

CDC 500 Cities: Healthcare Access, Behaviors, and Health Outcomes

Stat 198 Final Project

Maya Ghanem and Isabelle Xiong

11/1/2021

Description of Data

(Include description of how you edited the data)

Research Questions

- 1) Do cities with a greater lack of healthcare access have poorer mental health and/or physical health outcomes?
- 2) Does healthcare access, mental health, and/or physical health outcomes vary by state?

Variables of Interest

Explanatory Variables:

- 1) Healthcare Access for Adults (18+): Percent of City Population that Lacks Insurance, Percent of City Population with visits to doctor for routine checkup within the past year, Percent of City Population who have high blood pressure and are taking medicine for high blood pressure control.
- 2) Geographic Distribution by State

Response Variables:

- 1) Behavior for Adults (18+): Percent of city population currently smoking, percent of city population currently reporting binge drinking habits, percent of city population reporting No leisure-time physical activity
- 2) Health Outcomes for Adults (18+): Percent of city population with coronary heart disease, percent of population diagnosed with diabetes, percent of city population with kidney disease

Linear Regressions

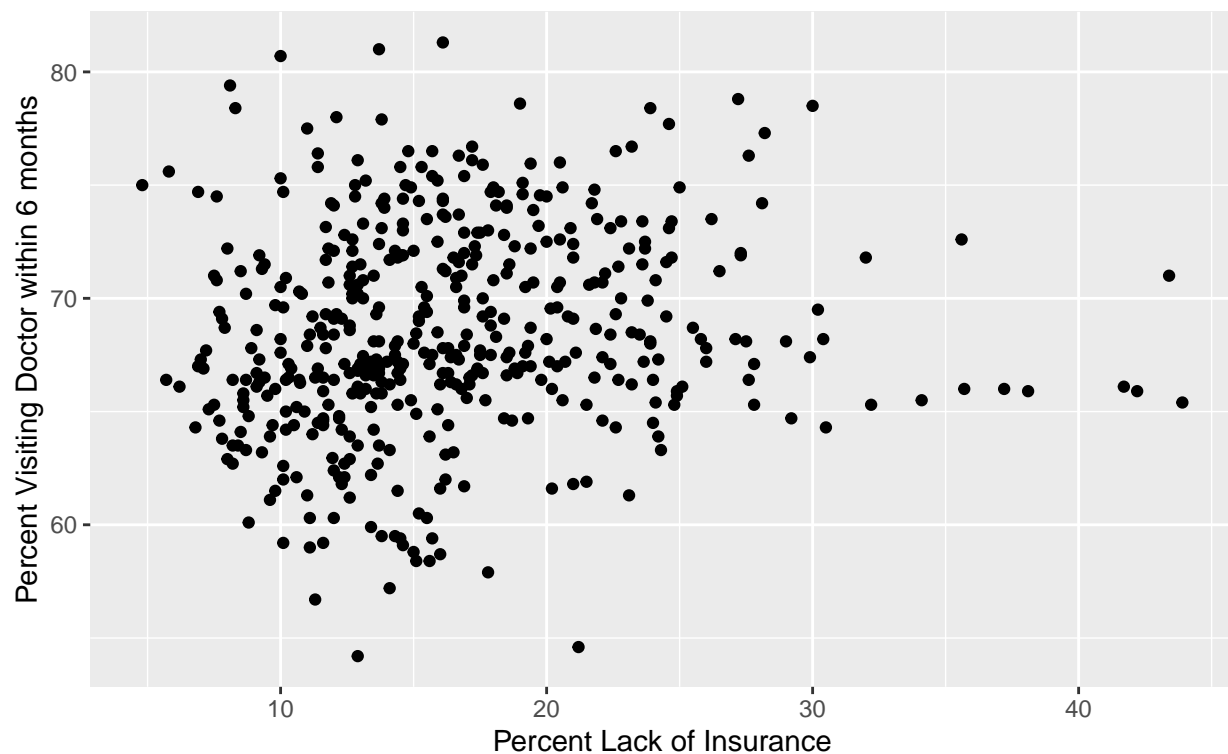
New Method:

- a) Run correlations between the explanatory variables
- b) Run linear regressions and adjusted r squared values
- c) Assess which regression is better
- d) Run the residual plot and the graph

Correlations between Explanatory Variables

```
data_500_cities %>%  
  ggplot(mapping = aes(x = insurance, y = visits_to_doctor)) +  
  geom_point() +  
  labs(  
    title = "Relationship Between Lack of Insurance and Visits to Doctor",  
    subtitle = "Data from CDC 500 Cities",  
    x = "Percent Lack of Insurance",  
    y = "Percent Visiting Doctor within 6 months"  
  )
```

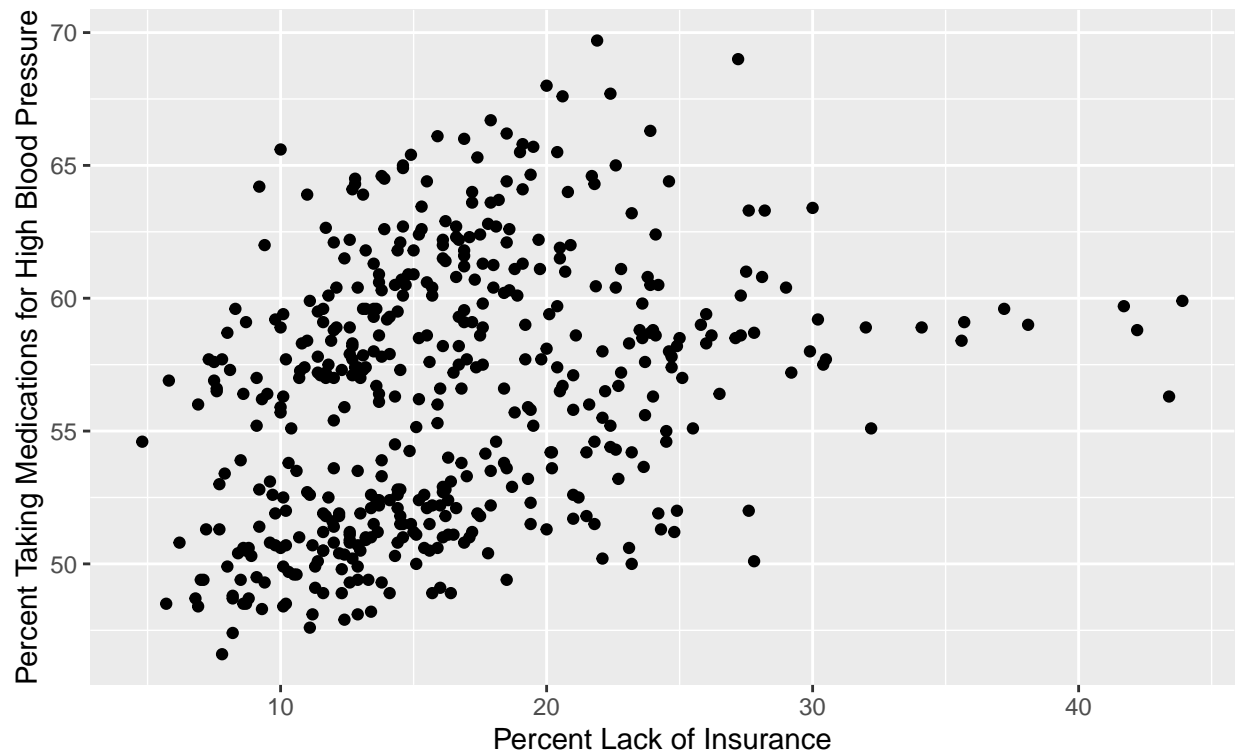
Relationship Between Lack of Insurance and Visits to Doctor
Data from CDC 500 Cities



There does not seem to be any significant correlation.

```
data_500_cities %>%  
  ggplot(mapping = aes(x = insurance, y = medicine_high_bp)) +  
  geom_point() +  
  labs(  
    title = "Relationship Between Lack of Insurance and Percent Pop Taking BP Meds",  
    subtitle = "Data from CDC 500 Cities",  
    x = "Percent Lack of Insurance",  
    y = "Percent Taking Medications for High Blood Pressure"  
  )
```

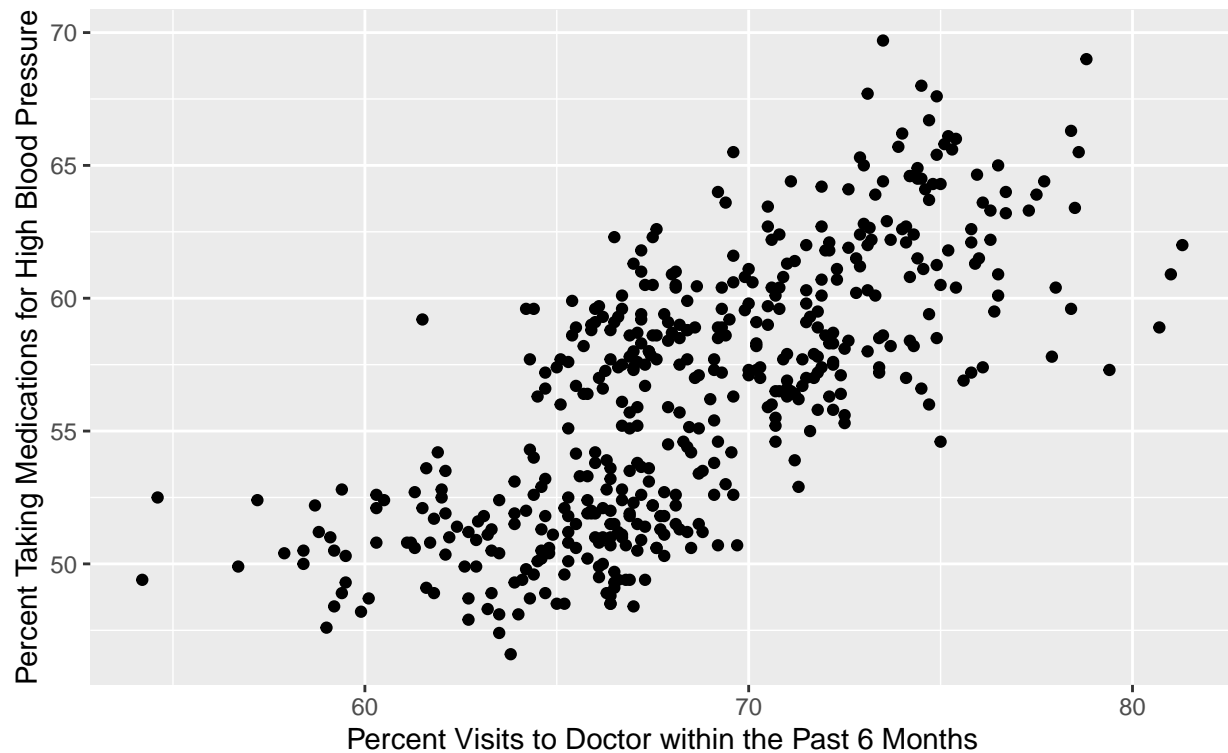
Relationship Between Lack of Insurance and Percent Pop Taking BP Meds
Data from CDC 500 Cities



There does not seem to be any significant correlation.

```
data_500_cities %>%  
  ggplot(mapping = aes(x = visits_to_doctor, y = medicine_high_bp)) +  
  geom_point() +  
  labs(  
    title = "Relationship Between Visits to Doctor and Percent Pop Taking BP Meds",  
    subtitle = "Data from CDC 500 Cities",  
    x = "Percent Visits to Doctor within the Past 6 Months",  
    y = "Percent Taking Medications for High Blood Pressure"  
  )
```

Relationship Between Visits to Doctor and Percent Pop Taking BP Meds
Data from CDC 500 Cities



There seems to be a significant correlation between Visits to Doctor and Taking Medications.

As a result, I will test three models: one with no interaction variables, one with only one interaction variable (Visits_to_Doctor * medicine_high_bp), and one with all three interaction variables.

Access Variables vs. Smoking

Running Linear Regressions

Linear Regression with All Interaction Variables:

```
access_smoking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(smoking ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)
access_smoking_fit_aug <- augment(access_smoking_fit$fit)
tidy(access_smoking_fit) %>%
  print()
```

```
## # A tibble: 4 x 5
##   term          estimate std.error statistic  p.value
##   <chr>          <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)   -15.0      2.08     -7.23 1.99e-12
## 2 insurance      0.0523    0.0237      2.21 2.79e- 2
## 3 visits_to_doctor -0.0966    0.0446     -2.17 3.08e- 2
## 4 medicine_high_bp  0.674     0.0438     15.4 1.59e-43
```

Linear Regression with one interaction variable:

```
one_access_smoking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(smoking ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_high_bp))
one_access_smoking_fit_aug <- augment(one_access_smoking_fit$fit)
tidy(one_access_smoking_fit) %>%
  print()
```

```
## # A tibble: 5 x 5
##   term                                estimate std.error statistic   p.value
##   <chr>                                <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                        84.8       24.7         3.43 0.000657
## 2 insurance                          0.0653    0.0235         2.77 0.00576
## 3 visits_to_doctor                   -1.54     0.360        -4.29 0.0000217
## 4 medicine_high_bp                   -1.12     0.444        -2.52 0.0121
## 5 visits_to_doctor:medicine_high_bp  0.0258    0.00637         4.05 0.0000594
```

Linear Regression with All Interaction Variables

```
int_access_smoking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(smoking ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor) + (insurance * medicine_high_bp) + (visits_to_doctor * medicine_high_bp))
int_access_smoking_fit_aug <- augment(int_access_smoking_fit$fit)
tidy(int_access_smoking_fit) %>%
  print()
```

```
## # A tibble: 7 x 5
##   term                                estimate std.error statistic   p.value
##   <chr>                                <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                        88.9       24.0         3.70 2.41e- 4
## 2 insurance                          0.872     0.417         2.09 3.71e- 2
## 3 visits_to_doctor                   -2.13     0.362        -5.90 6.95e- 9
## 4 medicine_high_bp                   -0.756     0.463        -1.63 1.03e- 1
## 5 insurance:visits_to_doctor          0.0227    0.00634         3.59 3.69e- 4
## 6 insurance:medicine_high_bp         -0.0414    0.00628        -6.58 1.25e-10
## 7 visits_to_doctor:medicine_high_bp  0.0299    0.00667         4.48 9.60e- 6
```

Comparing Adj R-Squared Values

Adj R-Squared Value with No Interactions:

```
glance(access_smoking_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.5150724
```

Adj R-Squared Value with One Interactions:

```
glance(one_access_smoking_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.5305757
```

Adj R-Squared Value with All Interactions:

```
glance(int_access_smoking_fit)$adj.r.squared %>%
  print()
```

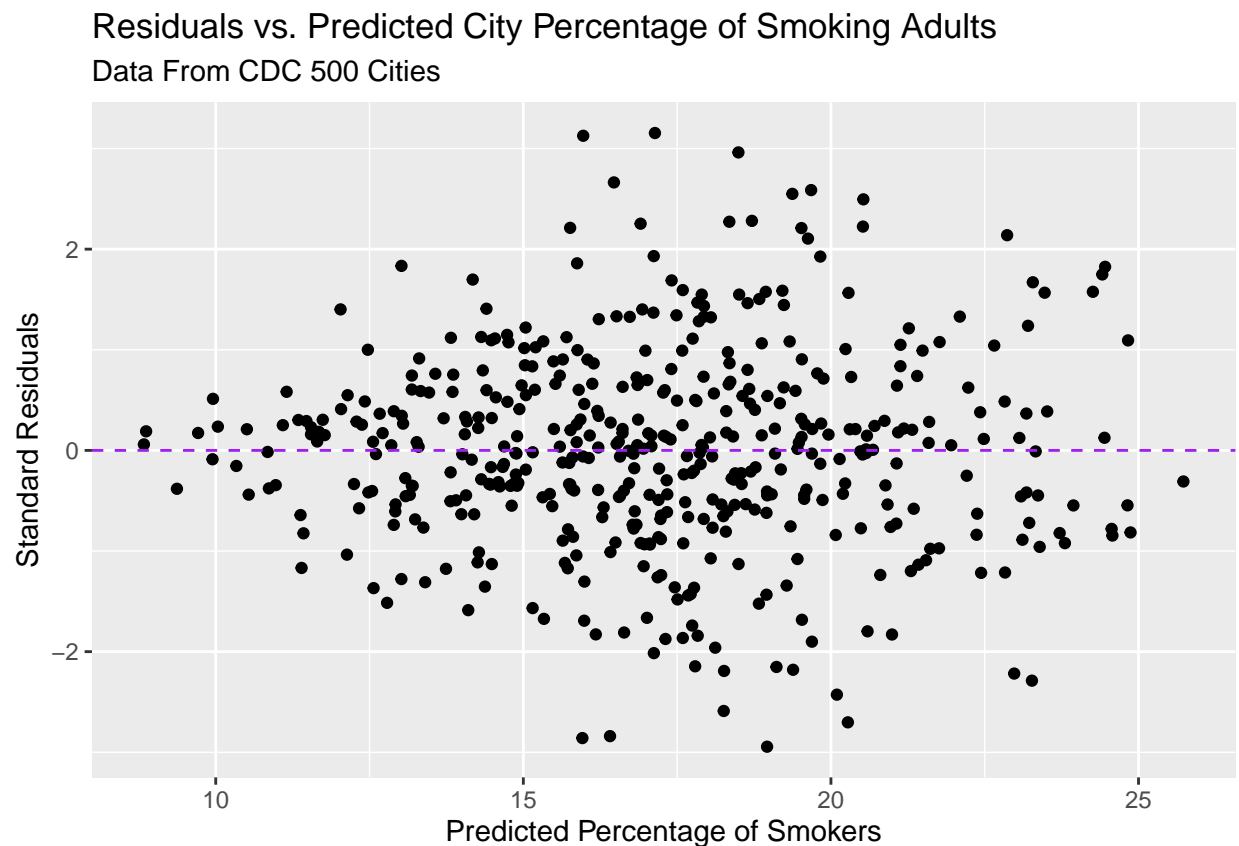
```
## [1] 0.5691301
```

The linear regression with all second order interactions that account for relationships between all explanatory variables is most appropriate because it has the highest adj R-squared value. We will use this regression in displaying our graphs.

Displaying Graphs

Residual Graph

```
int_access_smoking_fit_aug %>%
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +
  labs(
    title = "Residuals vs. Predicted City Percentage of Smoking Adults",
    subtitle = "Data From CDC 500 Cities",
    x = "Predicted Percentage of Smokers",
    y = "Standard Residuals"
  )
)
```



There does not seem to be any patterns in this residual graph, so a linear model would be appropriate.

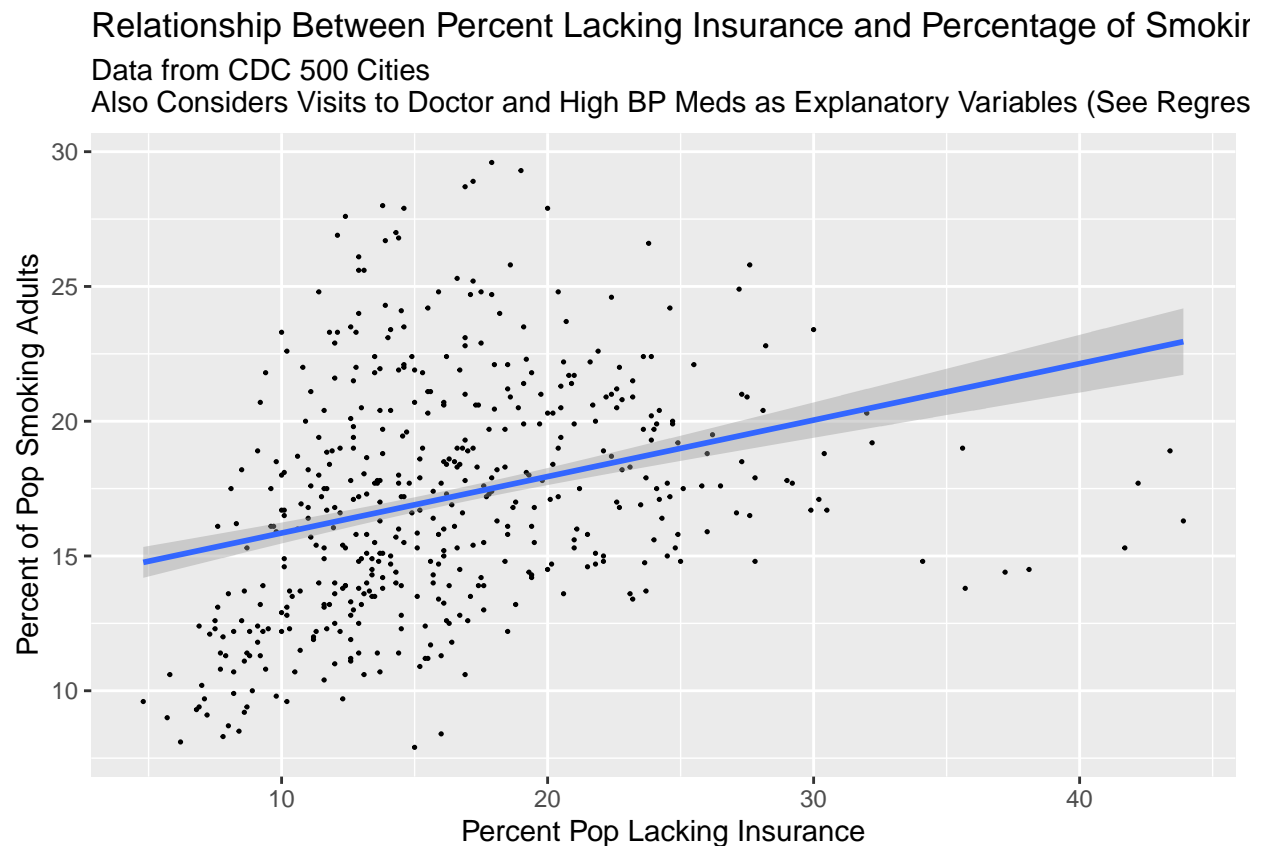
Graph Between Explanatory and Response Variables

```
data_500_cities %>%
  ggplot(mapping = aes(x = insurance, y = smoking)) +
  geom_point(size = 0.25) +
  geom_smooth(method = "lm", data = int_access_smoking_fit_aug, mapping = aes(x = insurance, y = .fitted)) +
  labs(
```

```

title = "Relationship Between Percent Lacking Insurance and Percentage of Smoking Adults",
subtitle = "Data from CDC 500 Cities
Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",
x = "Percent Pop Lacking Insurance",
y = "Percent of Pop Smoking Adults"
)

```



Percent of smoking adults in a city seems to increase with percent of adults in city lacking insurance.

Access Variables vs. Binge Drinking

Running Linear Regressions

Linear Regression for no interactions:

```

access_binge_drinking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(binge_drinking ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)
access_binge_drinking_fit_aug <- augment(access_binge_drinking_fit$fit)
tidy(access_binge_drinking_fit) %>%
  print()

```

```

## # A tibble: 4 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)        24.2      1.58     15.3 2.65e-43
## 2 insurance          -0.162    0.0179    -9.02 4.74e-18
## 3 visits_to_doctor   0.0565   0.0337     1.68 9.45e- 2

```

```
## 4 medicine_high_bp -0.137      0.0331      -4.13 4.39e- 5
```

Linear regression with one interaction:

```
one_access_binge_drinking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(binge_drinking ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_h
one_access_binge_drinking_fit_aug <- augment(one_access_binge_drinking_fit$fit)
tidy(one_access_binge_drinking_fit) %>%
  print()
```

```
## # A tibble: 5 x 5
```

##	term	estimate	std.error	statistic	p.value
##	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	(Intercept)	-133.	17.6	-7.57	1.97e-13
## 2	insurance	-0.183	0.0167	-10.9	8.43e-25
## 3	visits_to_doctor	2.34	0.256	9.13	2.03e-18
## 4	medicine_high_bp	2.69	0.316	8.50	2.50e-16
## 5	visits_to_doctor:medicine_high_bp	-0.0407	0.00453	-8.98	6.76e-18

Linear regression with all interactions:

```
int_access_binge_drinking_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(binge_drinking ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor)
int_access_binge_drinking_fit_aug <- augment(int_access_binge_drinking_fit$fit)
tidy(int_access_binge_drinking_fit) %>%
  print()
```

```
## # A tibble: 7 x 5
```

##	term	estimate	std.error	statistic	p.value
##	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	(Intercept)	-132.	17.8	-7.40	6.26e-13
## 2	insurance	-0.125	0.309	-0.406	6.85e- 1
## 3	visits_to_doctor	2.41	0.268	8.98	6.70e-18
## 4	medicine_high_bp	2.54	0.344	7.38	7.12e-13
## 5	insurance:visits_to_doctor	-0.00655	0.00470	-1.39	1.64e- 1
## 6	insurance:medicine_high_bp	0.00686	0.00466	1.47	1.42e- 1
## 7	visits_to_doctor:medicine_high_bp	-0.0401	0.00495	-8.10	4.93e-15

Comparing Adj R-Squared Values

Adj R-squared value for regression with no interactions:

```
glance(access_binge_drinking_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.2367489
```

Adj R-squared value for regression with one interaction:

```
glance(one_access_binge_drinking_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.347712
```

Adj R-squared value for regression with all interactions:

```
glance(int_access_binge_drinking_fit)$adj.r.squared %>%
  print()
```



```
## [1] 0.3488416
```

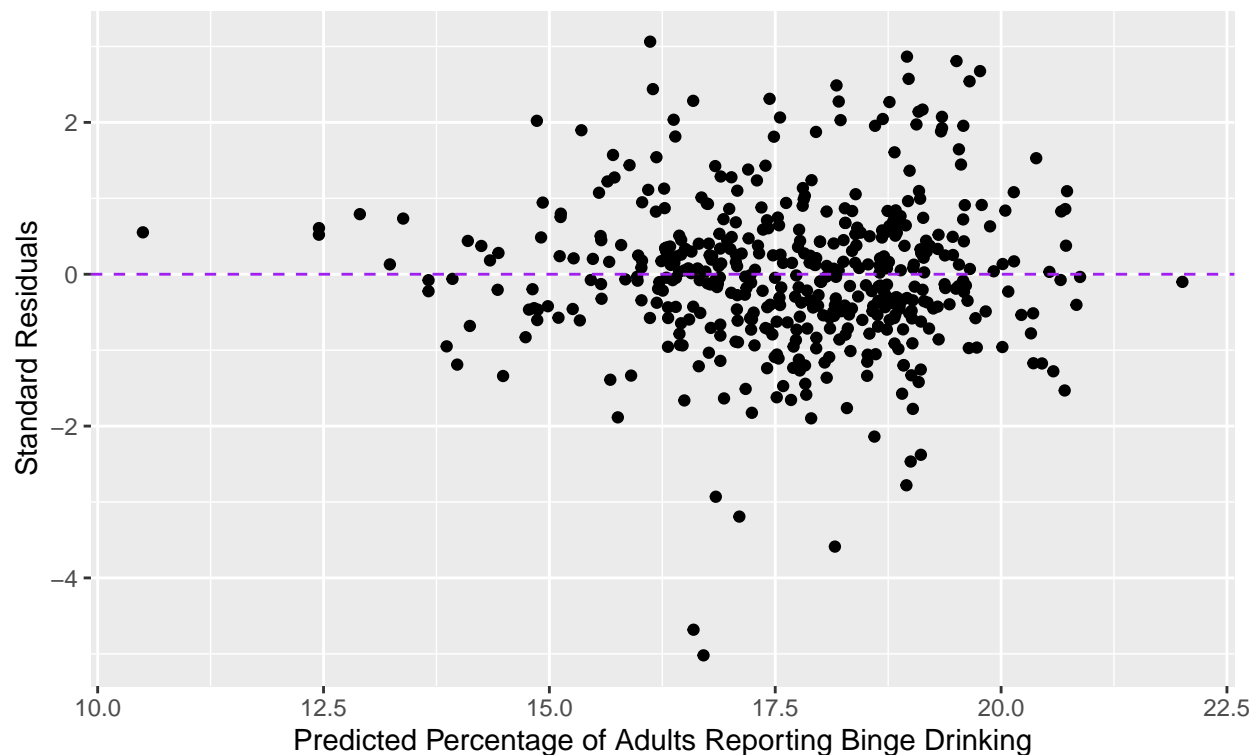
The linear regression with all second order interactions that account for relationships between explanatory variables is most appropriate because it has the highest adj R-squared value. We will use this regression in displaying our graphs.

Displaying Graphs

Residual Graph

```
int_access_binge_drinking_fit_aug %>%  
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +  
  geom_point() +  
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +  
  labs(  
    title = "Residuals vs. Predicted Percentage of City Reporting Binge Drinking",  
    subtitle = "Data From CDC 500 Cities",  
    x = "Predicted Percentage of Adults Reporting Binge Drinking",  
    y = "Standard Residuals"  
  )
```

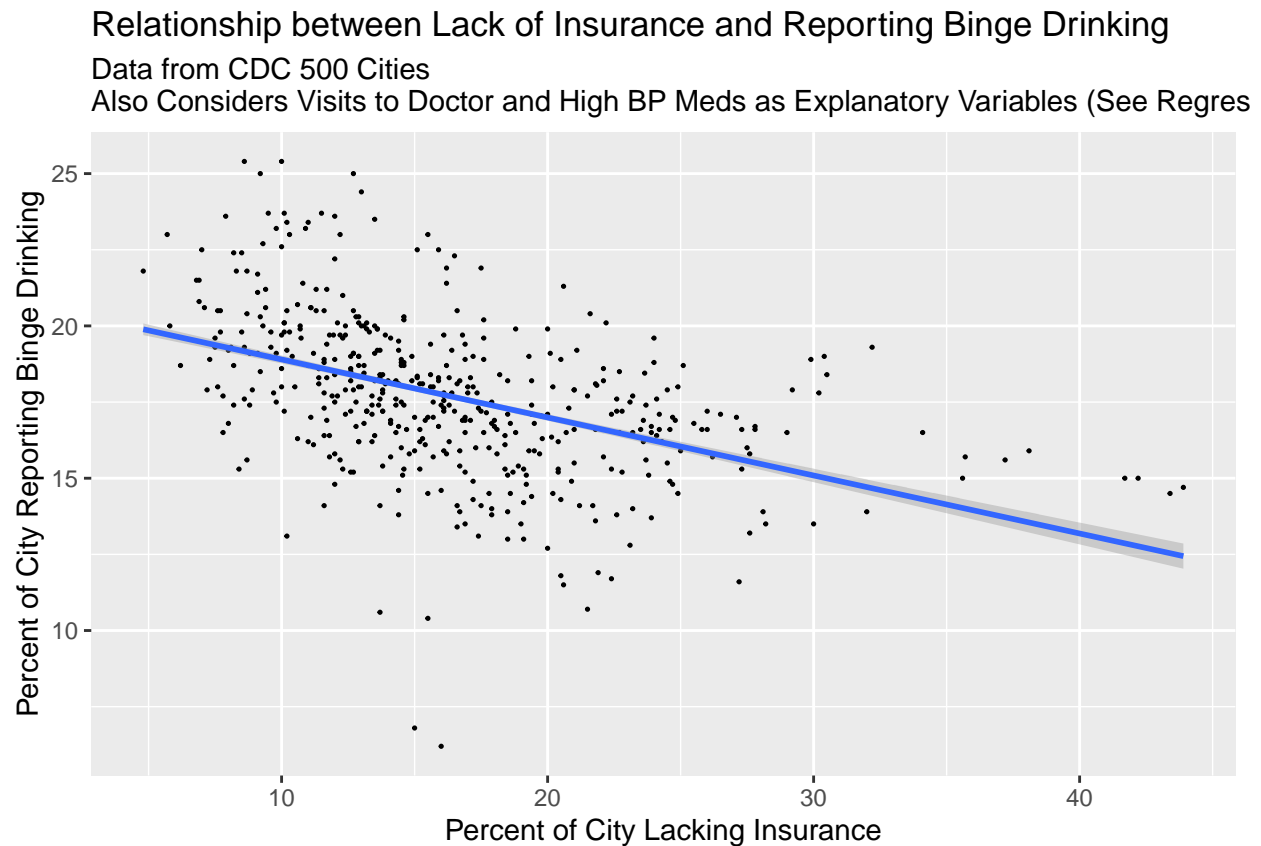
Residuals vs. Predicted Percentage of City Reporting Binge Drinking
Data From CDC 500 Cities



There doesn't seem to be any major patterns in this residual graph, except for some clumping around the mean residual. A linear regression still seems appropriate.

Graph Comparing Explanatory and Response Variables

```
data_500_cities %>%
  ggplot(mapping = aes(x = insurance, y = binge_drinking)) +
  geom_point(size = 0.25) +
  geom_smooth(method = "lm", data = int_access_binge_drinking_fit_aug, mapping = aes(x = insurance, y = .))
  labs(
    title = "Relationship between Lack of Insurance and Reporting Binge Drinking",
    subtitle = "Data from CDC 500 Cities
    Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",
    x = "Percent of City Lacking Insurance",
    y = "Percent of City Reporting Binge Drinking"
  )
```



As the percentage of city population lacking health insurance increases, the percentage of city reporting binge drinking decreases.

Access Variables vs. Physical Activity

Running Linear Regressions

Linear regression with no interactions

```
access_physical_activity_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(physical_activity ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)
access_physical_activity_fit_aug <- augment(access_physical_activity_fit$fit)
tidy(access_physical_activity_fit) %>%
  print()
```

```
## # A tibble: 4 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)      -28.1      1.77     -15.9  5.76e-46
## 2 insurance          0.533     0.0201     26.5  3.31e-95
## 3 visits_to_doctor  0.0625    0.0378      1.65  9.95e- 2
## 4 medicine_high_bp  0.738     0.0371     19.9  3.54e-64
```

Linear regression with one interaction

```
one_access_physical_activity_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(physical_activity ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_high_bp))
one_access_physical_activity_fit_aug <- augment(one_access_physical_activity_fit$fit)
tidy(one_access_physical_activity_fit) %>%
  print()
```

```
## # A tibble: 5 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)        43.5      21.1       2.06  3.98e- 2
## 2 insurance           0.543     0.0201     27.0  1.71e-97
## 3 visits_to_doctor   -0.976     0.307     -3.18  1.57e- 3
## 4 medicine_high_bp   -0.548     0.379     -1.44  1.49e- 1
## 5 visits_to_doctor:medicine_high_bp  0.0185    0.00543     3.41  7.11e- 4
```

Linear regression with all interactions

```
int_access_physical_activity_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(physical_activity ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor + insurance * medicine_high_bp + visits_to_doctor * medicine_high_bp))
int_access_physical_activity_fit_aug <- augment(int_access_physical_activity_fit$fit)
tidy(int_access_physical_activity_fit) %>%
  print()
```

```
## # A tibble: 7 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)        55.1      20.8       2.64  0.00845
## 2 insurance           1.96      0.361       5.42  0.0000000972
## 3 visits_to_doctor   -1.47      0.313      -4.69  0.00000361
## 4 medicine_high_bp   -0.744     0.402     -1.85  0.0646
## 5 insurance:visits_to_doctor  0.000790  0.00549     0.144  0.886
## 6 insurance:medicine_high_bp -0.0257    0.00545     -4.72  0.00000317
## 7 visits_to_doctor:medicine_high_bp  0.0271    0.00578     4.68  0.00000373
```

Comparing Adj R-Squared Values

Adj R-squared value for regression with no interactions

```
glance(access_physical_activity_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.8369087
```

Adj R-squared value for regression with one interaction

```
glance(one_access_physical_activity_fit)$adj.r.squared %>%  
  print()
```

```
## [1] 0.8405259
```

Adj R-squared value for regression with all interactions

```
glance(int_access_physical_activity_fit)$adj.r.squared %>%  
  print()
```

```
## [1] 0.8488063
```

The linear regression that includes all possible second order interactions for the three explanatory variables is most appropriate because it has the highest adjusted R-squared value. It will therefore be visualized in the residual plot and displayed in a graph.

Displaying Graphs

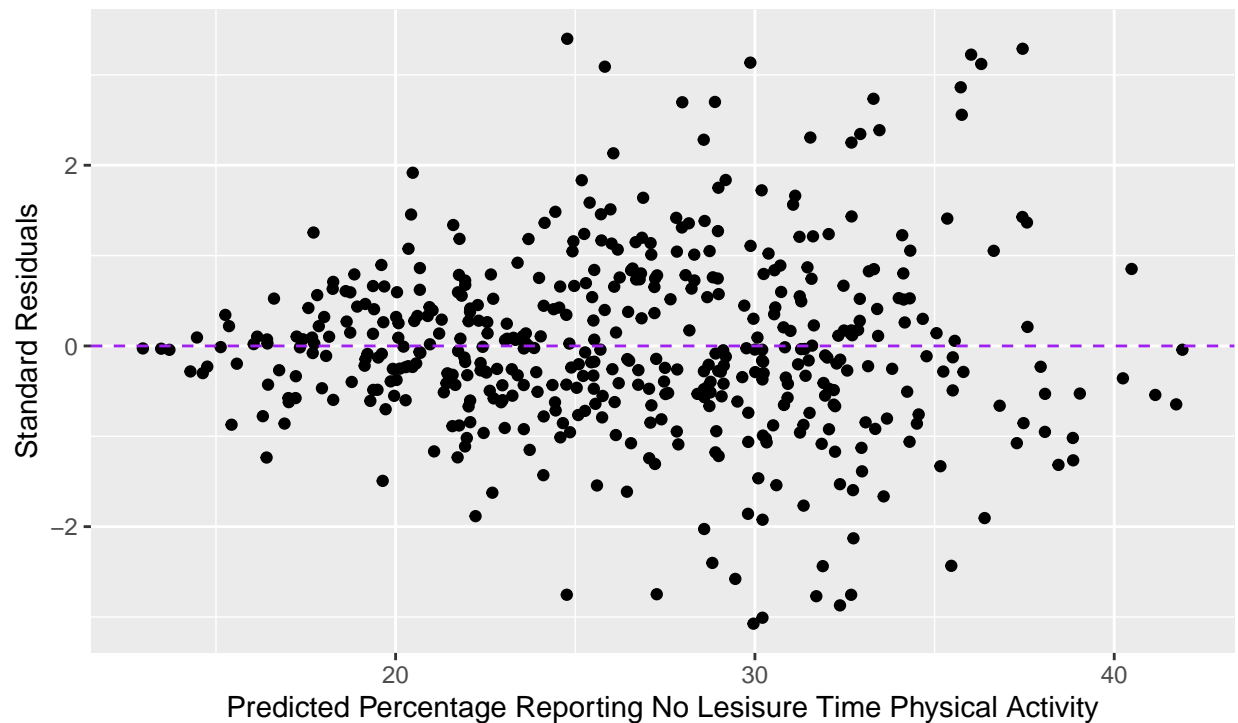
Residual Graph

```
int_access_physical_activity_fit_aug %>%  
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +  
  geom_point() +  
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +  
  labs(  
    title = "Residuals vs. Predicted Percentage of City Reporting No Physical Activity",  
    subtitle = "Data from CDC 500 Cities  
Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",  
    x = "Predicted Percentage Reporting No Lesisure Time Physical Activity",  
    y = "Standard Residuals"  
  )
```

Residuals vs. Predicted Percentage of City Reporting No Physical Activity

Data from CDC 500 Cities

Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regres



Because there does not seem to be any patterns in the residual plot, a linear model is likely appropriate.

Graph Comparing Explanatory and Response Variables

```
data_500_cities %>%
  ggplot(mapping = aes(x = insurance, y = physical_activity)) +
  geom_point(size = 0.25) +
  geom_smooth(method = "lm", data = int_access_physical_activity_fit_aug, mapping = aes(x = insurance, y = physical_activity)) +
  labs(
    title = "Relationship Between Lacking Insurance and No Physical Activity",
    subtitle = "Data from CDC 500 Cities
    Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",
    x = "Percent of City Lacking Insurance",
    y = "Percent of City Reporting No Physical Activity"
  )
```

Relationship Between Lacking Insurance and No Physical Activity

Data from CDC 500 Cities

Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regres



There seems to be a very strong positive correlation between percent of city lacking health insurance and percent of city reporting no physical activity.

Access Variables vs. Coronary Heart Disease

Running Linear Regressions

Linear regression with no interactions:

```
access_heart_disease_fit <- linear_reg() %>%  
  set_engine("lm") %>%  
  fit(heart_disease ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)  
access_heart_disease_fit_aug <- augment(access_heart_disease_fit$fit)  
tidy(access_heart_disease_fit) %>%  
  print()
```

```
## # A tibble: 4 x 5  
##   term                estimate std.error statistic  p.value  
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>  
## 1 (Intercept)      -1.54      0.427     -3.60 3.56e- 4  
## 2 insurance         0.0669    0.00487    13.7 2.32e-36  
## 3 visits_to_doctor -0.0113    0.00916    -1.23 2.20e- 1  
## 4 medicine_high_bp  0.122     0.00898    13.6 1.16e-35
```

Linear regression with one interaction

```
one_access_heart_disease_fit <- linear_reg() %>%  
  set_engine("lm") %>%
```

```
fit(heart_disease ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_high_bp))
access_heart_disease_fit_aug <- augment(access_heart_disease_fit$fit)
tidy(access_heart_disease_fit) %>%
  print()
```

```
## # A tibble: 4 x 5
##   term                estimate std.error statistic  p.value
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)       -1.54      0.427     -3.60 3.56e- 4
## 2 insurance          0.0669    0.00487    13.7 2.32e-36
## 3 visits_to_doctor  -0.0113    0.00916    -1.23 2.20e- 1
## 4 medicine_high_bp   0.122     0.00898    13.6 1.16e-35
```

Linear regression with all interactions

```
int_access_heart_disease_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(heart_disease ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor) + (insurance * medicine_high_bp) + (visits_to_doctor * medicine_high_bp))
int_access_heart_disease_fit_aug <- augment(int_access_heart_disease_fit$fit)
tidy(int_access_heart_disease_fit) %>%
  print()
```

```
## # A tibble: 7 x 5
##   term                                estimate std.error statistic  p.value
##   <chr>                                <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)                        23.9      4.94      4.84 1.74e- 6
## 2 insurance                          0.352     0.0857     4.10 4.79e- 5
## 3 visits_to_doctor                   -0.480     0.0743    -6.46 2.70e-10
## 4 medicine_high_bp                   -0.289     0.0952    -3.04 2.52e- 3
## 5 insurance:visits_to_doctor          0.00239    0.00130     1.84 6.67e- 2
## 6 insurance:medicine_high_bp         -0.00780    0.00129    -6.04 3.19e- 9
## 7 visits_to_doctor:medicine_high_bp  0.00767    0.00137     5.59 3.80e- 8
```

Comparing Adj R Squared Values

Adj R-squared values for regression with no interactions

```
glance(access_heart_disease_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.6254959
```

Adj R-squared values for regression with one interaction

```
glance(one_access_heart_disease_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.6413167
```

Adj R-squared values for regression with all interactions

```
glance(int_access_heart_disease_fit)$adj.r.squared %>%
  print()
```

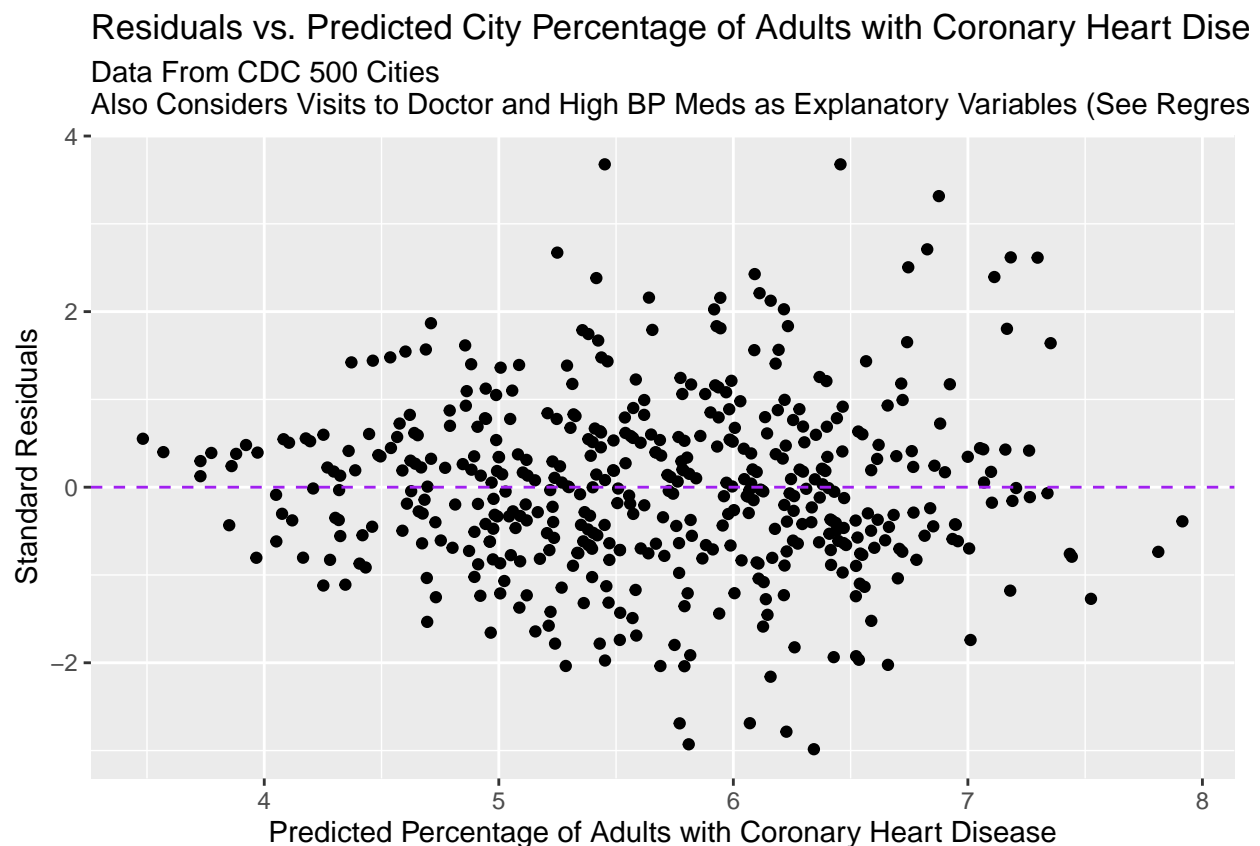
```
## [1] 0.6667498
```

The linear regression that includes all possible interactions between the three explanatory variables is most appropriate because it has the greatest adj R-squared value. This will then be used when displaying graphs.

Displaying Graphs

Residual Graphs

```
int_access_heart_disease_fit_aug %>%
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +
  labs(
    title = "Residuals vs. Predicted City Percentage of Adults with Coronary Heart Disease",
    subtitle = "Data From CDC 500 Cities
Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",
    x = "Predicted Percentage of Adults with Coronary Heart Disease",
    y = "Standard Residuals"
  )
```



There does seem to be a significant pattern in the residual model, so a linear model does not seem appropriate. Try a logistical model here?

Access Variables vs. Diabetes

Running linear regressions

Linear regression with one interaction

```
access_diabetes_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(diabetes ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)
access_diabetes_fit_aug <- augment(access_diabetes_fit$fit)
```



```
tidy(access_diabetes_fit) %>%
  print()
```

```
## # A tibble: 4 x 5
##   term                estimate std.error statistic  p.value
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)        -7.57      0.982     -7.71 7.45e-14
## 2 insurance           0.239     0.0112     21.4 2.12e-71
## 3 visits_to_doctor    0.0650    0.0210      3.09 2.13e- 3
## 4 medicine_high_bp    0.171     0.0206      8.29 1.18e-15
```

Linear regression with one interaction

```
one_access_diabetes_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(diabetes ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_high_bp))
access_diabetes_fit_aug <- augment(one_access_diabetes_fit$fit)
tidy(one_access_diabetes_fit) %>%
  print()
```

```
## # A tibble: 4 x 5
##   term                estimate std.error statistic  p.value
##   <chr>                <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)        -7.57      0.982     -7.71 7.45e-14
## 2 insurance           0.239     0.0112     21.4 2.12e-71
## 3 visits_to_doctor    0.0650    0.0210      3.09 2.13e- 3
## 4 medicine_high_bp    0.171     0.0206      8.29 1.18e-15
```

Linear regression with all interactions

```
int_access_diabetes_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(diabetes ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor) + (insurance * medicine_high_bp) + (visits_to_doctor * medicine_high_bp))
int_access_diabetes_fit_aug <- augment(int_access_diabetes_fit$fit)
tidy(int_access_diabetes_fit) %>%
  print()
```

```
## # A tibble: 7 x 5
##   term                                estimate std.error statistic  p.value
##   <chr>                                <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)                        69.9      11.4      6.12 1.97e- 9
## 2 insurance                          0.975     0.198      4.92 1.22e- 6
## 3 visits_to_doctor                   -1.07     0.172     -6.25 9.40e-10
## 4 medicine_high_bp                   -1.40     0.220     -6.36 4.72e-10
## 5 insurance:visits_to_doctor          -0.00935  0.00301     -3.10 2.03e- 3
## 6 insurance:medicine_high_bp          -0.00147  0.00299     -0.493 6.22e- 1
## 7 visits_to_doctor:medicine_high_bp  0.0230    0.00317      7.24 1.87e-12
```

Comparing Adj R-Squared Values

Adj R-squared value for regression with no interactions

```
glance(access_diabetes_fit)$adj.r.squared %>%
  print()
```

```
## [1] 0.6797326
```

Adj R-squared value for regression with one interaction

```
glance(one_access_diabetes_fit)$adj.r.squared %>%  
  print()
```

```
## [1] 0.703361
```

Adj R-squared value for regression with all interactions

```
glance(int_access_diabetes_fit)$adj.r.squared %>%  
  print()
```

```
## [1] 0.7110294
```

The linear regression including all possible second order interactions between the explanatory variables is most appropriate because it has the highest adj R-squared value. Graphs displayed will therefore use this model.

Displaying Graphs

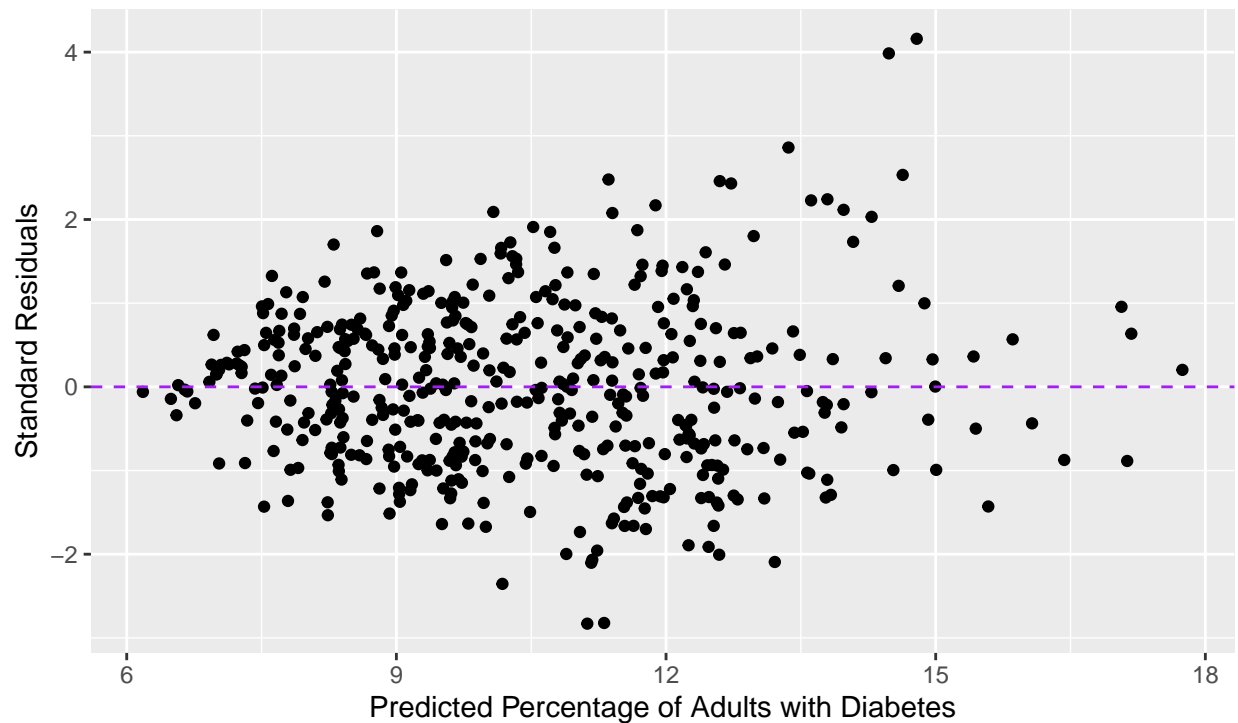
Residual Graph (Note any patterns)

```
int_access_diabetes_fit_aug %>%  
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +  
  geom_point() +  
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +  
  labs(  
    title = "Residuals vs. Predicted City Percentage of Adults with Diabetes",  
    subtitle = "Data From CDC 500 Cities  
Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",  
    x = "Predicted Percentage of Adults with Diabetes",  
    y = "Standard Residuals"  
  )
```

Residuals vs. Predicted City Percentage of Adults with Diabetes

Data From CDC 500 Cities

Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regres



There does not seem to be a significant pattern in the residual plot. Therefore, a linear model is appropriate.

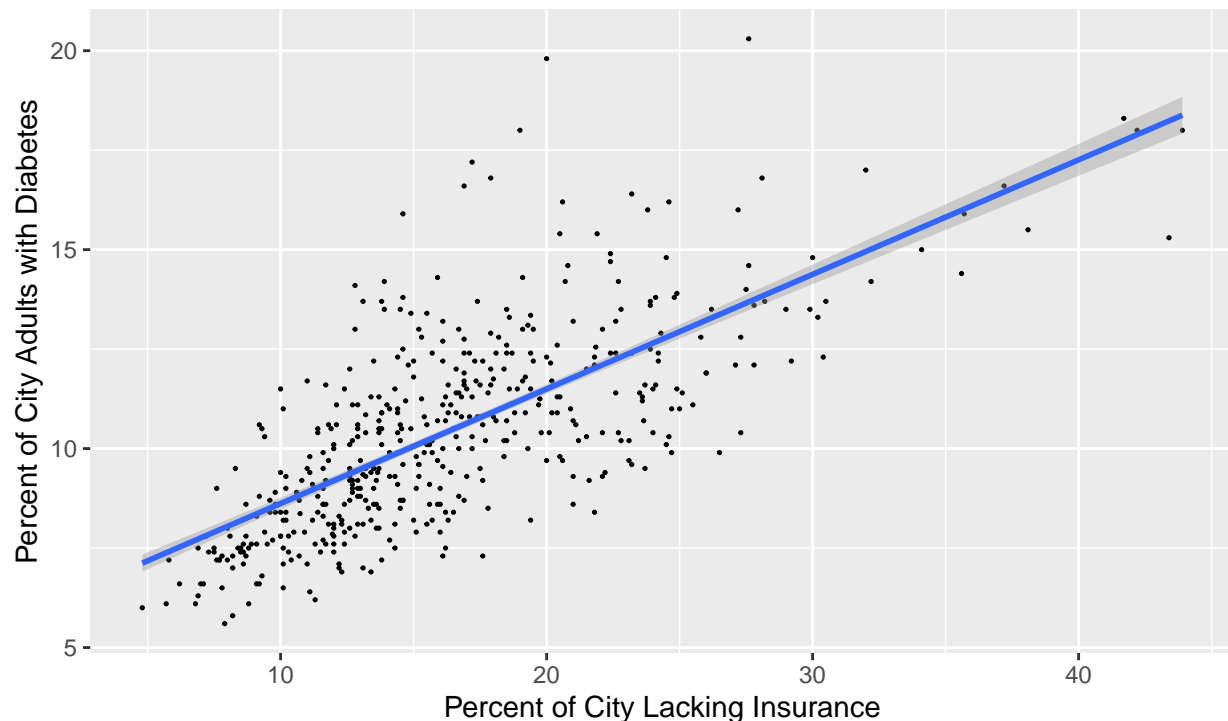
Graph comparing explanatory and response variables

```
data_500_cities %>%  
  ggplot( mapping = aes(x = insurance, y = diabetes)) +  
  geom_point(size = 0.25) +  
  geom_smooth(method = "lm", data = int_access_diabetes_fit_aug, mapping = aes(x = insurance, y = .fitted),  
    labs(  
      title = "Relationship Between Lacking Insurance and Adults with Diabetes",  
      subtitle = "Data from CDC 500 Cities  
Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",  
      x = "Percent of City Lacking Insurance",  
      y = "Percent of City Adults with Diabetes"  
    )  
  )
```

Relationship Between Lacking Insurance and Adults with Diabetes

Data from CDC 500 Cities

Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regres



There seems to be a strong positive correlation between percent of city lacking health insurance and percent of city adults diagnosed with diabetes.

Access Variables vs. Kidney Disease

Running Linear Regression Models

Linear Regression Model with no interactions

```
access_kidney_disease_fit <- linear_reg() %>%  
  set_engine("lm") %>%  
  fit(kidney_disease ~ insurance + visits_to_doctor + medicine_high_bp, data = data_500_cities)  
access_kidney_disease_fit_aug <- augment(access_kidney_disease_fit$fit)  
tidy(access_kidney_disease_fit) %>%  
  print()
```

```
## # A tibble: 4 x 5  
##   term                estimate std.error statistic  p.value  
##   <chr>              <dbl>    <dbl>    <dbl>    <dbl>  
## 1 (Intercept)        0.290      0.225      1.29 1.97e- 1  
## 2 insurance           0.0424     0.00256    16.6 7.48e-49  
## 3 visits_to_doctor    0.00522     0.00482     1.08 2.79e- 1  
## 4 medicine_high_bp    0.0305     0.00472     6.47 2.54e-10
```

Linear regression model with one interaction

```
one_access_kidney_disease_fit <- linear_reg() %>%  
  set_engine("lm") %>%
```

```

fit(kidney_disease ~ insurance + visits_to_doctor + medicine_high_bp + (visits_to_doctor * medicine_h
one_access_kidney_disease_fit_aug <- augment(one_access_kidney_disease_fit$fit)
tidy(one_access_kidney_disease_fit) %>%
print()

```

```

## # A tibble: 5 x 5
##   term                                estimate std.error statistic  p.value
##   <chr>                                <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                        21.7       2.53       8.59 1.34e-16
## 2 insurance                          0.0452    0.00241    18.8 4.81e-59
## 3 visits_to_doctor                   -0.305     0.0368    -8.30 1.16e-15
## 4 medicine_high_bp                   -0.354     0.0454    -7.79 4.40e-14
## 5 visits_to_doctor:medicine_high_bp  0.00554    0.000651     8.50 2.54e-16

```

Linear regression model with all interactions

```

int_access_kidney_disease_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(kidney_disease ~ insurance + visits_to_doctor + medicine_high_bp + (insurance * visits_to_doctor)
int_access_kidney_disease_fit_aug <- augment(int_access_kidney_disease_fit$fit)
tidy(int_access_kidney_disease_fit) %>%
print()

```

```

## # A tibble: 7 x 5
##   term                                estimate std.error statistic  p.value
##   <chr>                                <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)                        22.9       2.50       9.16 1.63e-18
## 2 insurance                          0.198     0.0435     4.56 6.44e- 6
## 3 visits_to_doctor                   -0.361     0.0377    -9.57 6.10e-20
## 4 medicine_high_bp                   -0.372     0.0483    -7.70 8.53e-14
## 5 insurance:visits_to_doctor          0.000243  0.000661     0.368 7.13e- 1
## 6 insurance:medicine_high_bp         -0.00297  0.000655    -4.53 7.40e- 6
## 7 visits_to_doctor:medicine_high_bp  0.00646   0.000696     9.28 6.23e-19

```

Comparing Adj R-Squared Values

Adj R-squared value for regression with no interactions

```

glance(access_kidney_disease_fit)$adj.r.squared %>%
print()

```

```
## [1] 0.5403031
```

Adj R-squared value for regression with one interaction

```

glance(one_access_kidney_disease_fit)$adj.r.squared %>%
print()

```

```
## [1] 0.6010605
```

Adj R-squared value for regression with all interactions

```

glance(int_access_kidney_disease_fit)$adj.r.squared %>%
print()

```

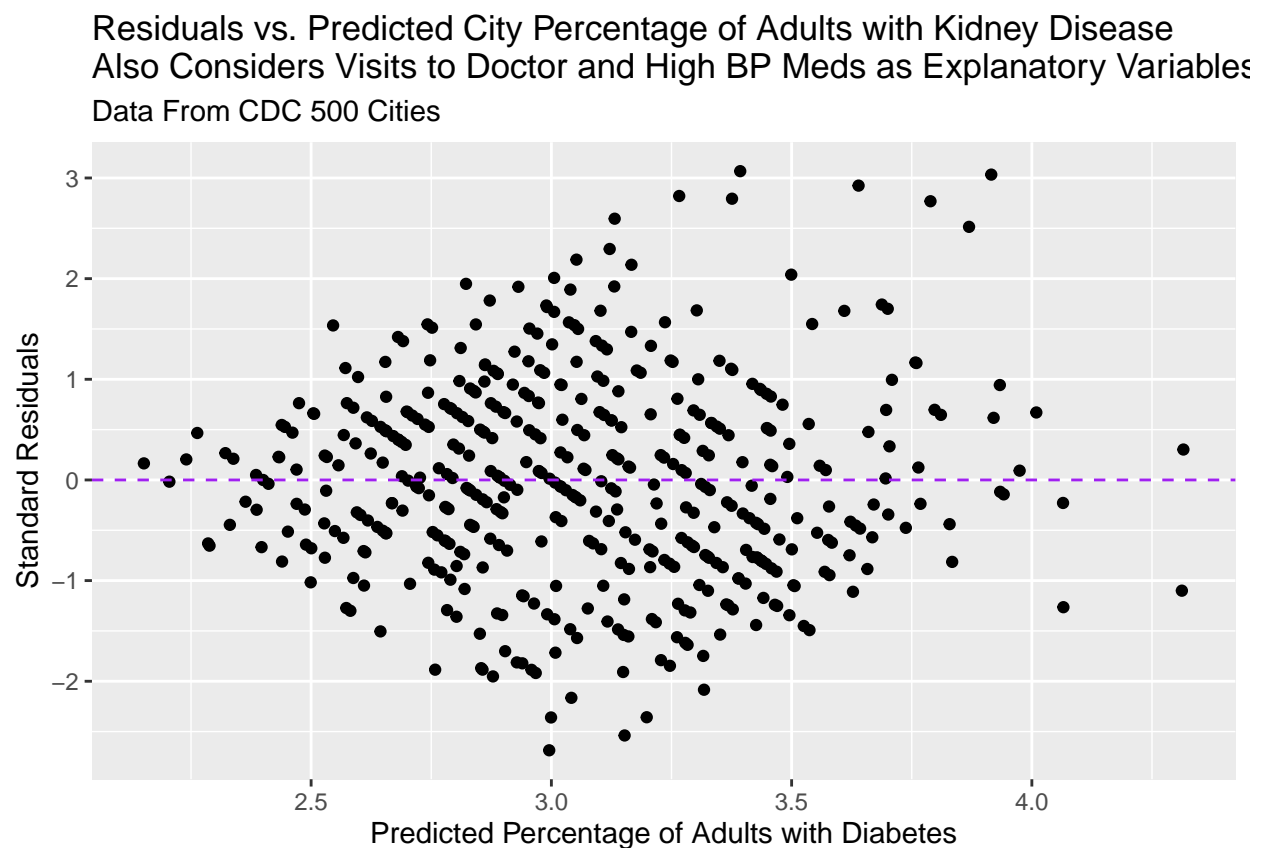
```
## [1] 0.6193093
```

The linear model with all possible second order interactions between the three explanatory variables is most appropriate because it has the highest R-squared value.

Displaying Graphs:

Residual Graph

```
int_access_kidney_disease_fit_aug %>%
  ggplot(mapping = aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "purple", lty = "dashed") +
  labs(
    title = "Residuals vs. Predicted City Percentage of Adults with Kidney Disease",
    subtitle = "Also Considers Visits to Doctor and High BP Meds as Explanatory Variables (See Regression)",
    x = "Predicted Percentage of Adults with Diabetes",
    y = "Standard Residuals"
  )
```



There seems to be a significant pattern in the residual plot, so a linear model would not be appropriate. Try a logistic model?

ANOVA Testing

Map Visualization

```
theme_set(theme_bw())
world <- ne_countries(scale = "medium", returnclass = "sf")
names(world)
```

```
## [1] "scalerank" "featurecla" "labelrank" "sovereight" "sov_a3"
```

```
## [6] "adm0_dif" "level" "type" "admin" "adm0_a3"
## [11] "geou_dif" "geounit" "gu_a3" "su_dif" "subunit"
## [16] "su_a3" "brk_diff" "name" "name_long" "brk_a3"
## [21] "brk_name" "brk_group" "abbrev" "postal" "formal_en"
## [26] "formal_fr" "note_adm0" "note_brk" "name_sort" "name_alt"
## [31] "mapcolor7" "mapcolor8" "mapcolor9" "mapcolor13" "pop_est"
## [36] "gdp_md_est" "pop_year" "lastcensus" "gdp_year" "economy"
## [41] "income_grp" "wikipedia" "fips_10" "iso_a2" "iso_a3"
## [46] "iso_n3" "un_a3" "wb_a2" "wb_a3" "woe_id"
## [51] "adm0_a3_is" "adm0_a3_us" "adm0_a3_un" "adm0_a3_wb" "continent"
## [56] "region_un" "subregion" "region_wb" "name_len" "long_len"
## [61] "abbrev_len" "tiny" "homepart" "geometry"
```

```
state.name
```

```
## [1] "Alabama" "Alaska" "Arizona" "Arkansas"
## [5] "California" "Colorado" "Connecticut" "Delaware"
## [9] "Florida" "Georgia" "Hawaii" "Idaho"
## [13] "Illinois" "Indiana" "Iowa" "Kansas"
## [17] "Kentucky" "Louisiana" "Maine" "Maryland"
## [21] "Massachusetts" "Michigan" "Minnesota" "Mississippi"
## [25] "Missouri" "Montana" "Nebraska" "Nevada"
## [29] "New Hampshire" "New Jersey" "New Mexico" "New York"
## [33] "North Carolina" "North Dakota" "Ohio" "Oklahoma"
## [37] "Oregon" "Pennsylvania" "Rhode Island" "South Carolina"
## [41] "South Dakota" "Tennessee" "Texas" "Utah"
## [45] "Vermont" "Virginia" "Washington" "West Virginia"
## [49] "Wisconsin" "Wyoming"
```

```
head(world)
```

```
## Simple feature collection with 6 features and 63 fields
## Geometry type: MULTIPOLYGON
## Dimension: XY
## Bounding box: xmin: -70.06611 ymin: -18.01973 xmax: 74.89131 ymax: 60.40581
## CRS: +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0
## scalerank featurecla labelrank sovereignt sov_a3 adm0_dif level
## 0 3 Admin-0 country 5 Netherlands NL1 1 2
## 1 1 Admin-0 country 3 Afghanistan AFG 0 2
## 2 1 Admin-0 country 3 Angola AGO 0 2
## 3 1 Admin-0 country 6 United Kingdom GB1 1 2
## 4 1 Admin-0 country 6 Albania ALB 0 2
## 5 3 Admin-0 country 6 Finland FI1 1 2
## type admin adm0_a3 geou_dif geounit gu_a3 su_dif
## 0 Country Aruba ABW 0 Aruba ABW 0
## 1 Sovereign country Afghanistan AFG 0 Afghanistan AFG 0
## 2 Sovereign country Angola AGO 0 Angola AGO 0
## 3 Dependency Anguilla AIA 0 Anguilla AIA 0
## 4 Sovereign country Albania ALB 0 Albania ALB 0
## 5 Country Aland ALD 0 Aland ALD 0
## subunit su_a3 brk_diff name name_long brk_a3 brk_name
## 0 Aruba ABW 0 Aruba Aruba ABW Aruba
## 1 Afghanistan AFG 0 Afghanistan Afghanistan AFG Afghanistan
## 2 Angola AGO 0 Angola Angola AGO Angola
## 3 Anguilla AIA 0 Anguilla Anguilla AIA Anguilla
## 4 Albania ALB 0 Albania Albania ALB Albania
```

```

## 5      Aland      ALD      0      Aland Aland Islands      ALD      Aland
## brk_group abbrev postal      formal_en formal_fr note_adm0
## 0      <NA>      Aruba      AW      Aruba      <NA>      Neth.
## 1      <NA>      Afg.      AF Islamic State of Afghanistan      <NA>      <NA>
## 2      <NA>      Ang.      AO People's Republic of Angola      <NA>      <NA>
## 3      <NA>      Ang.      AI      <NA>      <NA>      U.K.
## 4      <NA>      Alb.      AL      Republic of Albania      <NA>      <NA>
## 5      <NA>      Aland      AI      Åland Islands      <NA>      Fin.
## note_brk name_sort name_alt mapcolor7 mapcolor8 mapcolor9 mapcolor13
## 0      <NA>      Aruba      <NA>      4      2      2      9
## 1      <NA>      Afghanistan      <NA>      5      6      8      7
## 2      <NA>      Angola      <NA>      3      2      6      1
## 3      <NA>      Anguilla      <NA>      6      6      6      3
## 4      <NA>      Albania      <NA>      1      4      1      6
## 5      <NA>      Aland      <NA>      4      1      4      6
## pop_est gdp_md_est pop_year lastcensus gdp_year      economy
## 0      103065      2258.0      NA      2010      NA      6. Developing region
## 1      28400000      22270.0      NA      1979      NA      7. Least developed region
## 2      12799293      110300.0      NA      1970      NA      7. Least developed region
## 3      14436      108.9      NA      NA      NA      6. Developing region
## 4      3639453      21810.0      NA      2001      NA      6. Developing region
## 5      27153      1563.0      NA      NA      NA      2. Developed region: nonG7
## income_grp wikipedia fips_10 iso_a2 iso_a3 iso_n3 un_a3 wb_a2
## 0      2. High income: nonOECD      NA      <NA>      AW      ABW      533      533      AW
## 1      5. Low income      NA      <NA>      AF      AFG      004      004      AF
## 2      3. Upper middle income      NA      <NA>      AO      AGO      024      024      AO
## 3      3. Upper middle income      NA      <NA>      AI      AIA      660      660      <NA>
## 4      4. Lower middle income      NA      <NA>      AL      ALB      008      008      AL
## 5      1. High income: OECD      NA      <NA>      AX      ALA      248      248      <NA>
## wb_a3 woe_id adm0_a3_is adm0_a3_us adm0_a3_un adm0_a3_wb      continent
## 0      ABW      NA      ABW      ABW      NA      NA      North America
## 1      AFG      NA      AFG      AFG      NA      NA      Asia
## 2      AGO      NA      AGO      AGO      NA      NA      Africa
## 3      <NA>      NA      AIA      AIA      NA      NA      North America
## 4      ALB      NA      ALB      ALB      NA      NA      Europe
## 5      <NA>      NA      ALA      ALD      NA      NA      Europe
## region_un      subregion      region_wb name_len long_len
## 0      Americas      Caribbean Latin America & Caribbean      5      5
## 1      Asia      Southern Asia      South Asia      11      11
## 2      Africa      Middle Africa      Sub-Saharan Africa      6      6
## 3      Americas      Caribbean Latin America & Caribbean      8      8
## 4      Europe      Southern Europe      Europe & Central Asia      7      7
## 5      Europe      Northern Europe      Europe & Central Asia      5      13
## abbrev_len tiny homepart      geometry
## 0      5      4      NA MULTIPOLYGON (((-69.89912 1...
## 1      4      NA      1 MULTIPOLYGON (((74.89131 37...
## 2      4      NA      1 MULTIPOLYGON (((14.19082 -5...
## 3      4      NA      NA MULTIPOLYGON (((-63.00122 1...
## 4      4      NA      1 MULTIPOLYGON (((20.06396 42...
## 5      5      5      NA MULTIPOLYGON (((20.61133 60...

```

```

states <- map_data("state")
states %>%
  mutate(StateDesc = str_to_title(region)) -> states

```



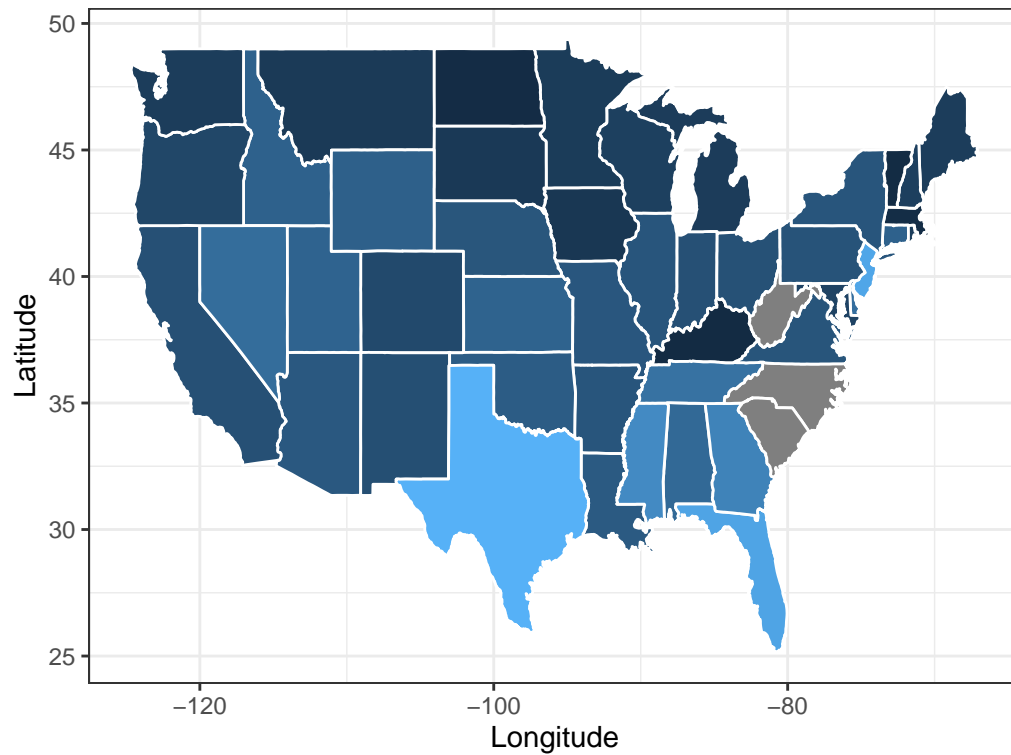
```

states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(insurance)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent No
Insurance",
       title = "Mean Percent Lack of Health Insurance across States",
       subtitle = "Data Retrieved from CDC 500 Cities")

```

Mean Percent Lack of Health Insurance across States
Data Retrieved from CDC 500 Cities



State Map of Health Insurance

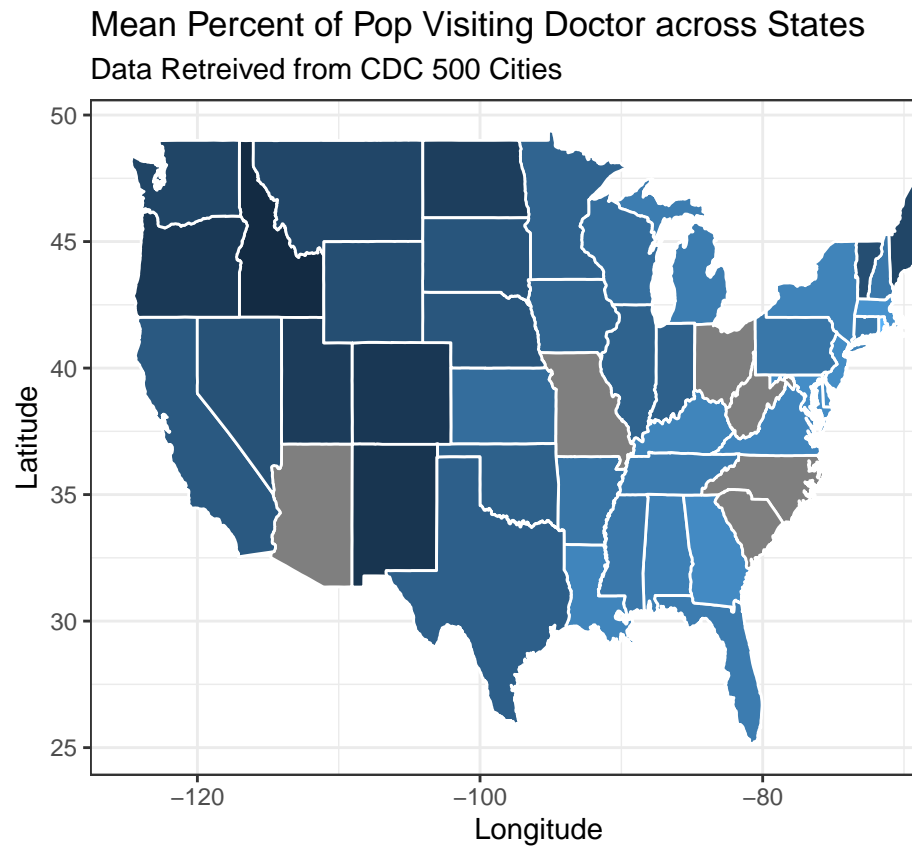
```

states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(visits_to_doctor)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +

```

```
geom_polygon(color = "white") +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Visiting Doctor",
       title = "Mean Percent of Pop Visiting Doctor across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

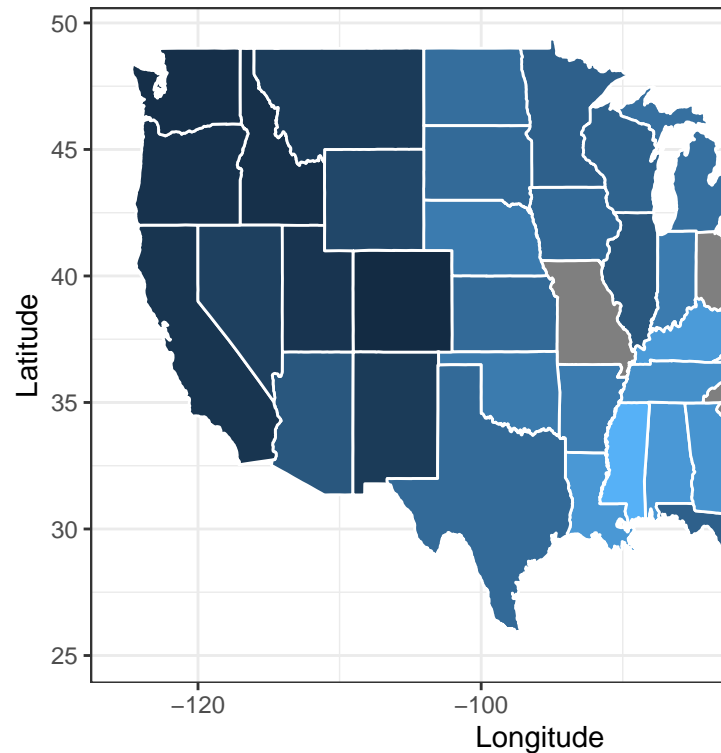


State Map of Visits to Doctor Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(medicine_high_bp)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent with
High BP Meds",
       title = "Mean Percent Pop with High BP Medicine across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

Mean Percent Pop with High BP Medicine
Data Retrieved from CDC 500 Cities

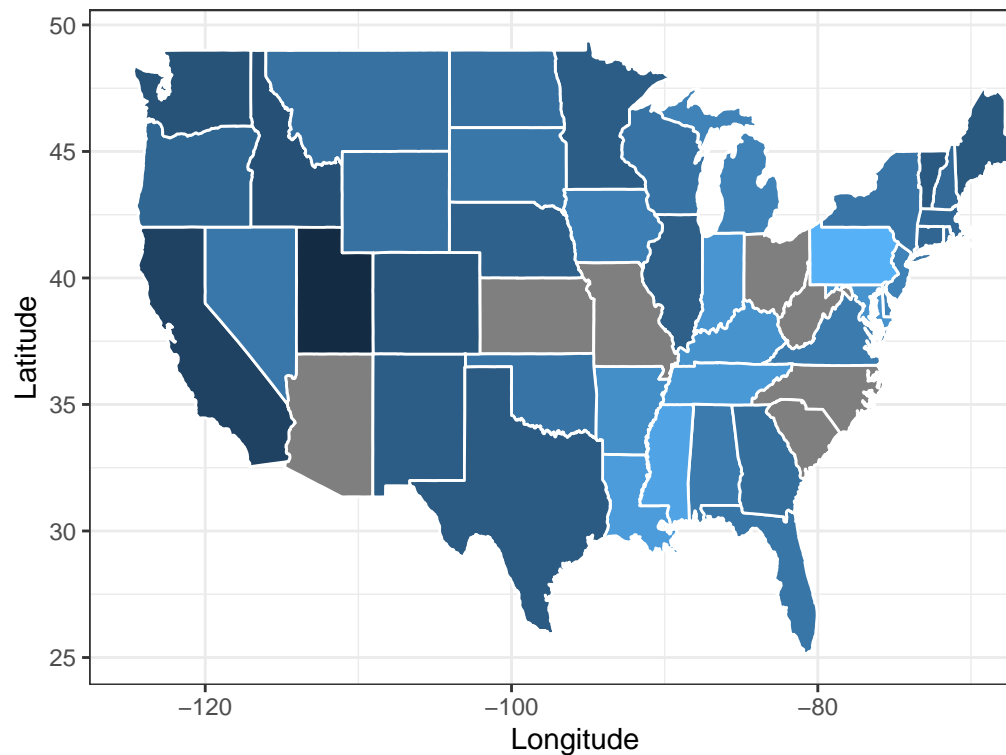


State Map of High Blood Pressure Medicine Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(smoking)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Smoking",
       title = "Mean Reported Percent Smoking across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

Mean Reported Percent Smoking across States
Data Retrieved from CDC 500 Cities

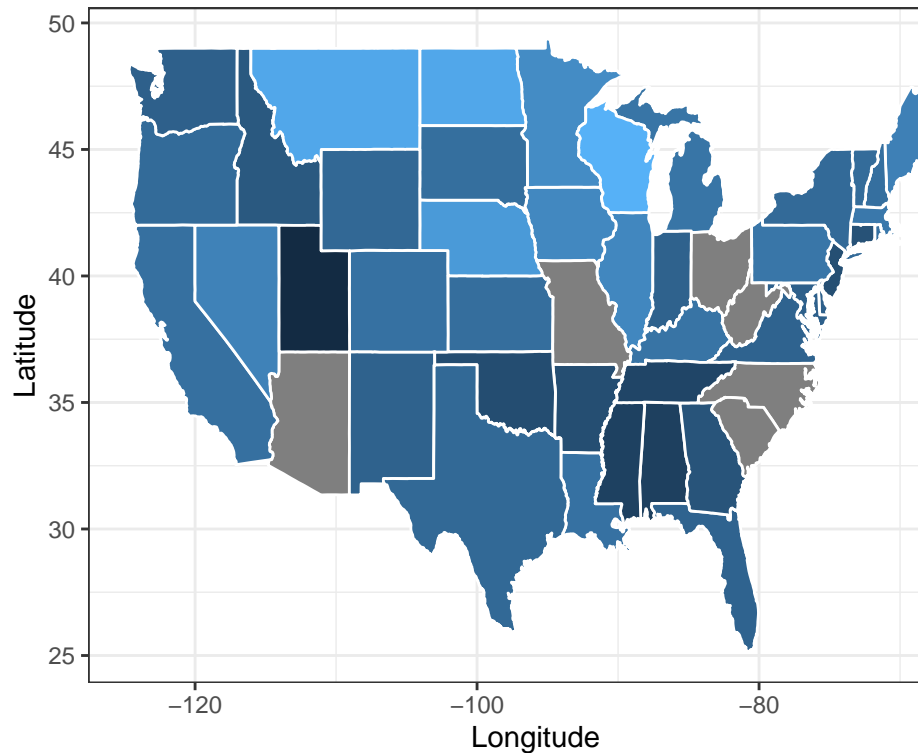


State Map of Smoking Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(binge_drinking)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Binge Drinking",
       title = "Mean Reported Percent Binge Drinking across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

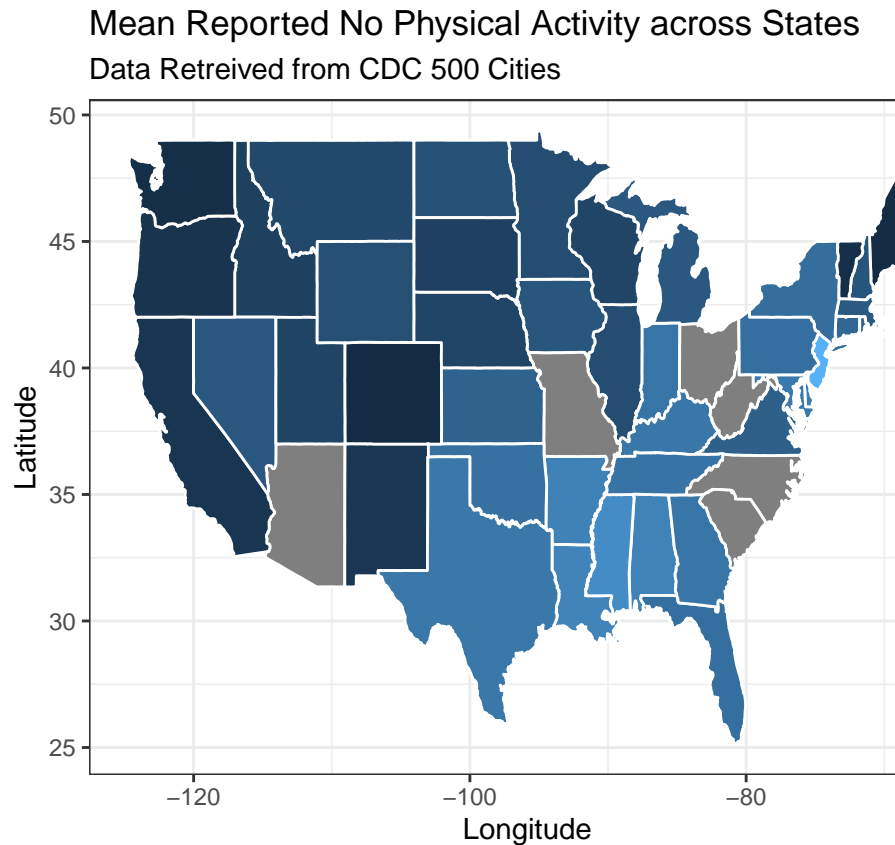
Mean Reported Percent Binge Drinking across States Data Retrieved from CDC 500 Cities



State Map of Binge Drinking Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(physical_activity)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent No
Physical Activity",
       title = "Mean Reported No Physical Activity across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

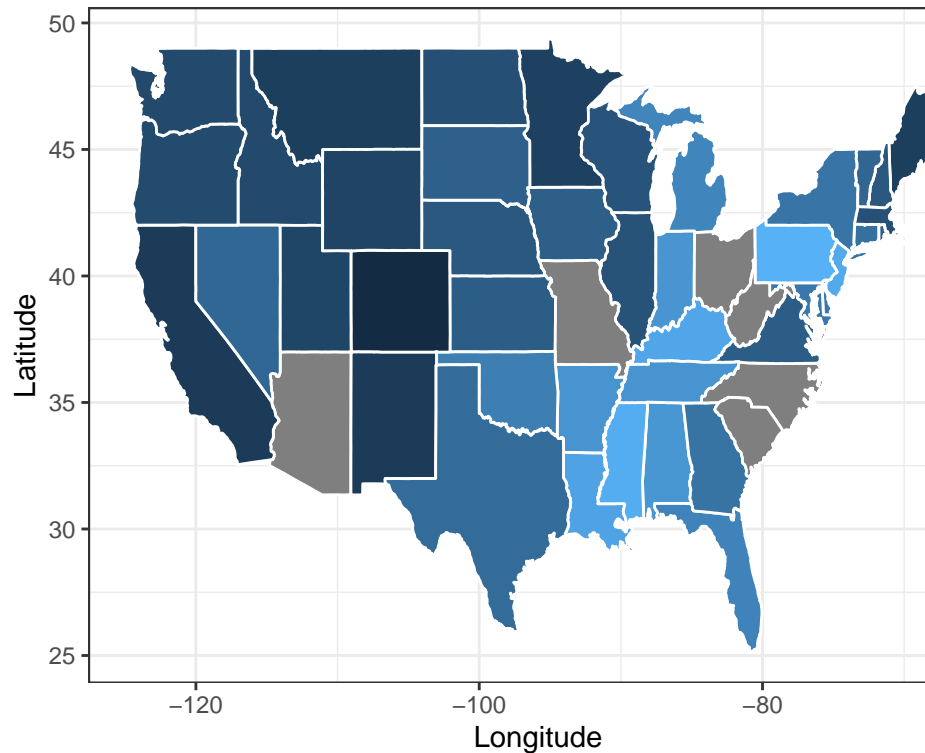


State Map of Physical Activity Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(heart_disease)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white") +
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Heart Disease",
       title = "Mean Percent Pop with Heart Disease across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

Mean Percent Pop with Heart Disease across States
Data Retrieved from CDC 500 Cities



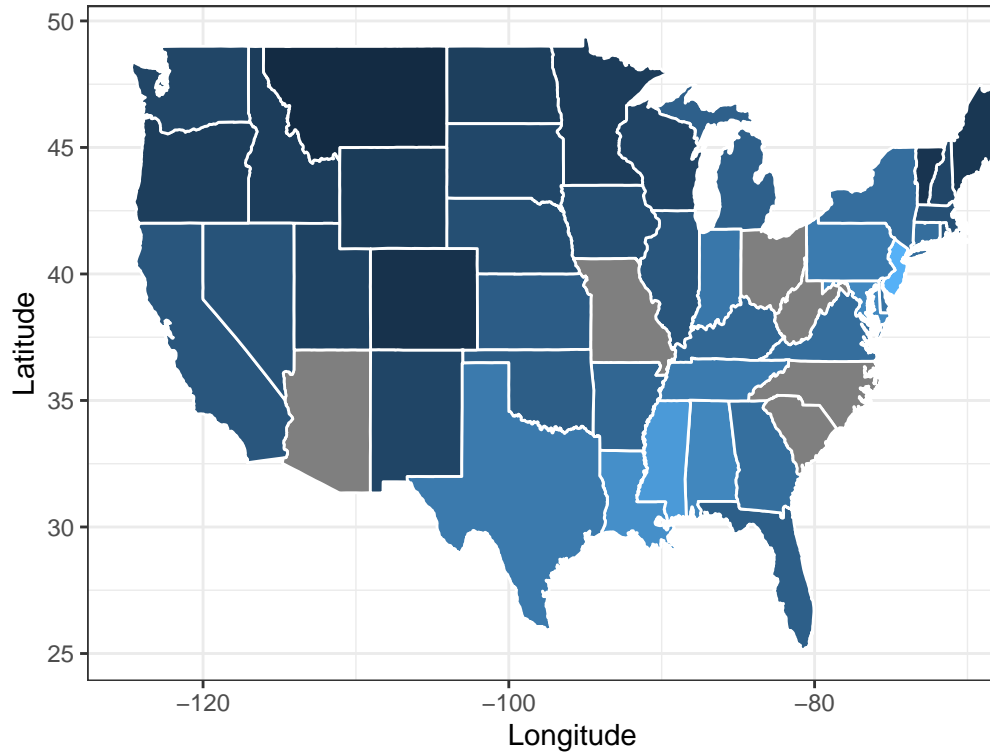
States Map of Heart Disease Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(diabetes)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Diabetes",
       title = "Mean Percent Pop with Diabetes across States",
       subtitle = "Data Retrieved from CDC 500 Cities")
```

Mean Percent Pop with Diabetes across States

Data Retrieved from CDC 500 Cities

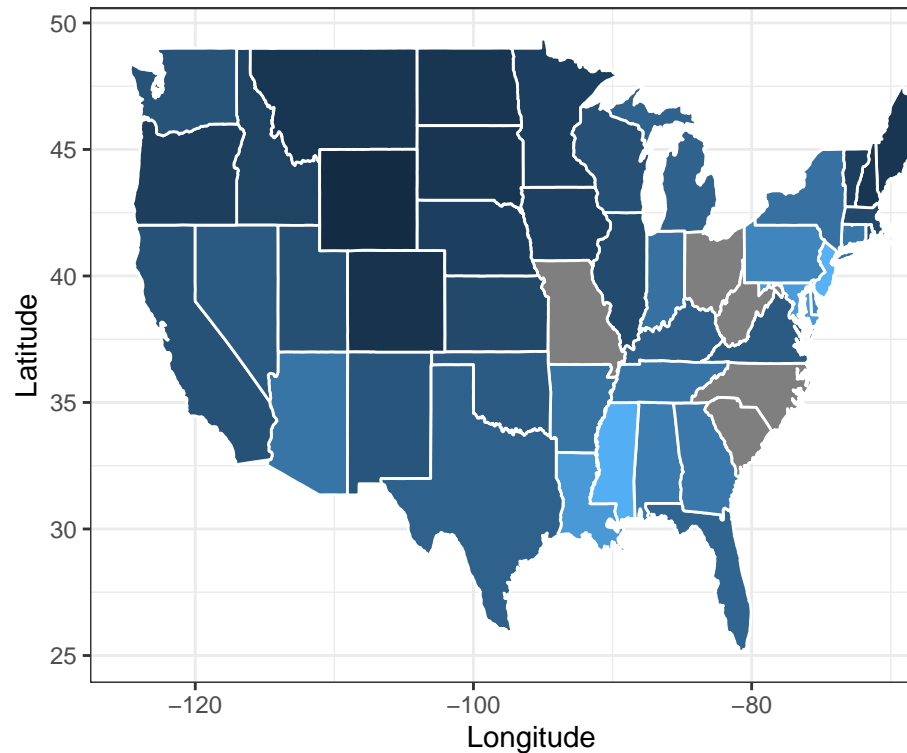


States Map of Diabetes Variable

```
states %>%
  left_join(data_500_cities, by = "StateDesc") -> test
data_500_cities %>%
  group_by(StateDesc) %>%
  summarise(test = mean(kidney_disease)) %>%
  right_join(states, by = "StateDesc") -> test2

ggplot(test2, aes(x = long, y = lat, group = group, fill = test)) +
  geom_polygon(color = "white")+
  labs(x = "Longitude",
       y = "Latitude",
       fill = "Percent
Kidney Disease",
       title = "Mean Percent Kidney Disease across States",
       subtitle = "Data from CDC 500 Cities")
```


Mean Percent Kidney Disease across States
Data from CDC 500 Cities



States Map of Kidney Disease Variable

ANOVA Assumptions Visualizations

First, I will filter the data so that states with at least 10 city observations are present.

```
data_500_cities %>%
  group_by(StateDesc) %>%
  summarize(n = n()) %>%
  print(n = 51)
```

```
## # A tibble: 51 x 2
##   StateDesc      n
##   <chr>        <int>
## 1 Alabama         6
## 2 Alaska          1
## 3 Arizona        12
## 4 Arkansas         5
## 5 California     120
## 6 Colorado        12
## 7 Connecticut      7
## 8 Delaware         1
## 9 District of C     1
## 10 Florida        33
## 11 Georgia         10
## 12 Hawaii          1
## 13 Idaho           3
## 14 Illinois        15
## 15 Indiana         10
```

## 16 Iowa	6
## 17 Kansas	6
## 18 Kentucky	2
## 19 Louisiana	5
## 20 Maine	1
## 21 Maryland	1
## 22 Massachusetts	11
## 23 Michigan	16
## 24 Minnesota	6
## 25 Mississippi	2
## 26 Missouri	7
## 27 Montana	2
## 28 Nebraska	2
## 29 Nevada	5
## 30 New Hampshire	2
## 31 New Jersey	8
## 32 New Mexico	4
## 33 New York	7
## 34 North Carolina	10
## 35 North Dakota	1
## 36 Ohio	9
## 37 Oklahoma	6
## 38 Oregon	7
## 39 Pennsylvania	7
## 40 Rhode Island	4
## 41 South Carolina	4
## 42 South Dakota	2
## 43 Tennessee	6
## 44 Texas	46
## 45 United States	1
## 46 Utah	9
## 47 Vermont	1
## 48 Virginia	10
## 49 Washington	14
## 50 Wisconsin	7
## 51 Wyoming	1

Based on the table above, Arizona, California, Colorado, Florida, Georgia, Illinois, Indiana, Massachusetts, Michigan, North Carolina, Texas, Virginia, and Washington are the states with at least 10 observations. These states will be the only ones considered in the ANOVA test. Note that California has significantly more observations than all other states.

I will now create a new dataset that filters for the states and mutates new log versions of each variable.

```
ANOVA_data_500_cities <- data_500_cities %>%
  filter(StateDesc %in% c("Arizona", "California", "Colorado", "Florida", "Georgia", "Illinois", "Indiana", "Massachusetts", "Michigan", "North Carolina", "Texas", "Virginia", "Washington")) %>%
  mutate(linsurance = log(insurance)) %>%
  mutate(lvisits_to_doctor = log(visits_to_doctor)) %>%
  mutate(lmedicine_high_bp = log(medicine_high_bp)) %>%
  mutate(lheart_disease = log(heart_disease)) %>%
  mutate(ldiabetes = log(diabetes)) %>%
  mutate(lkidney_disease = log(kidney_disease))
```

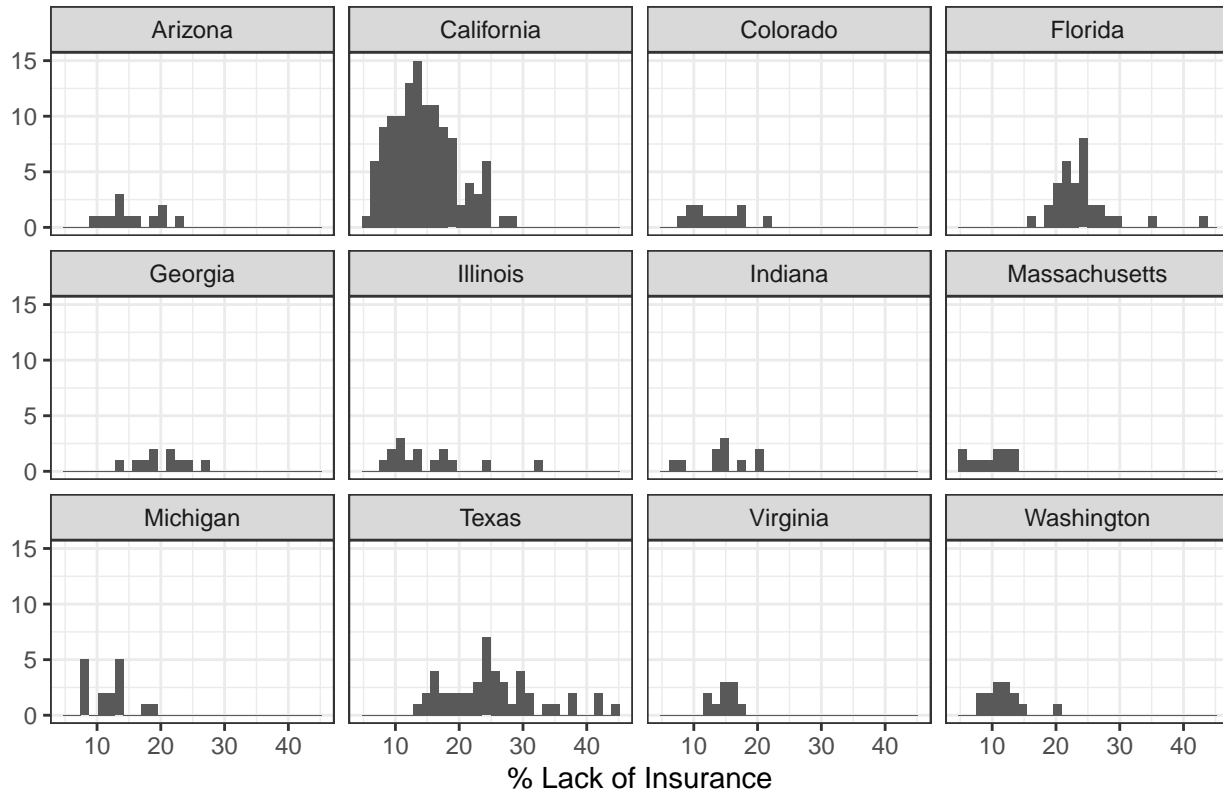
Assumptions of the ANOVA Test are as Follows:

- 1) Outcomes within groups are normally distributed

- 2) Homoscedastic variance (same variance of individual observations in each group)
- 3) Samples are independent. This is likely not the case for this data, so we can compensate for this with a Bonferroni Correction or a Random Effects Model.

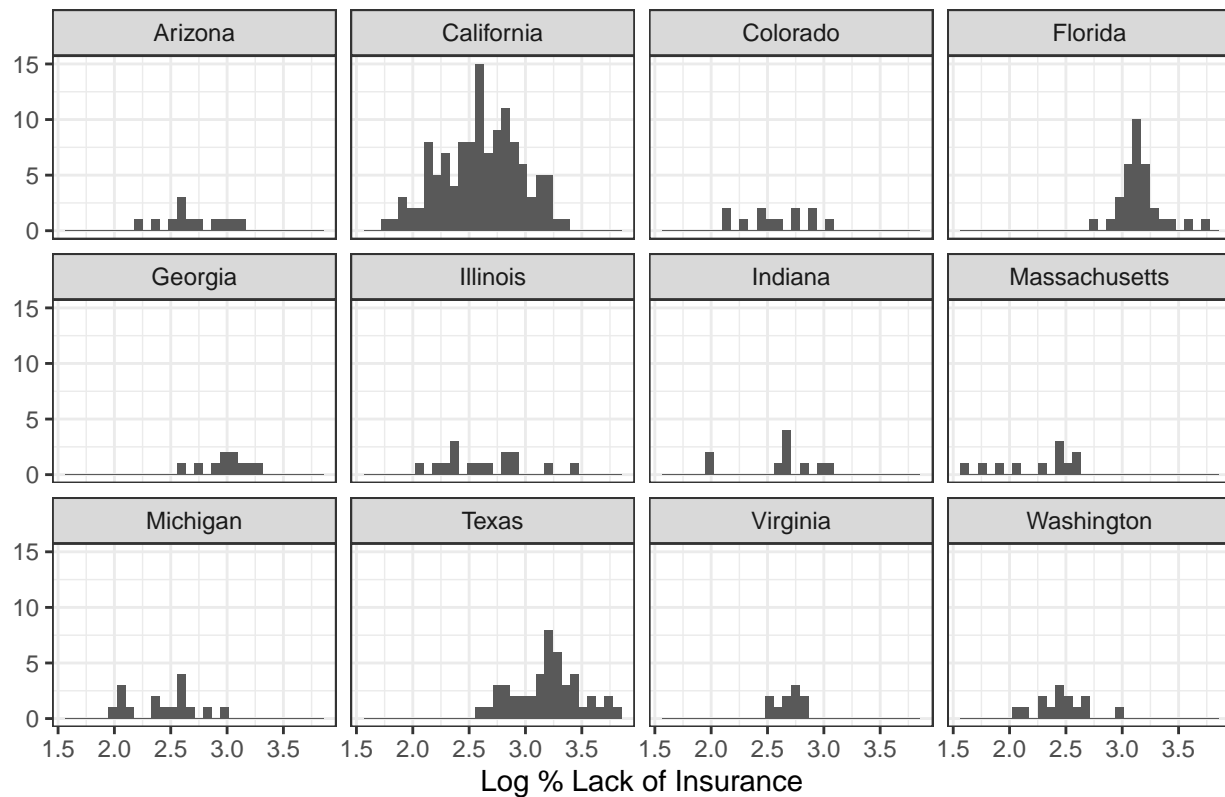
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = insurance)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% Lack of Insurance", y = NULL, title = "Distribution of % Lack of Insurance Grouped By S
```

Distribution of % Lack of Insurance Grouped By State



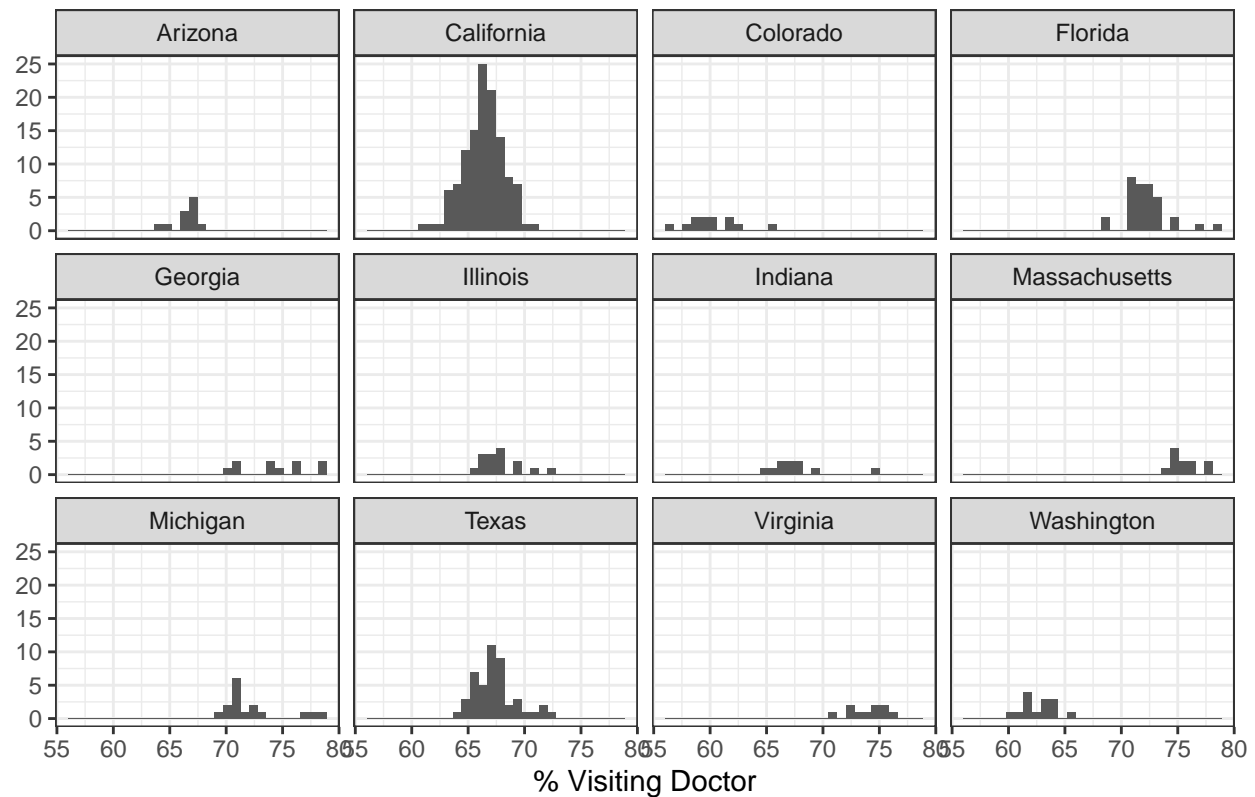
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = linsruance)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "Log % Lack of Insurance", y = NULL, title = "Distribution of Log % Lack of Insurance Group
```

Distribution of Log % Lack of Insurance Grouped By State



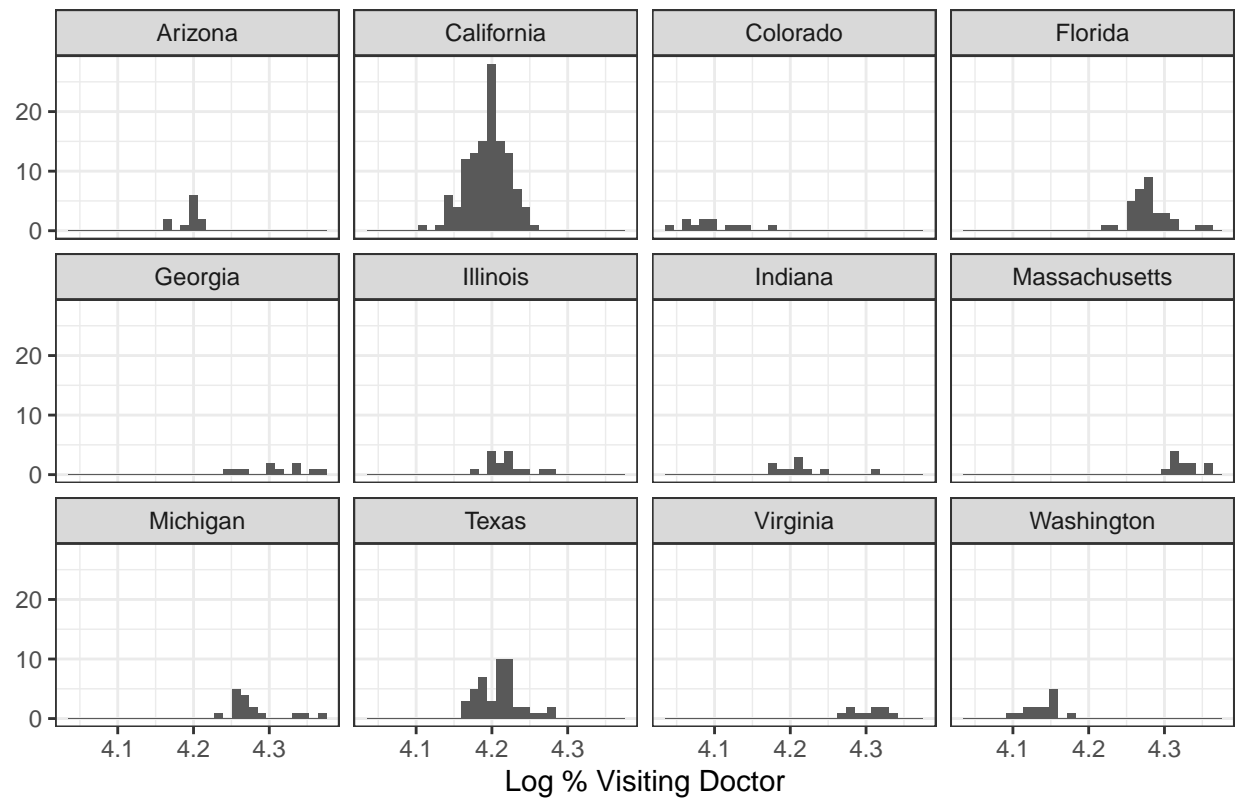
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = visits_to_doctor)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% Visiting Doctor", y = NULL, title = "Distribution of % Visiting Doctor Grouped By State")
```

Distribution of % Visiting Doctor Grouped By State



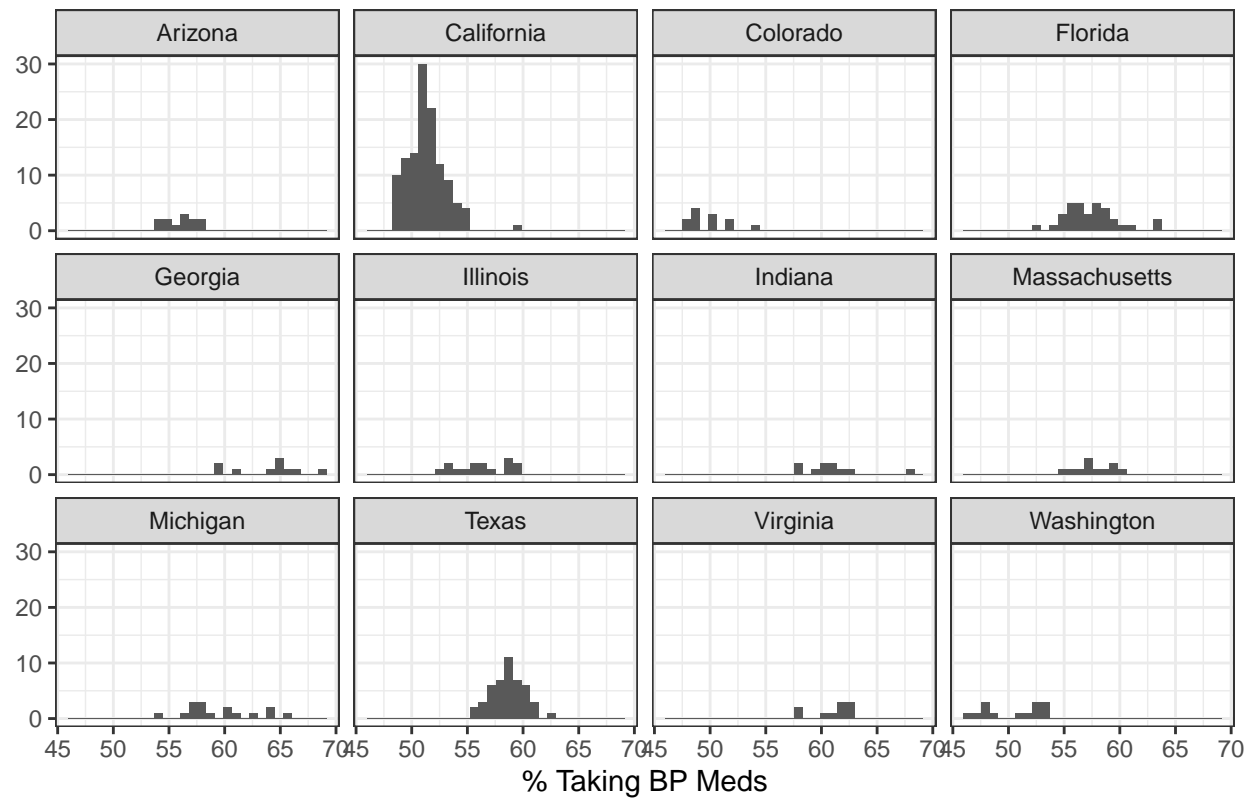
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = lvisits_to_doctor)) +
    geom_histogram() +
    facet_wrap(~StateDesc) +
    labs(x = "Log % Visiting Doctor", y = NULL, title = "Distribution of Log % Visiting Doctor Grouped By State")
```

Distribution of Log % Visiting Doctor Grouped By State



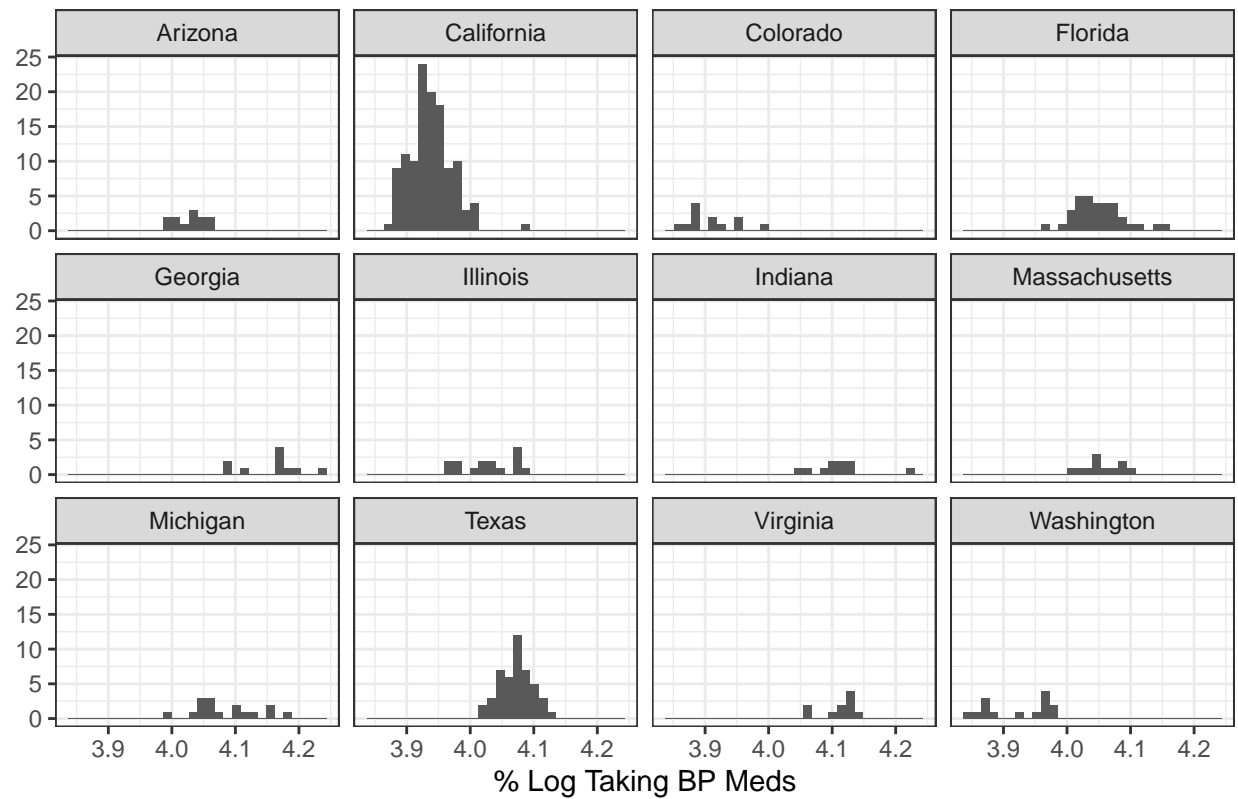
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = medicine_high_bp)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% Taking BP Meds", y = NULL, title = "Distribution of % Taking BP Meds Grouped By State")
```

Distribution of % Taking BP Meds Grouped By State



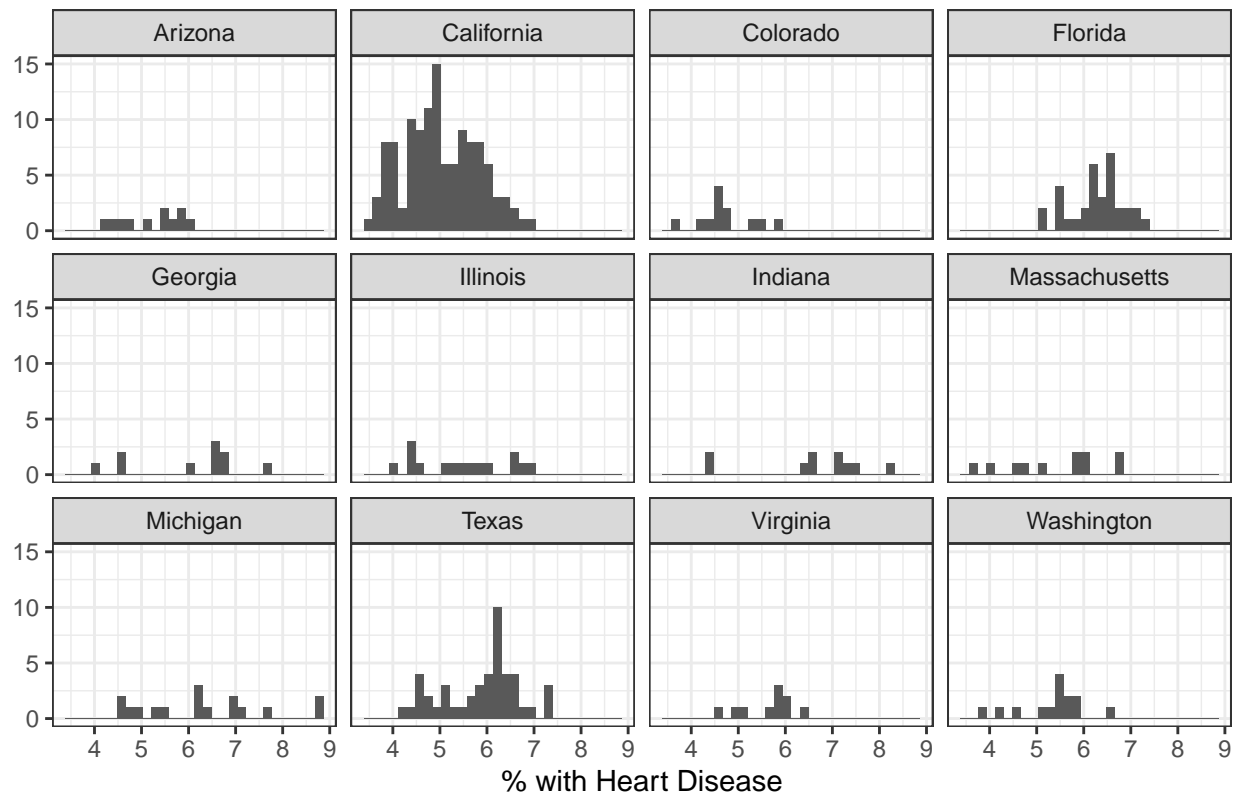
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = lmedicine_high_bp)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% Log Taking BP Meds", y = NULL, title = "Distribution of Log % Taking BP Meds Grouped By
```

Distribution of Log % Taking BP Meds Grouped By State



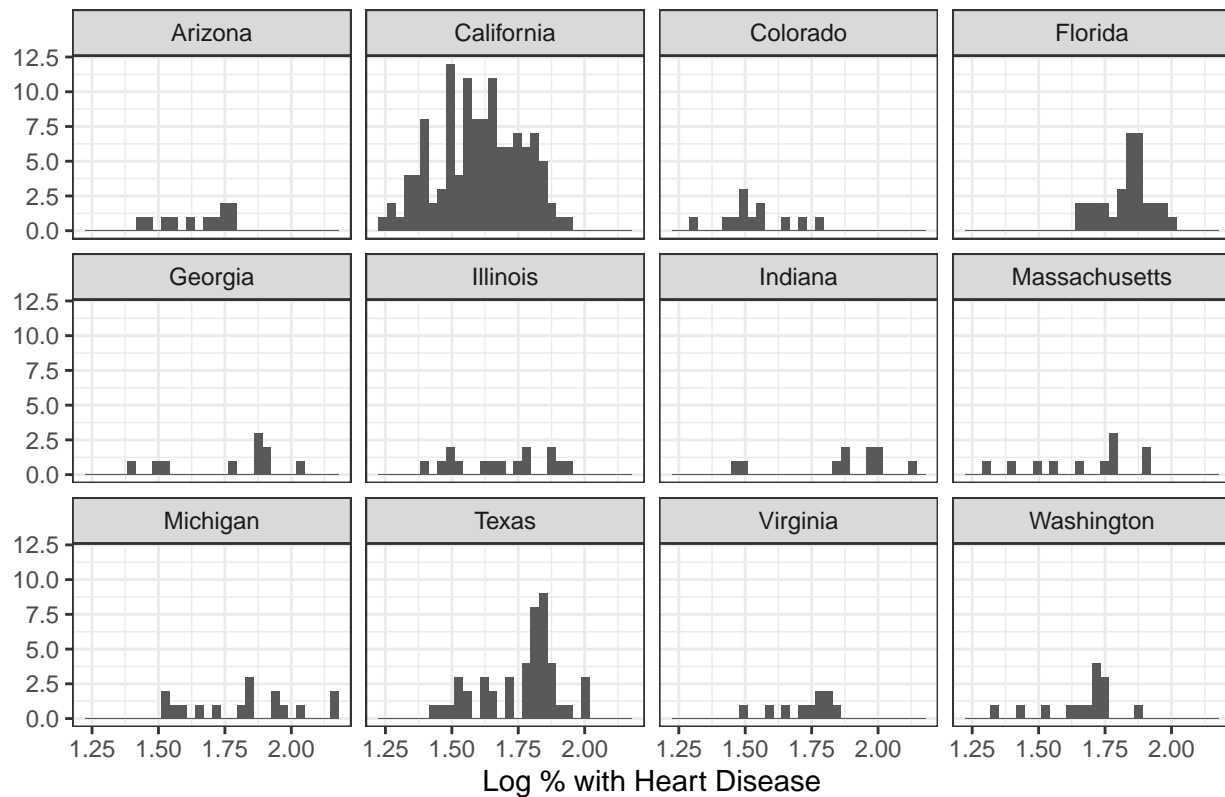
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = heart_disease)) +
    geom_histogram() +
    facet_wrap(~StateDesc) +
    labs(x = "% with Heart Disease", y = NULL, title = "Distribution of % with Heart Disease Grouped By
```


Distribution of % with Heart Disease Grouped By State



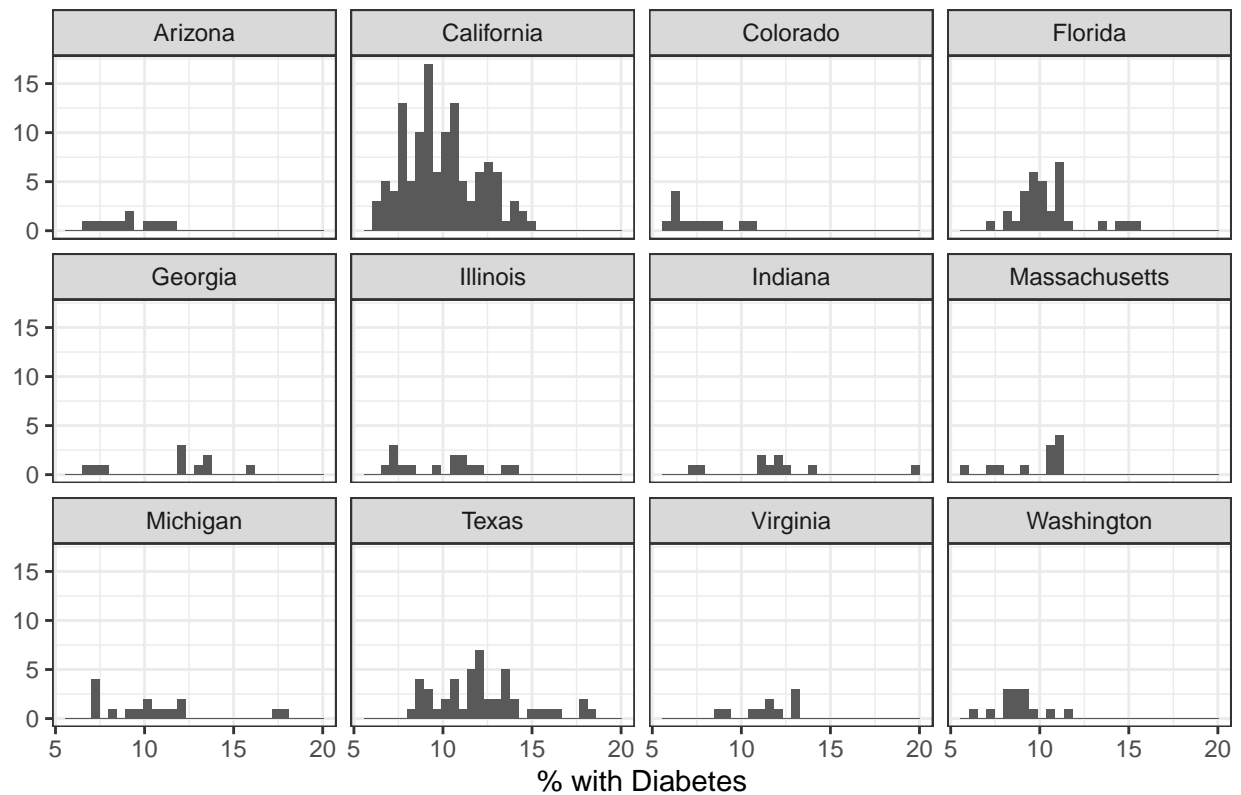
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = lheart_disease)) +
    geom_histogram() +
    facet_wrap(~StateDesc) +
    labs(x = "Log % with Heart Disease", y = NULL, title = "Distribution of % Log with Heart Disease Gr
```

Distribution of % Log with Heart Disease Grouped By State



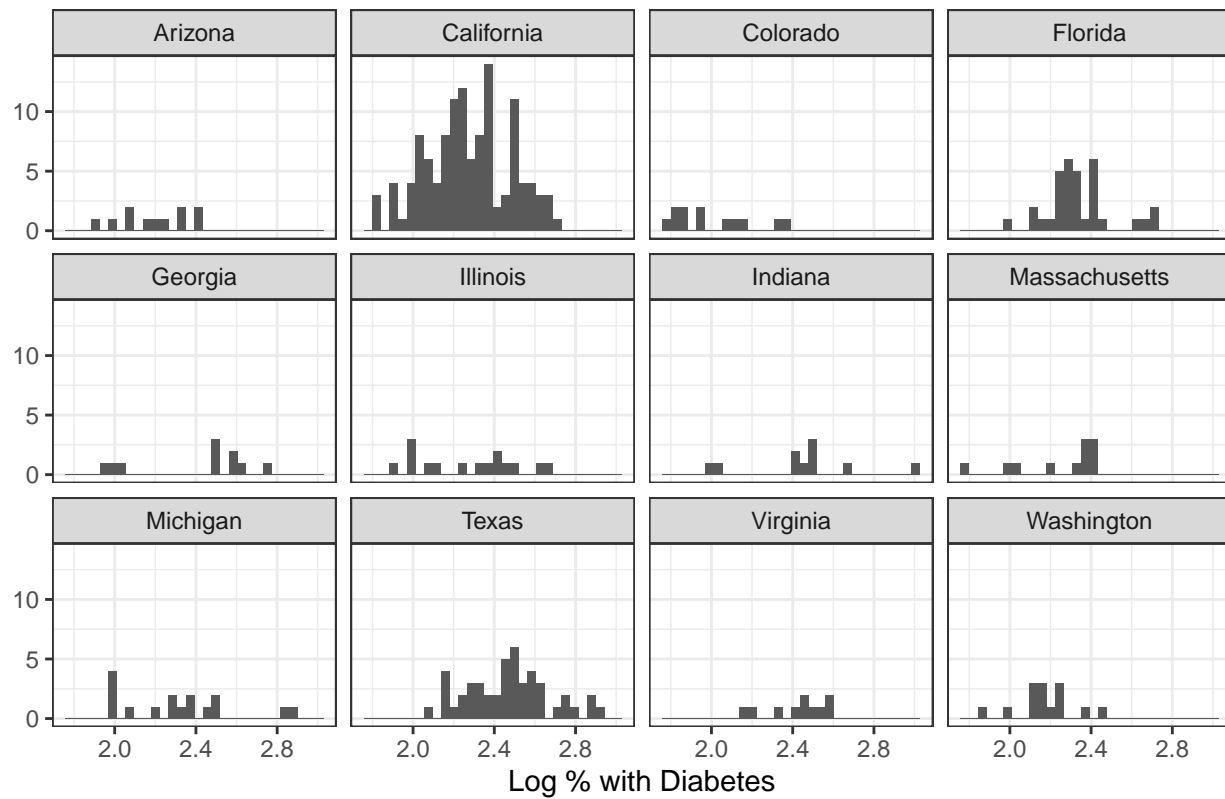
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = diabetes)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% with Diabetes", y = NULL, title = "Distribution of % with Diabetes Grouped By State")
```

Distribution of % with Diabetes Grouped By State



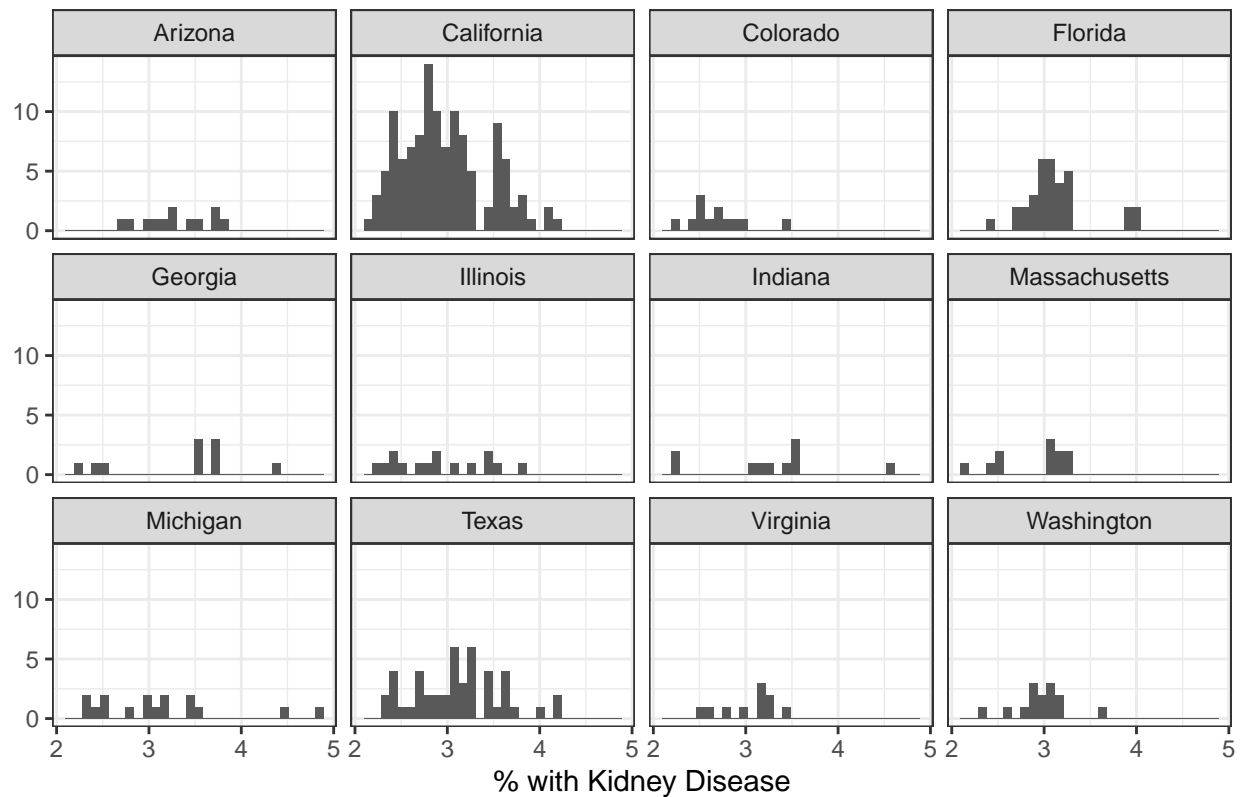
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = ldiabetes)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "Log % with Diabetes", y = NULL, title = "Distribution of Log % with Diabetes Grouped By S
```

Distribution of Log % with Diabetes Grouped By State



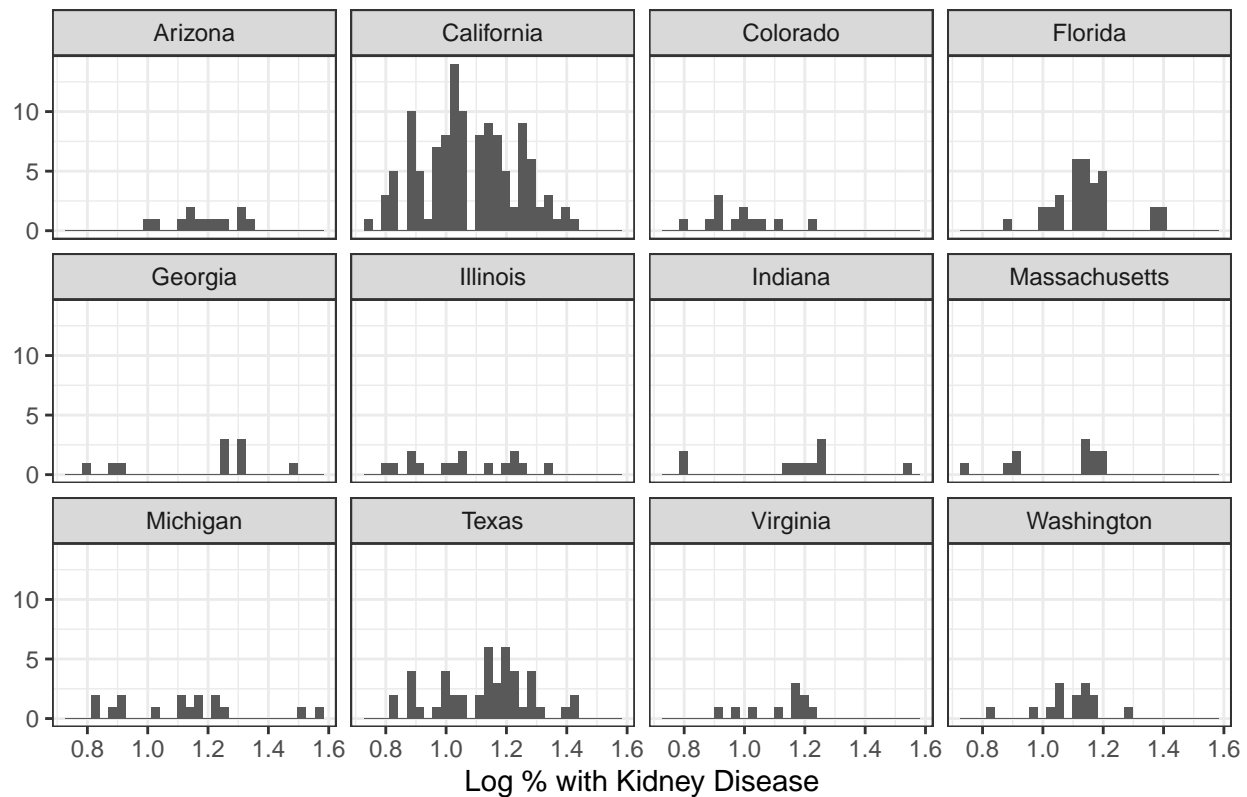
```
ANOVA_data_500_cities %>%
  ggplot(aes(x = kidney_disease)) +
  geom_histogram() +
  facet_wrap(~StateDesc) +
  labs(x = "% with Kidney Disease", y = NULL, title = "Distribution of % with Kidney Disease Grouped By State")
```

Distribution of % with Kidney Disease Grouped By State



```
ANOVA_data_500_cities %>%
  ggplot(aes(x = lkidney_disease)) +
    geom_histogram() +
    facet_wrap(~StateDesc) +
    labs(x = "Log % with Kidney Disease", y = NULL, title = "Distribution of Log % with Kidney Disease (
```

Distribution of Log % with Kidney Disease Grouped By State



Based on the visuals above, the variables with the most normal distribution and variance within state groups are insurance, visits_to_doctor, lmedicine_high_bp, and lheartdisease. These will be the variables examined in the ANOVA testing. Independence will be accounted for with a random effects model.

Overall Tests

Step Down Tests