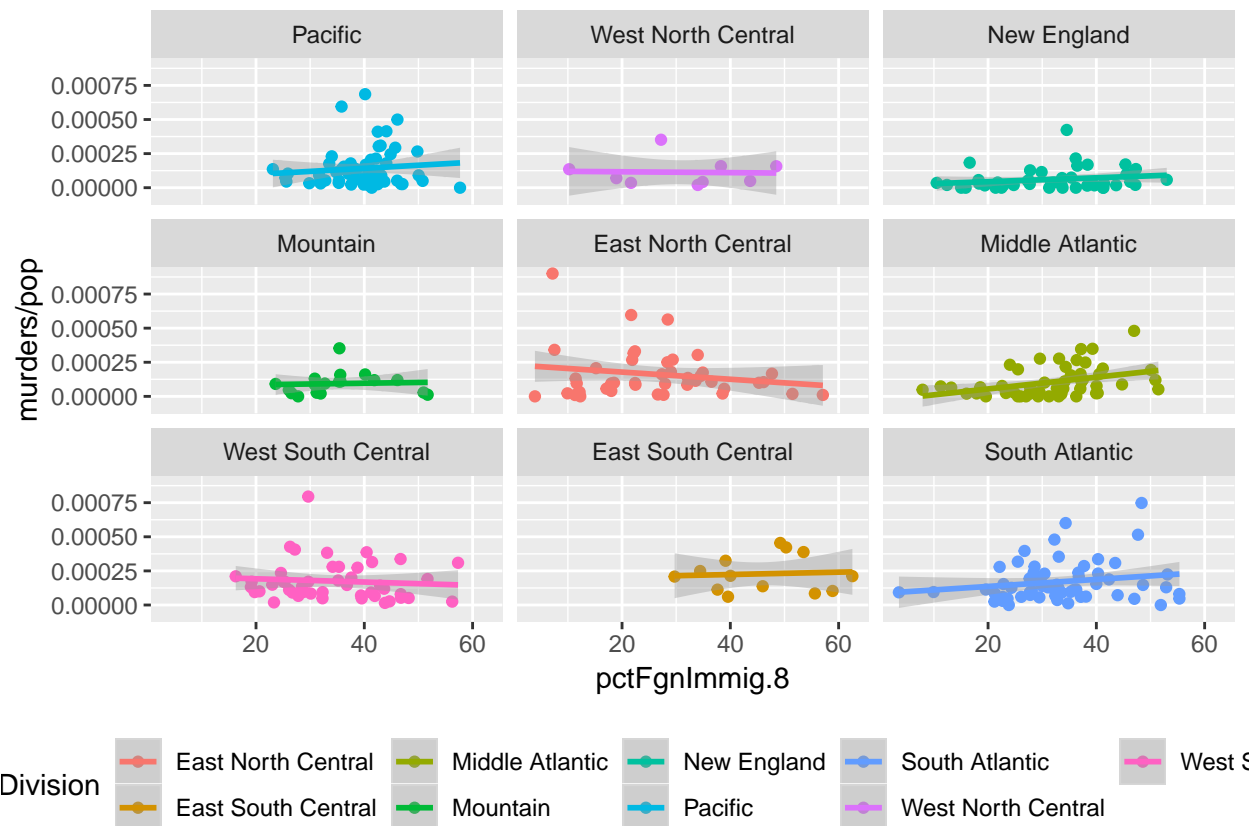
```
# PctImmigRecent: percentage of _immigrants_ who immigrated within last 3 years (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctFgnImmig.3, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctFgnImmig.3, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



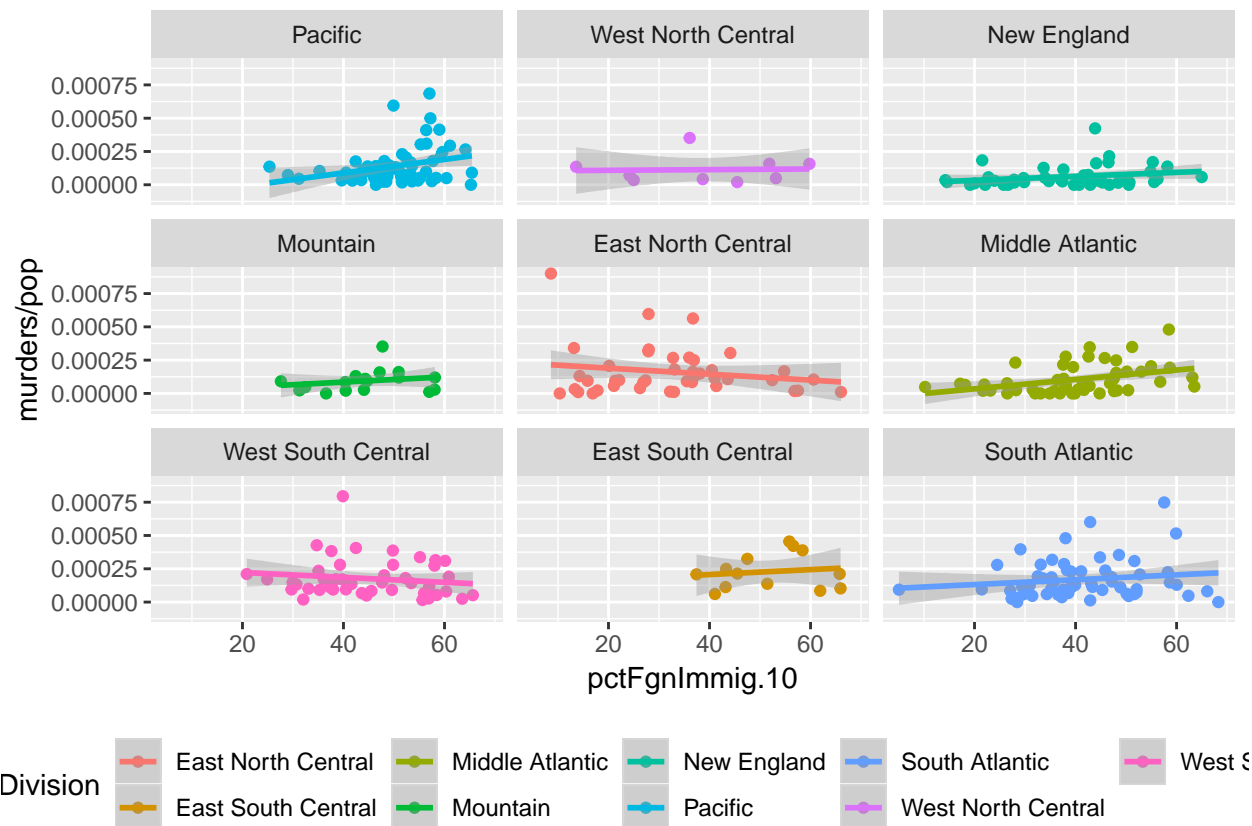
```
# PctImmigRec5: percentage of _immigrants_ who immigrated within last 5 years (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctFgnImmig.5, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctFgnImmig.5, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



```
# PctImmigRec8: percentage of _immigrants_ who immigrated within last 8 years (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctFgnImmig.8, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctFgnImmig.8, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+ xlab("")
  theme(legend.position="bottom")
```



```
# PctImmigRec10: percentage of _immigrants_ who immigrated within last 10 years (numeric - decimal)
ggplot(x) +
  geom_jitter(aes(x=pctFgnImmig.10, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=pctFgnImmig.10, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```



```
# NumKindsDrugsSeiz: number of different kinds of drugs seized (numeric - expected to be integer)
ggplot(x) +
  geom_jitter(aes(x=numDiffDrugsSeiz, y=murders/pop, colour=Division)) +
  geom_smooth(aes(x=numDiffDrugsSeiz, y=murders/pop, colour=Division), method="lm") +
  facet_wrap(~reorder(Division, order)) + #+xlab("")
  theme(legend.position="bottom")
```

