

# Causal Diagrams in R

Malcolm Barrett

Apple

2021-09-01 (updated: 2021-10-21)

**Draw your causal assumptions  
with causal directed acyclic  
graphs (DAGs)**

# The basic idea

- 1 Specify your causal question
- 2 Use domain knowledge
- 3 Write variables as nodes
- 4 Write causal pathways as arrows (edges)

ggdag

dagitty

ggplot2  
ggraph

dagitty

powerful,  
robust  
algorithms

ggplot2  
gggraph

dagitty

powerful,  
robust  
algorithms

ggplot2  
gggraph


unlimited  
flexibility

beautiful  
plots

dagitty

ggplot2  
ggraph

Data  
structure:  
tidy DAGs



```
graph TD; A[Data structure: tidy DAGs] --> B[dagitty]; A --> C[ggplot2]; A --> D[ggraph];
```



# Step 1: Specify your DAG

```
dagify(  
  cancer ~ smoking,  
  coffee ~ smoking  
)
```

# Step 1: Specify your DAG

```
dagify(  
  cancer ~ smoking,  
  coffee ~ smoking  
)
```

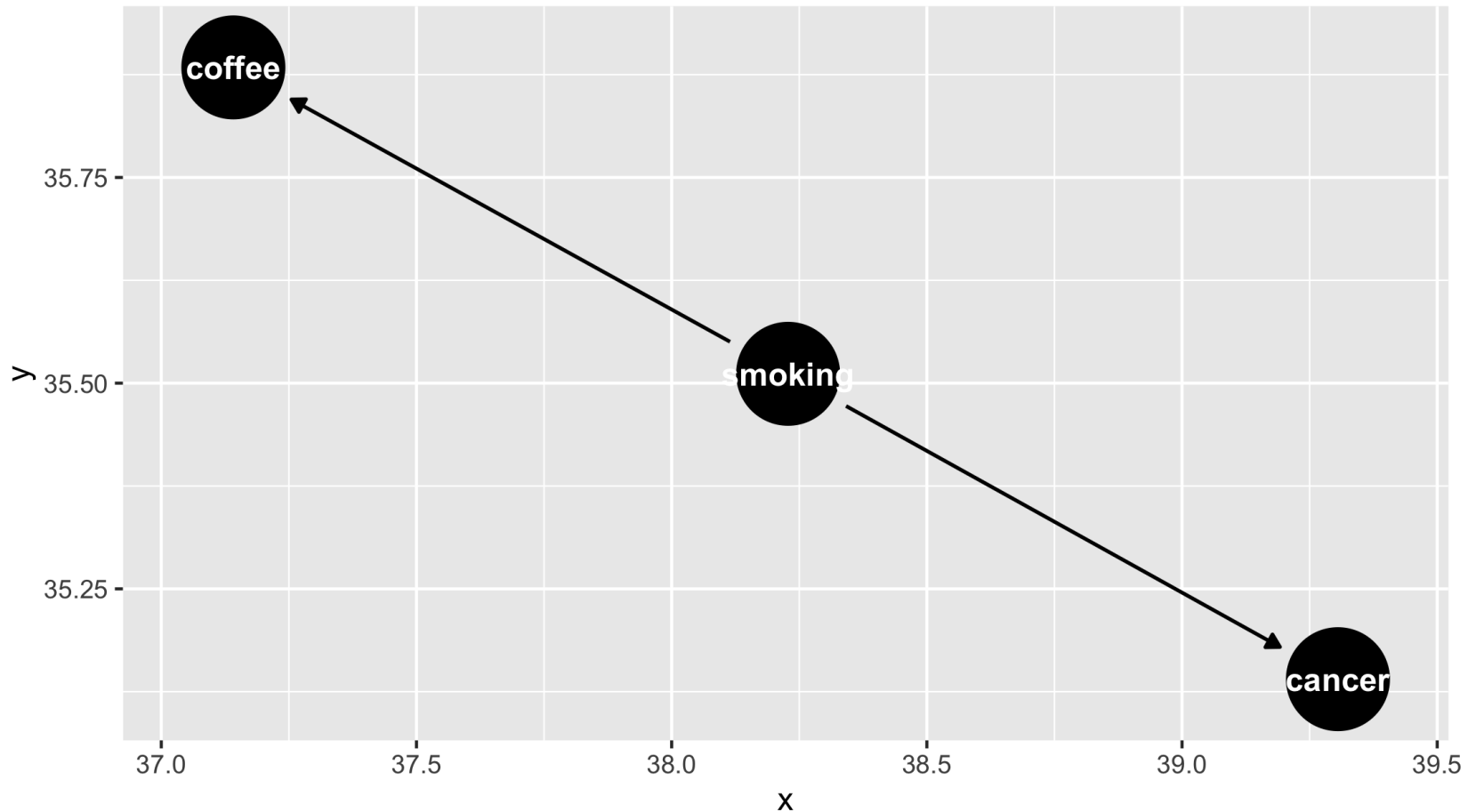
# Step 1: Specify your DAG

```
dagify(  
  cancer ~ smoking,  
  coffee ~ smoking  
)
```

# Step 1: Specify your DAG

```
dagify(  
  cancer ~ smoking,  
  coffee ~ smoking  
) %>% ggdag()
```

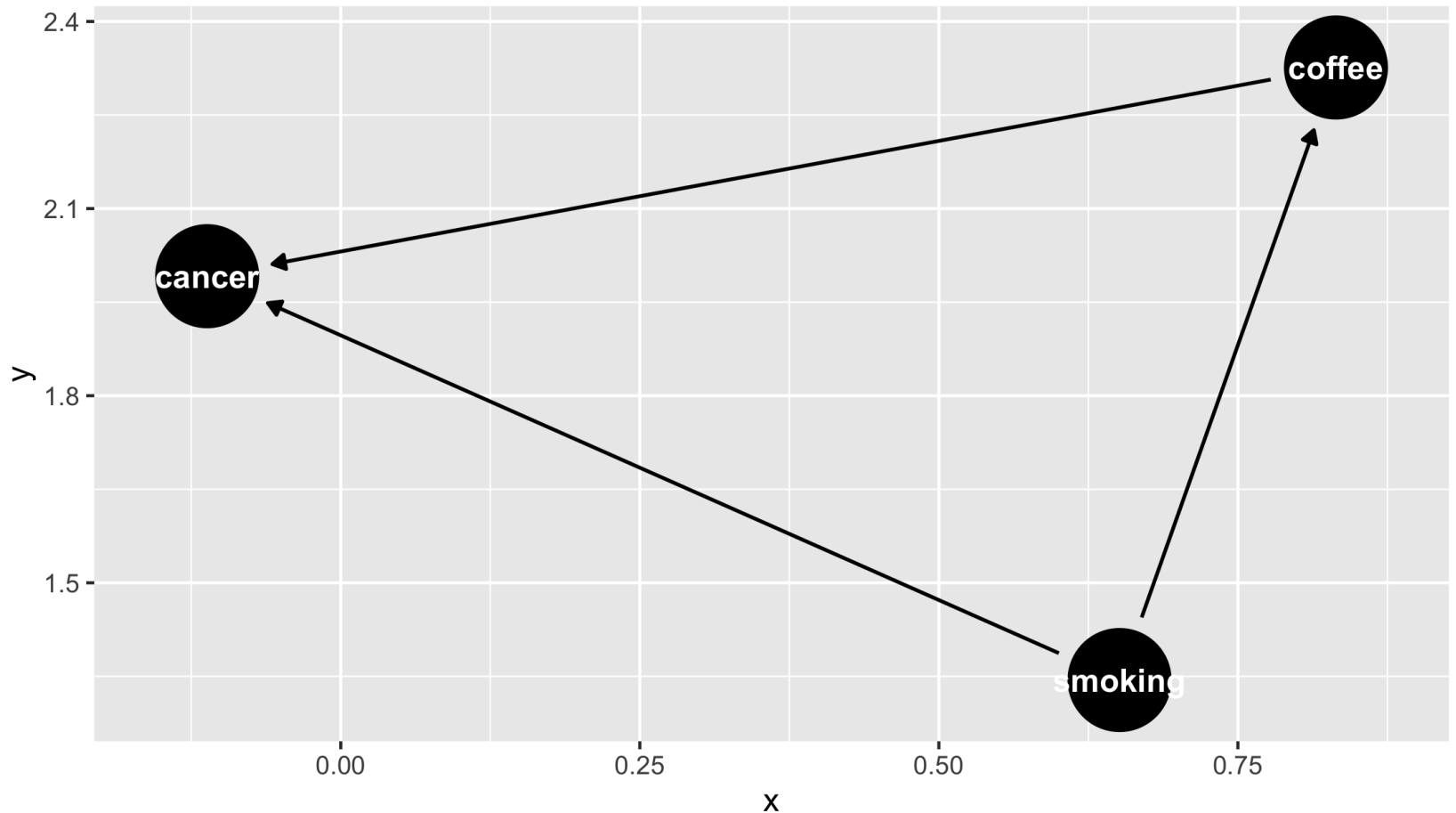
# Step 1: Specify your DAG



# Step 1: Specify your DAG

```
dagify(  
  cancer ~ smoking + coffee,  
  coffee ~ smoking  
) %>% ggdag()
```

# Step 1: Specify your DAG



## Your Turn 1 (02-dags-exercises.Rmd)

**Specify a DAG with `dagify()`. Write your assumption that smoking causes cancer as a formula.**

**We're going to assume that coffee does not cause cancer, so there's no formula for that. But we still need to declare our causal question. Specify "coffee" as the exposure and "cancer" as the outcome (both in quotations marks).**

**Plot the DAG using `ggdag()`**

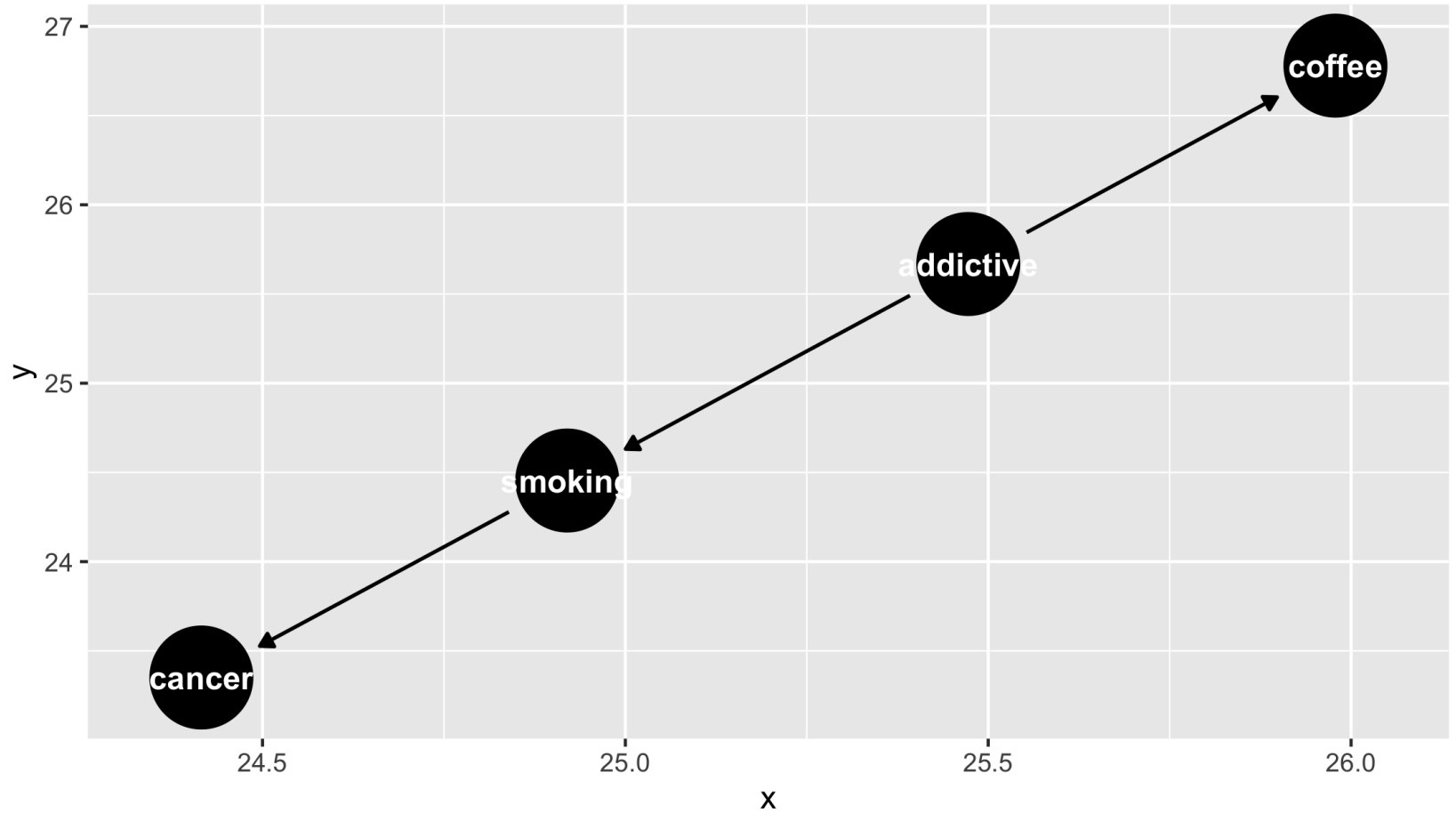
05:00



# Your Turn 1 (02-dags-exercises.Rmd)

```
coffee_cancer_dag <- dagify(  
  cancer ~ smoking,  
  smoking ~ addictive,  
  coffee ~ addictive,  
  exposure = "coffee",  
  outcome = "cancer",  
  labels = c(  
    "coffee" = "Coffee",  
    "cancer" = "Lung Cancer",  
    "smoking" = "Smoking",  
    "addictive" = "Addictive \nBehavior"  
  )  
)
```

```
ggdag(coffee_cancer_dag)
```



# Causal effects and backdoor paths

# Causal effects and backdoor paths

**Ok, correlation  $\neq$  causation. But why not?**

# Causal effects and backdoor paths

Ok, correlation  $\neq$  causation. But why not?

**We want to know if  $x \rightarrow y$ ...**

# Causal effects and backdoor paths

Ok, correlation  $\neq$  causation. But why not?

We want to know if  $x \rightarrow y$ ...

**But other paths also cause associations**

# ggdag\_paths()

## Identify "backdoor" paths

```
ggdag_paths(smk_wt_dag)
```





## Your Turn 2

**Call `tidy_dagitty()` on `coffee_cancer_dag` to create a tidy DAG, then pass the results to `dag_paths()`.  
What's different about these data?**

**Plot the open paths with `ggdag_paths()`. (Just give it `coffee_cancer_dag` rather than using `dag_paths()`; the quick plot function will do that for you.)**

**Remember, since we assume there is **no** causal path from coffee to lung cancer, any open paths must be confounding pathways.**

05:00

# Your Turn 2

```
coffee_cancer_dag %>%  
  tidy_dagitty() %>%  
  dag_paths()
```

```
## # A DAG with 4 nodes and 3 edges
```

```
## #
```

```
## # Exposure: coffee
```

```
## # Outcome: cancer
```

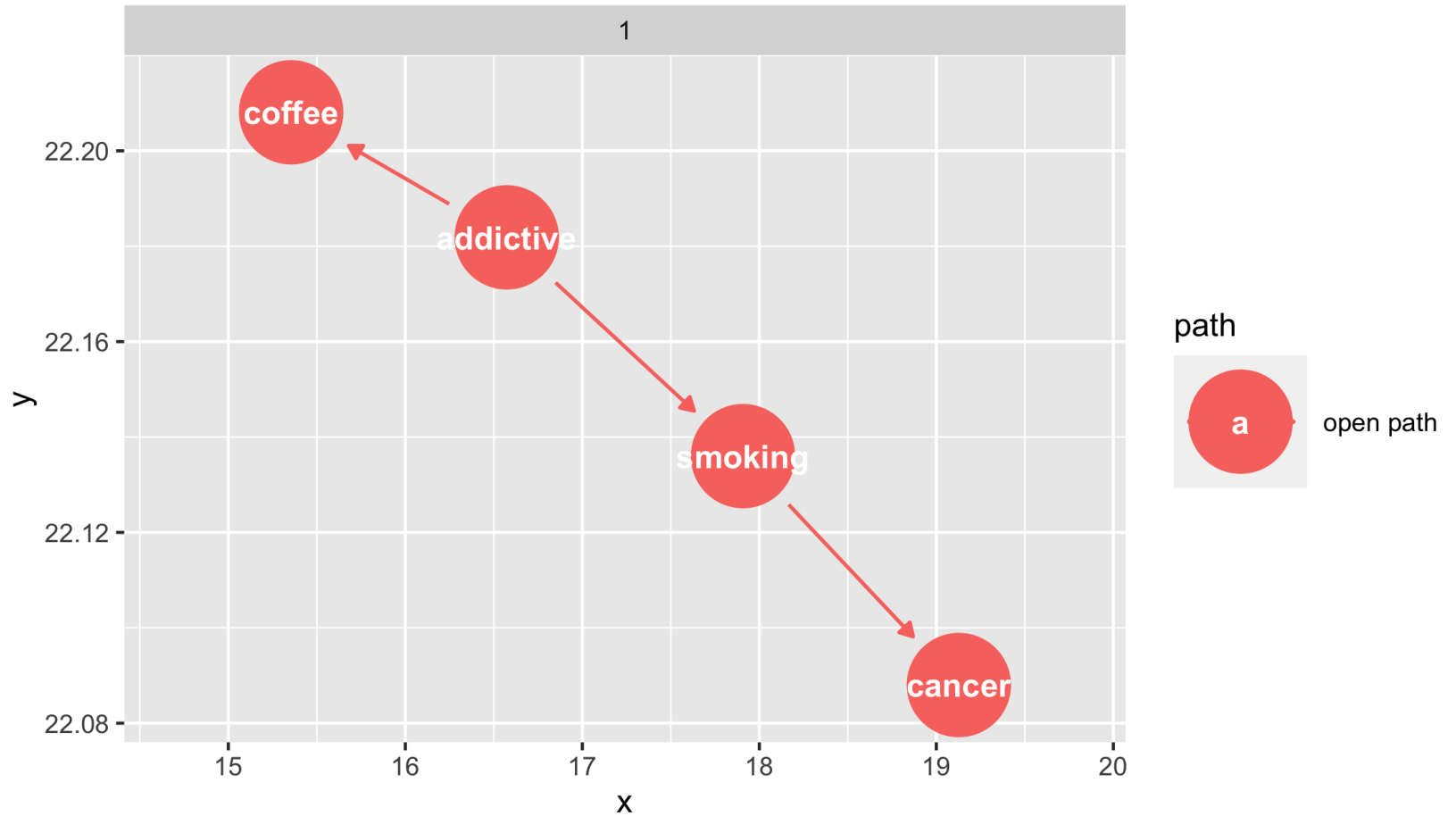
```
## #
```

```
## # A tibble: 5 × 11
```

##	set	name	x	y	direction	to	xend	yend
##	<chr>	<chr>	<dbl>	<dbl>	<fct>	<chr>	<dbl>	<dbl>
## 1	1	addictive	25.7	28.1	->	coffee	24.5	27.9
## 2	1	addictive	25.7	28.1	->	smoking	27.1	28.2
## 3	1	smoking	27.1	28.2	->	cancer	28.3	28.3
## 4	1	coffee	24.5	27.9	<NA>	<NA>	NA	NA
## 5	1	cancer	28.3	28.3	<NA>	<NA>	NA	NA

```
## # ... with 3 more variables: circular <lgl>, label <chr>,  
## # path <chr>
```

```
coffee_cancer_dag %>%  
  ggdag_paths()
```



# Closing backdoor paths

# Closing backdoor paths

**We need to account for these open, non-causal paths**

# Closing backdoor paths

We need to account for these open, non-causal paths

## Randomization

# Closing backdoor paths

We need to account for these open, non-causal paths

Randomization

**Stratification, adjustment, weighting, matching, etc.**

# Identifying adjustment sets

```
ggdag_adjustment_set(smk_wt_dag)
```





# Your Turn 3

Now that we know the open, confounding pathways (sometimes called "backdoor paths"), we need to know how to close them! First, we'll ask {ggdag} for adjustment sets, then we would need to do something in our analysis to account for at least one adjustment set (e.g. multivariable regression, weighting, or matching for the adjustment sets).

**Use `ggdag_adjustment_set()` to visualize the adjustment sets. Add the arguments `use_labels = "label"` and `text = FALSE`.**

**Write an R formula for each adjustment set, as you might if you were fitting a model in `lm()` or `glm()`**

05:00

# Your Turn 3

```
ggdag_adjustment_set(  
  coffee_cancer_dag,  
  use_labels = "label",  
  text = FALSE  
)
```



# Your Turn 3

cancer ~ coffee + addictive

cancer ~ coffee + smoking

# Resources: ggdag vignettes

An Introduction to ggdag

An Introduction to Directed Acyclic  
Graphs

Common Structures of Bias